Hindawi
Complexity
Volume 2019, Article ID 7189653, 36 pages
https://doi.org/10.1155/2019/7189653

Research Article

Dynamically Dimensioned Search Grey Wolf Optimizer Based on Positional Interaction Information

Fu Yan,[1] **Jianzhong Xu**,[1] **and Kumchol Yun**[1,2]

[1]*School of Economics and Management, Harbin Engineering University, Harbin 150001, China*
[2]*Faculty of Mechanics, Kim Il Sung University, Pyongyang 950003, Democratic People's Republic of Korea*

Correspondence should be addressed to Jianzhong Xu; xujianzhongxjz@163.com

Received 22 June 2019; Revised 13 September 2019; Accepted 21 October 2019; Published 5 December 2019

Guest Editor: Raúl Baños

Copyright © 2019 Fu Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The grey wolf optimizer (GWO) algorithm is a recently developed, novel, population-based optimization technique that is inspired by the hunting mechanism of grey wolves. The GWO algorithm has some distinct advantages, such as few algorithm parameters, strong global optimization ability, and ease of implementation on a computer. However, the paramount challenge is that there are some cases where the GWO is prone to stagnation in local optima. This drawback of the GWO algorithm may be attributed to an insufficiency in its position-updated equation, which disregards the positional interaction information about the three best grey wolves (i.e., the three leaders). This paper proposes an improved version of the GWO algorithm that is based on a dynamically dimensioned search, spiral walking predation technique, and positional interaction information (referred to as the DGWO). In addition, a nonlinear control parameter strategy, i.e., the control parameter that is nonlinearly increased with an increase in iterations, is designed to balance the exploration and exploitation of the GWO algorithm. The experimental results for 23 general benchmark functions and 3 well-known engineering optimization design applications validate the effectiveness and feasibility of the proposed DGWO algorithm. The comparison results for the 23 benchmark functions show that the proposed DGWO algorithm performs significantly better than the GWO and its improved variant for most benchmarks. The DGWO provides the highest solution precision, strongest robustness, and fastest convergence rate among the compared algorithms in almost all cases.

1. Introduction

The rapid development of artificial intelligence (AI) is primarily attributed to the considerable progress of computational intelligence (CI). CI that is based on complex systems mainly consists of two categories [1], i.e., single-solution-based metaheuristics and population-based metaheuristics. Both single-solution-based algorithms and population-based algorithms employ a variety of mechanisms and are designed to solve extremely challenging problems in different complex systems.

Single-solution-based metaheuristics are usually only suitable for specific complex optimization problems because of their single particle scale and weak coordination capability. A heuristic based on simulated annealing (SA) is designed to solve the machine reassignment problem [2]. The threshold-accepting (TA) metaheuristic method is applied to solve the job shop scheduling problem of dehydration plants [3]. The microcanonical annealing (MA) algorithm is proposed for remote sensing image segmentation [4]. The tabu search (TS) metaheuristic is collaborated with a regenerator-reducing procedure to solve the regenerator location problem [5]. The guided local search (GLS) approach is introduced for multiuser detection in ultra-wideband systems [6]. The dynamically dimensioned search (DDS) is introduced for automatic calibration in watershed simulation models [7–11].

Compared with single-solution-based algorithms, the research and applications of population-based algorithms are more extensive because of the following three main advantages [11, 12]: more information can be obtained to guide the trial solutions toward a promising area within the search space by a set of trial solutions; local optimization can be effectively avoided because of the interaction of a set of trial solutions; and in terms of exploration ability, population-based heuristic algorithms are superior to single-solution-based heuristic algorithms. The genetic algorithm (GA) is used to address the characterization of hyperelastic materials [12, 13]. Particle swarm optimization (PSO) is attributed to improving and evaluating the performance of automated engineering design optimization [13, 14]. Differential evolution (DE) is presented for mobile robots to avoid obstacles [14, 15]. The dragonfly algorithm (DA) has been improved to train multilayer perceptrons [15, 16]. Shuffled complex evolution (SCE) is designed to optimize the load balancing of gateways in wireless sensor networks [16, 17]. The dolphin echolocation algorithm (DEA) is applied to design a steel frame structure [17, 18]. The bat algorithm (BA) is introduced to optimize the placement of a steel plate shear wall [18, 19], and the artificial bee colony (ABC) algorithm is presented to image steganalysis [19, 20]. The grey wolf optimizer (GWO) is adopted for parameter estimation in surface waves [20, 21].

The GWO is one of the most impressive swarm intelligence algorithms and is the only algorithm that is based on leadership hierarchy theory; it was introduced by Mirjalili et al. [22]. The GWO algorithm has three advantages [23, 24]: it has universal applicability to some real-life optimization problems; it is insensitive to derived information in the initial search; and it requires fewer algorithm parameters for adjustment. These features render it a simple, flexible, adaptable, usable, and stable algorithm [24, 25]. Therefore, since the GWO was proposed, researchers have conducted a considerable amount of in-depth research and applications. Regarding the improvements in research on the GWO algorithm, researchers tend to improve the performance of the GWO from four aspects [25]: position-updating mechanisms, new control parameters, encoding scheme of individuals, and population structure and hierarchy. Typical study cases are listed as follows: Mittal et al. [26] used the exponential decay function a to enhance the exploration process in the GWO. However, this algorithm suffers premature convergence. Kishor and Singh [27] proposed a modified version of the GWO by incorporating a simple crossover operator between two randomly selected individuals. However, this technique has low capacity in solving high-dimensional complex problems. A complex-valued encoding strategy was employed by Luo et al. [28] to substitute a typical real-valued strategy that was adopted in the standard GWO and propose a complex-encoded GWO. The main shortcoming of this method is that it suffers premature convergence. Yang et al. [29] used an effective cooperative hunting group and a random scout group strategy to propose a novel grouped grey wolf optimizer. This approach employs a complex mechanism. Xu et al. [30] proposed a chaotic dynamic weight grey wolf optimizer (CDGWO), in which a new position-updated equation, formed by employing a chaotic map and dynamic weight, was built to guide the search process for potential candidate solutions. Gupta and Deep [23] proposed a novel random walk grey wolf optimizer (RW-GWO); in the RW-GWO, the random walk strategy was used for improving the search ability of the GWO. However, it shows low solution accuracy. In addition, an improved grey wolf optimization (VW-GWO) algorithm based on variable weight strategies and the social hierarchy in the searching positions was presented by Gao and Zhao [31]. However, it employs a complex methodology. In terms of successful applications of the GWO, representative application research can be summarized as flow shop scheduling [32], machine learning [33–36], economic load dispatch [37], robotics and path planning [38, 39], channel estimation in wireless communication systems [40], and other applications detailed in References [24, 25]. Theoretical and practical research has shown the potential of the GWO algorithm in real life. However, numerous studies and experimental results concluded that the optimization performance of the GWO algorithm needs improvement. Specifically, the trial solution diversity would be hampered by the three best wolves that were identified in the accumulative search [12]. Many metaheuristics, such as the GWO, can be easily trapped in the local optima when solving multimodal optimization problems, where multiple global optimum solutions exist [41] and the linear control parameter strategy is not the perfect design for balancing the exploration and exploitation. These drawbacks may lead to an undesirable optimal performance of the algorithm [27, 42]. In addition, existing research on the GWO algorithm does not discuss improvements in the performance of the GWO algorithm by considering the positional interaction information among the three leaders (i.e., the first three best wolves). In the actual hunting process, however, better predation efficiency can be obtained only when positional information is communicated among the three leaders. In this paper, the positional interaction information refers to the information communication among the three leaders in their predation process as reflected by the relative change in position. In addition to not considering the positional interaction information among the three leaders of the grey wolves, existing research does not explore other predation methods, such as spiral walking hunting, which may help hunting and increase the chance of jumping out of the local optima for the GWO algorithm. In summary, the GWO algorithm is a strong algorithm but suffers from the abovementioned shortcomings. Considering the drawbacks of the GWO algorithm, this paper decided to improve upon it.

Based on this analysis, this paper improves the GWO algorithm from the following three aspects: a hunting model is built based on the spiral walking, a position-updated equation is rebuilt based on the positional interaction information among the three leaders of grey wolves, and a nonlinear control parameter is designed to replace the linear control parameter of the standard GWO algorithm. The proposed algorithm is tested on 23 classical benchmark problems, CEC2014 suite, and three well-known engineering optimization problems. The experimental results reveal that the proposed method is robust, efficient, and superior compared to other algorithms.

The remaining sections of the paper are organized as follows: The original GWO algorithm and DDS are briefly overviewed in Section 2. In Section 3, the dynamically dimensioned search grey wolf optimizer, which is based on the deep search strategy (DGWOD), is proposed. The principle of searching the GWO by the DDS is detailed, and the position-updated equations, which are based on the deep search strategy and the nonlinear control parameter equation, are constructed. Section 4 provides the experimental results and a discussion of a set of well-known test functions. This paper is concluded and future research directions are presented in Section 5.

2. Overview of GWO and DDS

2.1. Standard GWO Algorithm. In this section, four parts of the basic GWO algorithm [22] inspired by the complete process of hunting the grey wolf for preying are described.

2.1.1. Foundation of the Social Hierarchy. In the GWO algorithm, the search is executed by the joint guidance of the first three best grey wolves (i.e., α, β, and δ), and the positions of the ω grey wolves (solutions) are constantly adjusted with the guidance of the first three best grey wolves with an increase in the iteration number.

2.1.2. Encircling Prey. Grey wolves hunt their prey by encircling them, which is considered to be wise behavior. To describe the principle of this predation from the perspective of a mathematical model, Mirjalili et al. [22] constructed the following equation:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_{\mathrm{p}}(t) - \vec{X}(t)\right|, \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_{\mathrm{p}}(t) - \vec{A} \cdot \vec{D}, \tag{2}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2, \tag{4}$$

where t is the current number of iterations, $\vec{X}(t+1)$ is the position vector of the grey wolf at the $(t+1)$ th iteration, the symbol "·" indicates dot product, $\vec{X}_{\mathrm{p}}(t)$ is the position vector of the prey at the (t) th iteration, $\vec{D}$ is a vector that is relative to the position of the prey $\vec{X}_{\mathrm{p}}(t)$, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\vec{a}$ is a vector whose values are linearly decreased from 2 to 0 over iterations, and $\vec{r}_1$ and $\vec{r}_2$ are randomly generated vectors whose values lie between 0 and 1.

2.1.3. Hunting. As described in Section 2.1.2, the action of the grey wolf that encircles its prey provides the leader of the grey wolf group with necessary position information and forces the prey into promising areas. After the leader of the grey wolf group receives the position information about the prey, the next step is to guide omega (ω) wolves to conduct the hunting. To describe the hunting behaviors of grey wolves from the perspective of a mathematical model, we assumed that $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ represent the positions of the α wolf, β wolf, and δ wolf, respectively. Therefore, the mathematical models for grey wolf hunting are described as follows:

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha, \tag{5}$$

$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta, \tag{6}$$

$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta, \tag{7}$$

$$\vec{X}(t+1) = \frac{1}{3}\left(\vec{X}_1 + \vec{X}_2 + \vec{X}_3\right). \tag{8}$$

$\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are calculated using equation (1) as follows:

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)\right|, \tag{9}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)\right|, \tag{10}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)\right|. \tag{11}$$

2.1.4. Attacking Prey. In the GWO algorithm, the behaviors of grey wolves that attack their prey are controlled by constant changes in the value of the linear control parameter $\vec{a}$. According to equation (3), the expression of the vector $\vec{A}$ is correlated with the parameter $\vec{a}$. When the value of a linearly decreases from 2 to 0, the value of the vector $\vec{A}$ also decreases. When $|\vec{A}| \leq 1$, the hunting of a grey wolf will occur at any position between its current position and that of its prey. When $|\vec{A}| > 1$, wolves will search the entire solution space to locate the prey (optima). Therefore, $|\vec{A}|$ represents a controlling parameter vector that causes exploration and exploitation. We determine that different values of the control parameter $\vec{a}$ have a different role in the exploration and exploitation of the GWO algorithm. According to Reference [42], a larger $\vec{a}$ is favorable for global exploration, while a smaller $\vec{a}$ facilitates local exploitation. Therefore, the control parameter $\vec{a}$ has an important role in balancing the exploration and exploitation of the GWO algorithm. However, for the standard GWO algorithm, several studies have shown that the value of the control parameter $\vec{a}$ linearly changes and the design of the position-updated equation will cause some drawbacks, such as premature convergence of the algorithm and powerlessness when solving multimodal problems [12, 27, 42].

Based on this description, the pseudocode of the GWO algorithm is shown in Algorithm 1.

2.2. DDS Algorithm. The DDS algorithm is a powerful single-solution-based metaheuristic algorithm, which was employed for calibration problems that arise in watershed simulation models. DDS was developed by Tolson and Shoemaker in 2007 [7] and was proposed for optimization problems that are bound constrained. Thus, achieving excellent optimization results for bounded constrained global

```
Input: population size $N$ and the maximum number of iterations MaxIter
Output: optimal individual position $\overrightarrow{X}_\alpha$ and best fitness values $f(\overrightarrow{X}_\alpha)$
Randomly initialize $N$ individuals' position to construct a population
Calculate the fitness value of each individual and find $\alpha$, $\beta$, and $\delta$
while $t \leq$ MaxIter or stopping criteria not met do
    for each individual do
        Update current individuals' position according to equation (8)
    end
    Update $\overrightarrow{a}$, $\overrightarrow{A}_i$, and $\overrightarrow{C}_i$, $i = 1, 2, 3$
    Evaluate the fitness value of each individual
    Update $\alpha$, $\beta$, and $\delta$
end while
```

Algorithm 1: Pseudocode of the GWO algorithm.

optimization problems is the advantage of the DDS algorithm.

DDS is a point-to-point stochastic-based heuristic global search algorithm with no parameter tuning; global solutions are obtained by scaling within a user-specified maximum number of function evaluations (MaxIter) [43]. Since it is a simple model that is easily programmed and a global search algorithm, many researchers have focused their great attention on it. At the beginning, when the number of iterations is small, the global search of the algorithm is dominant. As the number of iterations approached the maximum, the algorithm evolved into a local search. The key idea for the DDS algorithm to transit from a global search to a local search is to dynamically and probabilistically reducing the number of dimensions to be perturbed in the neighborhood of the current best solution [11, 43]. The operation to dynamically and probabilistically reduce the number of dimensions to be perturbed can be summarized as follows: in each iteration, the jth variable is randomly selected with the probability P_t from m decision variables for inclusion in the neighborhood I_{perturb}. The probability P_t is expressed as

$$P_t = 1 - \frac{\ln(t)}{\ln(\text{MaxIter})}, \tag{12}$$

where t indicates the current iteration and MaxIter represents the maximum number of iterations.

At each iteration t, a new potential solution $\overrightarrow{X}^{\text{new}}(t)$ is obtained by perturbing the current best $\overrightarrow{X}^{\text{best}}(t)$ in randomly selected dimensions. These perturbation magnitudes are sampled using the standard normal random variable $N(0, 1)$ and reflecting decision variable bounds as [11]

$$\overrightarrow{X}_j^{\text{new}}(t) = \begin{cases} \overrightarrow{X}_j^{\text{best}}(t) + r \times \overrightarrow{\mu}_j \times \left(\text{ub}_j - \text{lb}_j\right), & \text{if } j \in I_{\text{perturb}}, \\ \overrightarrow{X}_j^{\text{best}}(t), & \text{otherwise}, \end{cases} \tag{13}$$

where $j = 1, 2, \ldots, m$; r is a scalar neighborhood size perturbation factor; $\overrightarrow{\mu}_j$ is a random vector that is generated for the jth variable to be perturbed; ub_j and lb_j correspond to the upper bound and lower bound of the jth variable; and $\overrightarrow{X}_j^{\text{new}}(t)$ and $\overrightarrow{X}_j^{\text{best}}(t)$ denote the jth variable of the trial potential solution and the current best solution, respectively.

To accurately choose the best solution between the current best $\overrightarrow{X}^{\text{best}}(t)$ and the trial potential $\overrightarrow{X}^{\text{new}}(t)$ for the next iteration, the greedy search method is employed. The current best solution $\overrightarrow{X}^{\text{best}}(t)$ will be replaced by the trial solution $\overrightarrow{X}^{\text{new}}(t)$ if the objective function value of $\overrightarrow{X}^{\text{new}}(t)$ is smaller than that of $\overrightarrow{X}^{\text{best}}(t)$, i.e., $f(\overrightarrow{X}^{\text{new}}(t)) < f(\overrightarrow{X}^{\text{best}}(t))$; otherwise, the current best solution $\overrightarrow{X}^{\text{new}}(t)$ is reserved for the next iteration. The pseudocode description of the DDS algorithm is presented in Algorithm 2 [11].

3. Proposed Algorithm

As presented in the previous sections, the GWO algorithm encounters a few drawbacks, such as premature convergence and a low capability to handle the difficulties of a multimodal search landscape [25]. To overcome these weaknesses, the most effective improvement is to increase the diversity of candidate solutions and further improve the balance between exploration and exploitation during the iteration. In terms of increasing the diversity of candidate solutions, inspired by the core idea of the DDS algorithm, this study adopts two improved strategies to increase the performance of the GWO algorithm. One approach is to dynamically and probabilistically reduce the number of dimensions to be perturbed in the neighborhood, which enables candidate solutions to be perturbed between the current solutions and each of the three best solutions. Another method is to use the positional interaction information about the first three best grey wolf individuals (i.e., α, β, and δ) in the process of encircling and preying on the prey to perform a deep search. The position-updated equation of the GWO algorithm, which is based on the dynamically dimensioned perturbation and position interaction information, is proposed, which is referred to as the DGWO. To balance the exploration and exploitation of the GWO algorithm, the introduction of the search mechanism of the DDS algorithm enables the GWO algorithm to gradually transform from a

```
Input: scalar neighborhood size perturbation factor $r = 0.2$, maximum number of iterations MaxIter, number of variables
    (dimension) $m$, and upper bounds ub and lower bounds lb
Output: $\overrightarrow{X}^{\text{best}}$ and $f^{\text{best}}$
Initialization
    $\overrightarrow{X}^{0} = [x_1^0, x_2^0, \ldots, x_m^0]$, $x_i^0 = \text{lb}_j + \text{rand} \cdot (\text{ub}_j - \text{lb}_j)$
    Set $t = 1$, $\overrightarrow{X}^{\text{best}} = X^0$, $f^{\text{best}} = f(\overrightarrow{X}^{\text{best}})$, $I_{\text{perturb}} = [0, 0, \ldots, 0]$
while $t \leq$ MaxIter do
    Compute the probability of perturbing the decision variables $P_t$ using equation (12)
    for $j = 1$ to $m$ do
        Generate uniform random numbers, rand$(j)$
        if rand$(j) < P_t$ then
            Set $I_{\text{perturb}}(j) = 1$
        end if
    end for
        Generate a standard normal random vector, $\overrightarrow{\mu}$
        for $j = 1$ to $m$ do
            $\overrightarrow{X}_j^{\text{new}} = \overrightarrow{X}_j^{\text{best}} + (\text{ub}_j - \text{lb}_j) \cdot I_{\text{perturb}}(j) \cdot r \cdot \overrightarrow{\mu}$ //equation (13)
    end for
    for $j = 1$ to $m$ do
            if $\overrightarrow{X}_j^{\text{new}} < \text{lb}_j$ then
                Set $\overrightarrow{X}_j^{\text{new}} = \text{lb}_j + (\text{lb}_j - \overrightarrow{X}_j^{\text{new}})$
                if $\overrightarrow{X}_j^{\text{new}} > \text{ub}_j$ then
                    Set $\overrightarrow{X}_j^{\text{new}} = \text{lb}_j$
                end if
            end if
            if $\overrightarrow{X}_j^{\text{new}} > \text{ub}_j$ then
                Set $\overrightarrow{X}_j^{\text{new}} = \text{ub}_j - (\overrightarrow{X}_j^{\text{new}} - \text{ub}_j)$
                if $\overrightarrow{X}_j^{\text{new}} < \text{lb}_j$ then
                    Set $\overrightarrow{X}_j^{\text{new}} = \text{ub}_j$
                end if
            end if
        end for
    Evaluate $f(\overrightarrow{X}^{\text{new}})$
    if $f(\overrightarrow{X}^{\text{new}}) < f^{\text{best}}$ then
        Set $\overrightarrow{X}^{\text{best}} = \overrightarrow{X}^{\text{new}}$, $f^{\text{best}} = f(\overrightarrow{X}^{\text{new}})$
    end if
    Set $t = t + 1$
end while
```

Algorithm 2: Pseudocode of the DDS algorithm.

global search to a local search with an increase in the number of iterations. The GWO algorithm has a strong exploration ability in the initial search stage and a strong exploitation ability in the subsequent stage of iteration. The nonlinear control parameter $\overrightarrow{a}'$ is proposed to replace the linear control parameter $\overrightarrow{a}$ of the standard GWO algorithm. This nonlinear control parameter strategy produces a GWO algorithm with a strong exploitation ability in the early stage of searching and a strong exploration ability in the subsequent stage of searching. Therefore, the introduction of the DDS and the nonlinear control parameter strategy strengthens the balance between exploration and exploitation of the GWO algorithm, and positional interaction information is utilized to conduct in-depth search and ensure the diversity of the candidate solution.

3.1. Two Ways to Hunt Prey Are Freely Switched Using DDS. As described in Section 2.1.3, a grey wolf hunts by direct encirclement. However, though actual observation, we found that, in addition to the previously mentioned hunting strategy, the grey wolf also approaches its prey by spiral walking when hunting. This way of spiral walking around the prey is often considered to be a very effective way to hunt [44]. Although we have determined that a grey wolf hunts by direct encirclement and spiral walking, a reasonable conversion of these two hunting methods has not been performed via research. The traditional method is to randomly convert between those two hunting methods by equal probability [44]. In an actual situation, the conversion probability between these two methods is not equal. A reasonable conversion method is that the grey wolf can freely switch between these two hunting methods during its predation process. Thus, the grey wolf has the best hunting effect, that is, to ensure that the grey wolf achieves the best prey (global optimum) in the best situation or obtains the relatively better prey (global approximation solution) in poorer conditions. We determined that the DDS method employs the conversion technique that we expected. According to Section 2.2, the core principle of the DDS algorithm is to transit the search from global to local by dynamically and probabilistically reducing the number of dimensions to be perturbed in the neighborhood of the current best solution, which causes the DDS to converge to the desired region to locate the global optimum in the best case or a reasonable local optimum in the worst case. Based on this analysis, the DDS method is introduced in the GWO algorithm to conduct free switching of the hunting behavior between direct encirclement and spiral walking to improve the quality of the solution of the GWO algorithm. The implementation steps are described as follows:

(i) First, at each iteration t, $\overrightarrow{D}_\alpha$, $\overrightarrow{D}_\beta$, and $\overrightarrow{D}_\delta$ are calculated using equations (9)–(11).

(ii) Second, at each iteration t, $\overrightarrow{D}_\alpha^{\text{new}}$, $\overrightarrow{D}_\beta^{\text{new}}$, and $\overrightarrow{D}_\delta^{\text{new}}$ are computed using the following equations:

$$\overrightarrow{d}_\alpha = \left|\overrightarrow{X}_\alpha(t) - \overrightarrow{r}_3 \cdot \overrightarrow{X}(t)\right|, \tag{14}$$

$$\overrightarrow{d}_\beta = \left|\overrightarrow{X}_\beta(t) - \overrightarrow{r}_4 \cdot \overrightarrow{X}(t)\right|, \tag{15}$$

$$\overrightarrow{d}_\delta = \left|\overrightarrow{X}_\delta(t) - \overrightarrow{r}_5 \cdot \overrightarrow{X}(t)\right|, \tag{16}$$

$$\overrightarrow{D}_\alpha^{\text{new}} = \overrightarrow{X}(t) + r \cdot \overrightarrow{d}_\alpha \cdot \exp\left(-2\pi \cdot \overrightarrow{d}_\alpha\right) \cdot \cos\left(\pi \cdot \overrightarrow{d}_\alpha\right), \tag{17}$$

$$\overrightarrow{D}_\beta^{\text{new}} = \overrightarrow{X}(t) + r \cdot \overrightarrow{d}_\beta \cdot \exp\left(-2\pi \cdot \overrightarrow{d}_\beta\right) \cdot \cos\left(\pi \cdot \overrightarrow{d}_\beta\right), \tag{18}$$

$$\overrightarrow{D}_\delta^{\text{new}} = \overrightarrow{X}(t) + r \cdot \overrightarrow{d}_\delta \cdot \exp\left(-2\pi \cdot \overrightarrow{d}_\delta\right) \cdot \cos\left(\pi \cdot \overrightarrow{d}_\delta\right). \tag{19}$$

(iii) Finally, using the ideas of the DDS algorithm to transition the search from global to local, $\overrightarrow{D}_\alpha$, $\overrightarrow{D}_\beta$, and $\overrightarrow{D}_\delta$ are recalculated using the following equations:

$$\overrightarrow{D}_{\alpha,j} = \begin{cases} \overrightarrow{D}_{\alpha,j}^{\text{new}}, & \text{if } P_t > \text{rand}(j), \\ \overrightarrow{D}_{\alpha,j}, & \text{otherwise}, \end{cases} \tag{20}$$

$$\overrightarrow{D}_{\beta,j} = \begin{cases} \overrightarrow{D}_{\beta,j}^{\text{new}}, & \text{if } P_t > \text{rand}(j), \\ \overrightarrow{D}_{\beta,j}, & \text{otherwise}, \end{cases} \tag{21}$$

$$\overrightarrow{D}_{\delta,j} = \begin{cases} \overrightarrow{D}_{\delta,j}^{\text{new}}, & \text{if } P_t > \text{rand}(j), \\ \overrightarrow{D}_{\delta,j}, & \text{otherwise}, \end{cases} \tag{22}$$

where $\overrightarrow{r}_3$, $\overrightarrow{r}_4$, and $\overrightarrow{r}_5$ are random vectors between 0 and 1 and r is a scalar neighborhood size perturbation factor, whose value is 0.2 in this paper.

3.2. Position-Updated Equation Based on the Positional Interaction Information.

As described in the original literature [22] of the GWO algorithm, the alpha (α) wolf is the supreme leader of the grey wolf pack and is primarily responsible for commanding all wolves to hunt, sleep, and wake. The leader in the second tier is referred to as the beta (β) wolf, which is controlled by α and is responsible for commanding the remaining wolves. The third tier of leadership entails the delta (δ) wolf, who has to submit to α and β but dominate the ω wolf. ω is the common wolf and has the subordinate role of listening to the orders of the first three leaders. This top-down leadership mechanism of the grey wolf pack enables the GWO algorithm to have strong exploration ability. As previously described, the cooperative hunting behavior of the grey wolf group is outstanding. One situation is that the first three best grey wolves (leaders) directly lead the ω wolf to hunt. In another situation, the α wolf commands the β wolf and the δ wolf to hunt and the β wolf commands the δ wolf to hunt. The leadership relationship between these three leaders usually occurs via the relative position changes, that is, positional interaction information. In the standard GWO algorithm, however, only the former case is considered and the latter case is disregarded, which is very important for the hunting of the grey wolf group. Based on this shortcoming of the standard GWO algorithm, we design a position-updated equation that is based on positional interaction information as

$$\overrightarrow{X}_{11} = \overrightarrow{X}_\alpha(t) - \overrightarrow{r}_6 \cdot \overrightarrow{A}_1 \cdot \overrightarrow{X}_\beta(t), \tag{23}$$

$$\overrightarrow{X}_{22} = \overrightarrow{X}_\beta(t) - \overrightarrow{r}_7 \cdot \overrightarrow{A}_2 \cdot \overrightarrow{X}_\delta(t), \tag{24}$$

$$\overrightarrow{X}_{33} = \overrightarrow{X}_\alpha(t) - \overrightarrow{r}_8 \cdot \overrightarrow{A}_3 \cdot \overrightarrow{X}_\delta(t), \tag{25}$$

$$\vec{X}_j(t+1) = \begin{cases} w_1 \cdot \frac{1}{3} \cdot \left(\vec{X}_{11,j} + \vec{X}_{22,j} + \vec{X}_{33,j}\right) + w_2 \cdot \frac{1}{3}\left(\vec{X}_{1,j} + \vec{X}_{2,j} + \vec{X}_{3,j}\right), & \text{if } P_n > \text{rand}(j), \\ w_2 \cdot \frac{1}{3} \cdot \left(\vec{X}_{11,j} + \vec{X}_{22,j} + \vec{X}_{33,j}\right) + w_1 \cdot \frac{1}{3}\left(\vec{X}_{1,j} + \vec{X}_{2,j} + \vec{X}_{3,j}\right), & \text{otherwise,} \end{cases} \tag{26}$$

where $\vec{X}_{11}$, $\vec{X}_{22}$, and $\vec{X}_{33}$ indicate the positional interaction information about the three leaders; $\vec{r}_6$, $\vec{r}_7$, and $\vec{r}_8$ are random vectors between 0 and 1; and w_1 and w_2 are the position weights.

3.3. Nonlinear Control Parameter Design. As previously presented, the DDS algorithm has a strong global search ability (i.e., strong exploration) at the initial search stage and a strong local search ability (i.e., exploitation) in the subsequent search stage. In addition, the exploration and exploitation of the GWO algorithm are primarily controlled by the control parameter $\vec{a}$. When $\vec{a}$ linearly decreases from 2 to 0, the algorithm exhibits strong exploration and weak exploitation in the initial stage of searching and exhibits strong exploitation and weak exploration in the subsequent stage. Since both the GWO algorithm and DDS algorithm have strong exploration in the initial iteration and strong exploitation in the subsequent iteration, introducing the search mechanism of the DDS algorithm into the GWO algorithm has further aggravated the imbalance between the exploration and the exploitation of the GWO algorithm. Therefore, an imbalance between exploration and exploitation occurs at different search stages of the algorithm. To address this problem, we designed the new nonlinear control parameter $\vec{a}'$, which nonlinearly increases from −2 to 2, to substitute the linear control parameter $\vec{a}$ of the standard GWO algorithm. In the initial search phase, because the population has a higher diversity, small $\vec{a}'$ values are needed to enhance the exploitation capability and accelerate convergence. In contrast, in the later stage of the search, since the diversity decreases in the population, larger $\vec{a}'$ values facilitate exploration and can help the search agents to be away from the local optimum. Therefore, the new nonlinear control parameter $\vec{a}'$ can ensure the relative strong exploitation at the initial iteration and a strong exploration in the subsequent iteration of the GWO algorithm. The nonlinear control parameter $\vec{a}'$ is designed as

$$\vec{a}' = 2 - 4 \cdot \exp\left(-\frac{t}{\text{MaxIter}}\right) \cdot \cos\left(-\frac{\pi}{2} \cdot \sqrt{\frac{t}{\text{MaxIter}}}\right), \tag{27}$$

where t and MaxIter indicate the current iteration and the maximum iteration number, respectively.

Figure 1 shows the transition between exploration and exploitation that was caused by the adaptive values of the control parameters $\vec{a}$ and $\vec{a}'$. As shown in Figure 1, with respect to the GWO algorithm, half of the iterations are adapted to exploration ($|\vec{A}| > 1$) and the rest of the iterations are devoted to exploitation ($|\vec{A}| \leq 1$). However, with regard to the DGWO algorithm, the number of iterations used for exploration and exploitation is 60.2% and 39.8%, respectively.

3.4. Framework and Pseudocode of the DGWO Algorithm. In this paper, the proposed hunt strategy of spiral walking is added to the GWO algorithm to enhance the ability of the predation. This strategy is freely switched between the original encirclement methods using the search mechanism of DDS. This method combines the proposed nonlinear control parameter strategy and the position-updated equation, which considers the positional interaction information, and develops the DGWO algorithm. The pseudocode of the proposed DGWO algorithm is shown in Algorithm 3.

3.5. Time Complexity of DGWO. The time complexities of the DGWO and GWO are summarized as follows:

(1) In the initialization phase, the DGWO and GWO require $O(N \times m)$ time, where N represents the population size and m represents the dimension of the problem

(2) Calculation of the control parameters of the DGWO and GWO requires $O(N \times m)$ time

(3) Update of the agents' position-updated equations of the DGWO and GWO requires $O(N \times m)$ time

(4) Evaluation of the fitness value of each agent requires $O(N \times m)$ time

Based on the above analysis, for each generation, the total time complexity is $O(N \times m)$, and given a maximum number of iterations, the total time complexity of the DGWO and GWO is $O(N \times m \times \text{MaxIter})$, where MaxIter indicates the maximum number of iterations.

3.6. Analysis and Comparison of the Diversity between GWO and DGWO. From equations (5) to (8), we can know that the grey wolves update their position under the leadership of the three best wolves. However, when the three fittest wolves get into local optimum, the whole search agents will all concentrate in this region, which leads to the decrease in diversity in the population, and the algorithm easily falls into local optimum. Based on this point, the DGWO algorithm is proposed to enhance the diversity of the GWO algorithm. To analyze and compare the diversity between the GWO and the DGWO, we choose the Sphere function as the benchmark test problem to see the difference in diversity between the GWO and the DGWO at different iterations. We set the population size as 30, the dimension of the problem as 2, and the upper and lower boundaries of

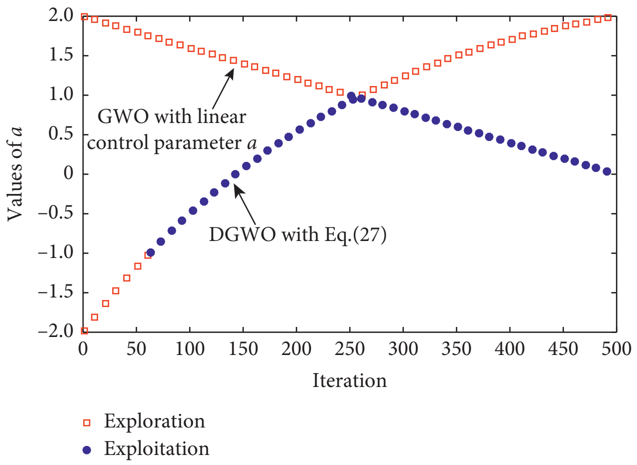


Figure 1: Updating the values of control parameters $\vec{a}$ and $\vec{a}'$ over the course of iterations.

```
Input: population size $N$, scalar neighborhood size perturbation factor $r$, maximum number of iterations MaxIter, number of
variables $m$, and upper bounds ub and lower bounds lb
Output: optimal individual position $\vec{X}_\alpha$ and best fitness value $f(\vec{X}_\alpha)$
Randomly initialize $N$ individuals' position $r$ to construct a population
Calculate the fitness value of each individual, find $\alpha$, $\beta$, and $\delta$, and set $t = 0$
while $t \leq$ MaxIter do
    Compute the probability ($P_n$) of perturbing the decision variables using equation (12) and the value of the nonlinear control
    parameter $\vec{a}'$ using equation (27)
    Generate uniform random numbers rand$(j) \in [0, 1]$
    for $i = 1$ to $N$ do
        for $j = 1$ to $m$ do
            if $P_n(t) < \text{rand}(j)$ then
                Calculate $\vec{D}_\alpha^{\text{new}}$, $\vec{D}_\beta^{\text{new}}$, and $\vec{D}_\delta^{\text{new}}$ according to equations (14)–(19)
                Calculate $\vec{X}_{11}$, $\vec{X}_{22}$, and $\vec{X}_{33}$ using equations (23)–(25)
                Calculate $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ using equations (5)–(7)
                Update current individuals' position according to equation (26)
            else
                Calculate $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ according to equations (9)–(11)
                Calculate $\vec{X}_{11}$, $\vec{X}_{22}$, and $\vec{X}_{33}$ using equations (23)–(25)
                Calculate $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ using equations (5)–(7)
                Update current individuals' position according to equation (26)
            end if
        end for
    end for
    Update $\vec{a}'$, $\vec{A}_i$, and $\vec{C}_i$, $i = 1, 2, 3$
    Evaluate the fitness value of each individual
    Update $\alpha$, $\beta$, and $\delta$
    Set $t = t + 1$
end while
```

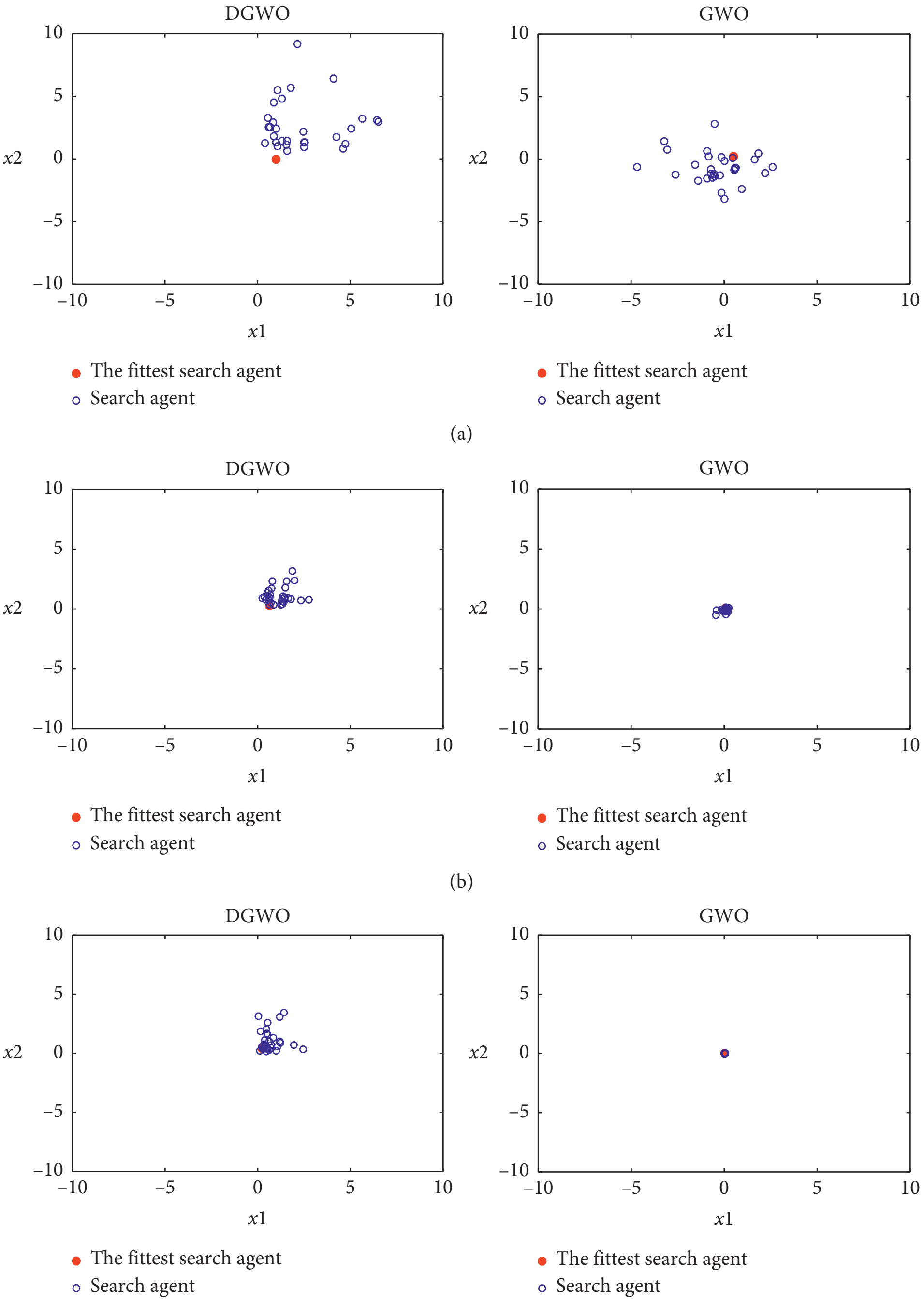
Algorithm 3: Pseudocode of the proposed DGWO algorithm.

the problem as 10 and −10, respectively. The diversity distributions of the DGWO and GWO at different iterations are plotted in Figure 2.

From Figure 2(a), when the number of iterations is 2, both the DGWO and GWO have high diversity individuals. However, from Figures 2(b) to 2(d), the DGWO algorithm shows better diversity of solutions than the GWO algorithm. This comparison confirms that the DGWO algorithm has higher diversity of solutions than the standard GWO algorithm.

4. Results and Discussion

4.1. Test Function Selection and Control Parameter Settings. In this section, to validate the performance of the proposed DGWO algorithm, 23 benchmark problems with various complexity and sizes are collected from studies [21, 23, 43]. The characteristics of the selected test functions are summarized in Table 1, where $f_{\min}$ denotes the global optimal value. In this table, the key to test functions, the mathematical expression of

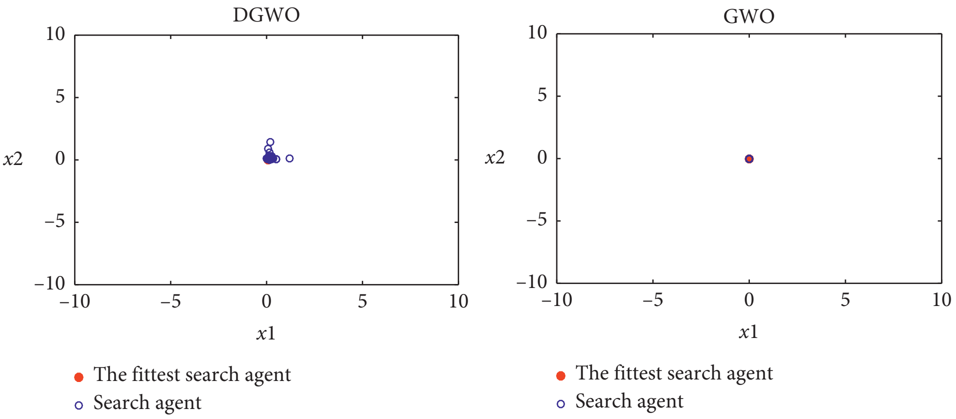


(a)

(b)

(c)

Figure 2: Continued.



(d)

FIGURE 2: Search agents' distribution observed at different iterations in the DGWO and GWO for solving Sphere function ($m = 2$). (a) Number of iterations $t = 2$. (b) Number of iterations $t = 4$. (c) Number of iterations $t = 6$. (d) Number of iterations $t = 12$.

TABLE 1: Description of 23 classic benchmark functions.

Key	Function	Range	m	C	$f_{\min}$
f_1	$\sum_{i=1}^{n} x_i^2$	[−100, 100]	10, 30, 50, 100	U	0
f_2	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[−10, 10]	10, 30, 50, 100	U	0
f_3	$\sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[−100, 100]	10, 30, 50, 100	U	0
f_4	$\max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[−100, 100]	10, 30, 50, 100	U	0
f_5	$\sum_{i=1}^{n-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−30, 30]	10, 30, 50, 100	U	0
f_6	$\sum_{i=1}^{n} ([x_i + 0.5])^2$	[−100, 100]	10, 30, 50, 100	U	0
f_7	$\sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	[−128, 128]	10, 30, 50, 100	U	0
f_8	$\sum_{i=1}^{n} - x_i \sin(\sqrt{\lvert x_i \rvert})$	[−500, 500]	10, 30, 50, 100	M	−418.9829 × D
f_9	$\sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[−5.12, 5.12]	10, 30, 50, 100	M	0
f_{10}	$-20 \exp(-0.2 \sqrt{1/n \sum_{i=1}^{n} x_i^2}) -$ $\exp(1/n \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	[−32, 32]	10, 30, 50, 100	M	0
f_{11}	$1/4000 \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	[−600, 600]	10, 30, 50, 100	M	0
f_{12}	$\pi/n \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})$ $+ (y_n - 1)^2]\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[−50, 50]	10, 30, 50, 100	M	0
f_{13}	$1/10\{\sin$ $2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] +$ $(x_n - 1)^2 [1 + \sin^2(2\pi x_n + 1)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[−50, 50]	10, 30, 50, 100	M	0
f_{14}	$(1/500 + \sum_{j=1}^{25} (1/j + \sum_{i=1}^{2} (x_i - a_{ij})^6))^{-1}$	[−65, 65]	2	F	1
f_{15}	$\sum_{i=1}^{11} [a_i - (x_i (b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4)]^2$	[−5, 5]	4	F	0.0003
f_{16}	$4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[−5, 5]	2	F	−1.0316
f_{17}	$(x_2 - (5.1/4\pi^2)x_1^2 +$ $(5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos x_1 + 10$	[−5, 5]	2	F	0.398
f_{18}	$[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 +$ $3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 -$ $32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	[−2, 2]	2	F	3
f_{19}	$-\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij} (x_j - p_{ij})^2)$	[1, 3]	3	F	−3.86
f_{20}	$-\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij} (x_j - p_{ij})^2)$	[0, 1]	6	F	−3.32
f_{21}	$-\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	F	−10.1532
f_{22}	$-\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	F	−10.4029
f_{23}	$-\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	F	−10.5363

m: dimension; C: category; U: unimodal; M: multimodal; F: fixed-dimension multimodal.

each benchmark test problem, the boundary of variables, the dimensions of the solution, and the category of each function are detailed. These test problems were divided into three categories: unimodal, multimodal, and fixed-dimension multimodal. In Table 1, $f_1 - f_7$ are unimodal problems that are used to benchmark the exploitation of algorithms because they have one global optimum and no local optima. Conversely, functions $f_8 - f_{23}$ are multimodal and fixed-dimension multimodal problems, which are helpful in examining exploration and local optima avoidance of algorithms, since they have a large number of local optima [26, 42].

The same control parameter settings for the GWO and DGWO algorithms are listed in Table 2, where "m" represents the dimension of the problem, "N" represents the size of the population, "MaxIter" represents the maximum number of iterations, "w_1" and "w_2" represent the position weight values, and "R" represents the independent simulation experiment times for each test problem. The proposed DGWO and standard GWO algorithms were coded in MATLAB R2015a. All simulation experiments were performed on a personal computer with Windows 10 64 bit professional OS and 4 GB RAM.

4.2. Impact of Position Weights w_1 and w_2.

As described in Section 3.2, the strategy of the modified position-updated equation (i.e., equation (26)) has an important role in balancing between exploration and exploitation in the evolution process. However, in equation (26), w_1 and w_2 are two crucial position weights for improving the optimization performance of the DGWO. In this section, to further investigate the impact of the position weight coefficients w_1 and w_2, several independent experiments were designed and conducted. We varied the values of w_1 and $w_2$and kept other algorithm's parameters fixed for all benchmark functions. Values $w_1 = 0.1$, $w_2 = 0.9$, $w_1 = 0.3$, $w_2 = 0.7$, $w_1 = 0.5$, $w_2 = 0.5$, $w_1 = 0.7$, $w_2 = 0.3$, and $w_1 = 0.9$, $w_2 = 0.1$ are selected to conduct experiments using 23 test functions. Among these test functions, the dimension of 13 test problems (f_1–f_{13}) is 30. All experimental results are reported in Table 3. "Mean" and "St. dev." values are two performance evaluation indexes shown in Table 3.

As shown in Table 3, the comprehensive optimization performance of the DGWO algorithm with position weights $w_1 = 0.1$ and $w_2 = 0.9$ is superior to that of other algorithms. Compared with the DGWO with $w_1 = 0.1$ and $w_2 = 0.9$, the proposed DGWO algorithm with position weights $w_1 = 0.3$ and $w_2 = 0.7$ achieved better and similar results for 9 functions (i.e., f_4–f_8, f_{12}–f_{14}, and f_{16}) and 3 functions (i.e., f_{11}, f_{17}, and f_{18}), respectively, and achieved worse results for 11 functions (i.e., f_1–f_3, f_9-f_{10}, f_{15}, and f_{19}–f_{23}). Compared with the DGWO with $w_1 = 0.1$ and $w_2 = 0.9$, the DGWO with $w_1 = 0.5$ and $w_2 = 0.5$ obtained better and similar results for 6 problems (i.e., f_5–f_7, f_{12}-f_{13}, and f_{16}) and 3 problems (i.e., f_{11} f_{14}, and f_{18}), respectively, and presented worse optimization results for 14 functions (i.e., f_1–f_4, f_8–f_{10}, f_{15}, f_{17}, and f_{19}–f_{23}). Compared with the DGWO with $w_1 = 0.1$ and $w_2 = 0.9$, the DGWO method with $w_1 = 0.7$ and $w_2 = 0.3$ attained better results for 6 functions (i.e., f_5, f_{12}–f_{14}, f_{16}, and f_{20}), obtained similar results for two functions (i.e., f_{11} and f_{18}), and obtained worse results for 15 functions (i.e., f_1–f_4, f_6–f_{10}, f_{15}, f_{17}, f_{19}, and f_{21}–f_{23}). Compared with the DGWO algorithm with $w_1 = 0.1$ and $w_2 = 0.9$, the DGWO with $w_1 = 0.9$ and $w_2 = 0.1$ achieved better optimization performance for 4 functions (f_5, f_{14}, f_{16}, and f_{20}), achieved similar results for one function (i.e., f_{18}), and achieved worse results for the remaining functions. Based on this analysis, the optimization performance of the DGWO worsens as the position weight value w_1 increases and w_2 decreases. Therefore, considering all w_1 and w_2 values, we concluded that setting the position weight values $w_1 = 0.1$ and $w_2 = 0.9$ for the DGWO algorithm was an ideal choice, and the position weights w_1 and w_2 of the DGWO algorithm were set as 0.1 and 0.9, respectively, in the next experiments.

The convergence curves of the average objective function values of the DGWO with different position weight values w_1 and w_2 for 10 typical benchmark functions are plotted in Figure 3.

4.3. Effectiveness Analysis of the Two Components in DGWO.

In the DGWO algorithm, two main components, namely, the modified position-updated equation and the nonlinear control parameter strategy, are proposed. To validate the effectiveness of these two components in improving the optimization performance of the DGWO, two experiments were conducted for 23 benchmark functions recorded in Table 1. Among those functions, the dimension of f_1–f_{13} is 30. The algorithm parameters are set the same as in Table 2. In the first experiment, the DGWO employed the modified position-updated equation (i.e., equation (26)), and the linear control parameter $\overrightarrow{a}$ that is similar to that in the study of Mirjalili et al. [22] is referred to as the DGWO-1. In the second experiment, the DGWO only used the nonlinear control parameter strategy (i.e., equation (27)), and the position-updated equation (8) is referred to as the DGWO-2. Two statistical criteria, "Mean" and "St. dev.," and the results of the DGWO-1, DGWO-2, and DGWO are shown in Table 4. Sign-rank sum tests at 0.05 and 0.1 significance levels were performed between the DGWO and each of DGWO-1 and DGWO-2.

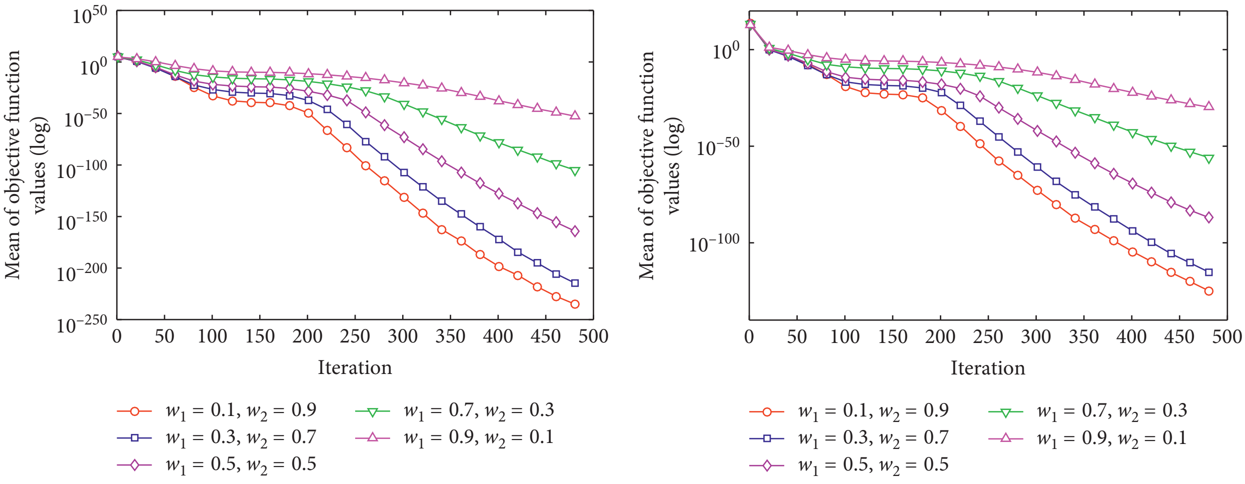
From Table 4, compared to the DGWO, the DGWO-1 achieved better results on 6 functions (i.e., f_4, f_6, f_8, f_{16}, f_{21}, and f_{22}), showed similar or approximate performance on 3 test functions (i.e., f_9, f_{10}, and f_{11}), and provided slightly poorer results on the rest of the test functions. It should be emphasized that the DGWO-1 could obtain very competitive optimization results compared to the DGWO, and its performance is not significantly inferior to that of the DGWO. We attribute the first experiment results to the fact that the modified position-updated equation has more advantages in balancing between exploration and exploitation and could ensure more potential solution diversity in the evolution process. Therefore, we can conclude that the performance differences between the DGWO and the DGWO-1 were not significant. From the results of the second experiment, it is found that the DGWO surpassed the DGWO-2 on 19 test functions and obtained similar results

Table 2: Experimental parameter settings for the GWO and DGWO.

Algorithm	Parameter					
	m	N	MaxIter	w_1	w_2	R
GWO	10, 30, 50, 100	30	500	—	—	30
DGWO	10, 30, 50, 100	30	500	0.1	0.9	30

Table 3: Experimental results of the DGWO using different position weight values w_1 and w_2 for 23 functions.

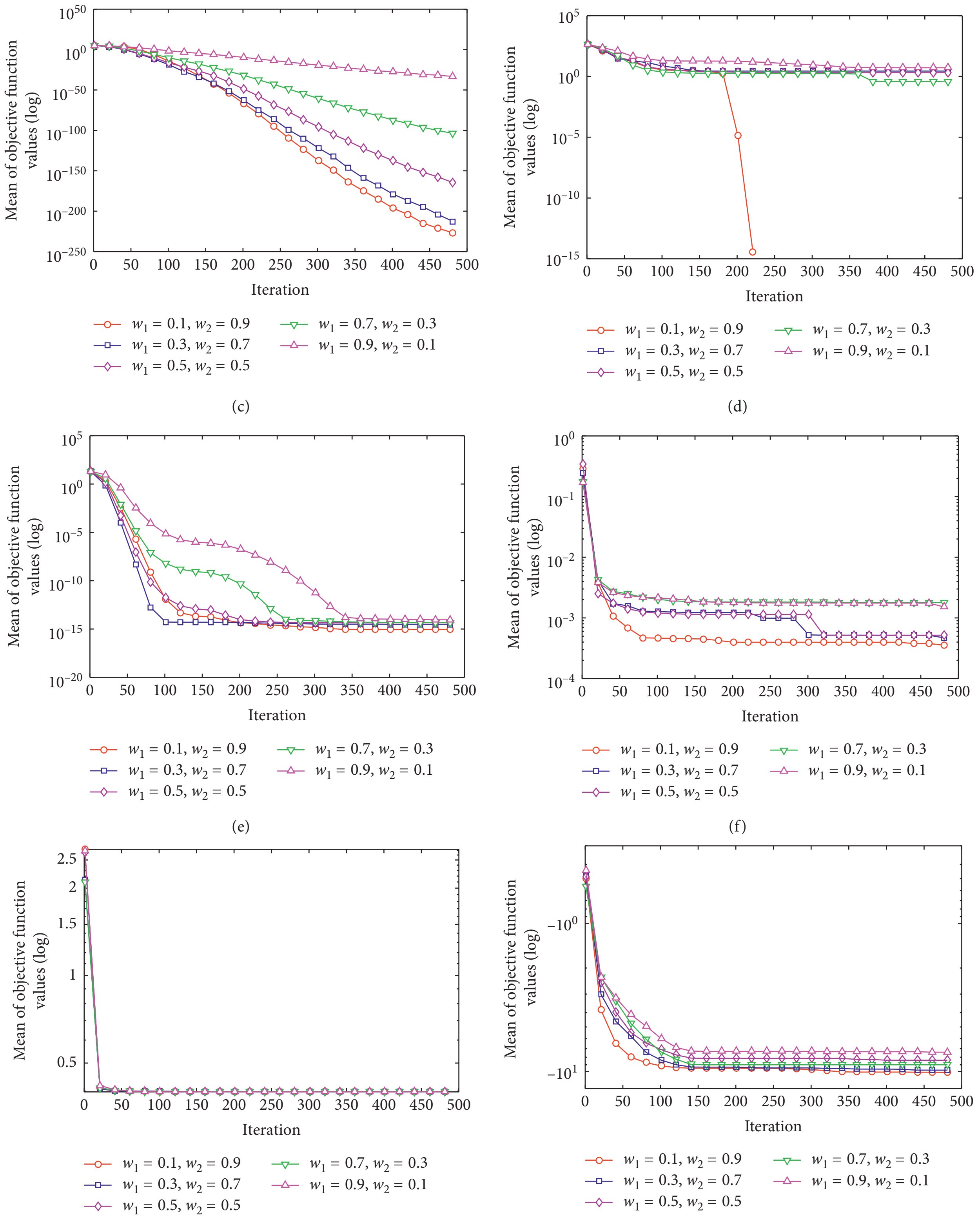
Function	$w_1 = 0.1$, $w_2 = 0.9$		$w_1 = 0.3$, $w_2 = 0.7$		$w_1 = 0.5$, $w_2 = 0.1$		$w_1 = 0.7$, $w_2 = 0.3$		$w_1 = 0.9$, $w_2 = 0.1$	
	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.
f_1	$2.38E-241$	$0.00E+00$	$2.70E-220$	$0.00E+00$	$8.78E-173$	$0.00E+00$	$1.64E-109$	$8.92E+109$	$3.62E-56$	$1.53E-55$
f_2	$1.39E-127$	$7.05E-127$	$2.10E-118$	$1.06E-117$	$2.38E-90$	$1.25E-89$	$1.05E-58$	$4.64E-58$	$3.81E-32$	$1.29E-31$
f_3	$9.45E-232$	$0.00E+00$	$9.94E-217$	$0.00E+00$	$1.71E-168$	$0.00E+00$	$2.40E-107$	$5.95E-105$	$5.61E-35$	$2.93E-34$
f_4	$3.11E-88$	$1.70E-87$	$2.69E-89$	$1.43E-88$	$7.19E-71$	$2.55E-70$	$4.14E-42$	$1.10E-41$	$4.80E-04$	$1.43E-03$
f_5	$2.85E+01$	$1.28E-01$	$2.82E+01$	$1.63E-01$	$2.79E+01$	$2.71E-01$	$2.64E+01$	$5.05E-01$	$2.59E+01$	$3.03E-01$
f_6	$9.70E-06$	$4.71E-06$	$7.73E-06$	$3.38E-06$	$8.52E-06$	$2.51E-06$	$2.19E-05$	$7.35E-06$	$1.84E-04$	$5.37E-05$
f_7	$1.66E-04$	$1.40E-04$	$1.26E-04$	$8.78E-05$	$1.30E-04$	$1.03E-04$	$1.69E-04$	$1.11E-04$	$5.48E-04$	$2.99E-04$
f_8	$-1.06E+04$	$2.43E+03$	$-1.16E+04$	$1.85E+03$	$-8.34E+03$	$2.63E+03$	$-4.81E+03$	$9.26E+02$	$-3.51E+03$	$4.46E+02$
f_9	$0.00E+000$	$0.00E+00$	$2.78E+00$	$7.09E+00$	$2.05E+00$	$4.23E+00$	$3.18E+00$	$1.28E+00$	$5.22E+00$	$1.72E+01$
f_{10}	$8.88E-16$	$0.00E+00$	$3.02E-15$	$1.77E-15$	$4.44E-15$	$0.00E+00$	$4.68E-15$	$9.01E-16$	$8.94E-15$	$2.63E-15$
f_{11}	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$2.35E-03$	$5.60E-03$
f_{12}	$1.79E-06$	$2.19E-06$	$5.43E-07$	$3.72E-07$	$6.73E-07$	$3.03E-07$	$1.48E-06$	$4.76E-07$	$1.22E-05$	$3.15E-06$
f_{13}	$2.49E-05$	$2.75E-05$	$7.63E-06$	$7.09E-06$	$9.13E-06$	$4.55E-06$	$2.18E-05$	$8.46E-06$	$1.58E-03$	$4.23E-03$
f_{14}	$2.94E+00$	$1.43E+00$	$2.55E+00$	$4.04E+00$	$2.94E+00$	$4.42E+00$	$2.17E+00$	$3.56E+00$	$2.64E+00$	$3.29E+00$
f_{15}	$3.56E-04$	$2.31E-04$	$4.70E-04$	$4.64E-04$	$5.18E-04$	$4.92E-04$	$1.79E-03$	$5.09E-03$	$1.24E-03$	$3.71E-03$
f_{16}	-1.0302	$5.17E-03$	-1.0313	$1.00E-03$	-1.0316	$9.56E-05$	-1.0316	$1.91E-05$	-1.0316	$3.03E-06$
f_{17}	0.3980	$2.11E-04$	0.3980	$6.02E-05$	0.3979	$7.36E-05$	0.3979	$3.39E-05$	0.3979	$1.16E-05$
f_{18}	3	$2.41E-04$	3	$1.07E-03$	3	$6.36E-04$	3	$2.71E-04$	3	$1.24E-04$
f_{19}	-3.8590	$5.87E-03$	-3.8501	$1.27E-02$	-3.8533	$8.33E-03$	-3.8577	$3.16E-03$	-3.8612	$1.09E-03$
f_{20}	-3.1725	$5.27E-02$	-3.1402	$7.59E-02$	-3.1681	$6.65E-02$	-3.2156	$7.58E-02$	-3.2394	$6.61E-02$
f_{21}	-10.0984	$1.56E-01$	-9.7434	$1.12E+00$	-8.3562	$2.40E+00$	-8.9645	$2.19E+00$	-7.3721	$2.90E+00$
f_{22}	-10.2969	$2.58E-01$	-9.8997	$1.00E+00$	-9.5178	$2.01E+00$	-10.0489	$1.34E+00$	-9.7407	$1.73E+00$
f_{23}	-10.5307	$2.94E-02$	-10.3556	$9.87E-01$	-10.1956	$1.30E+00$	-10.1751	$1.37E+00$	-10.3090	$1.01E+00$
−		13			15		16		18	
+		9			6		6		5	
≈		1			2		1		0	



(a)

(b)

Figure 3: Continued.



(c)

(d)

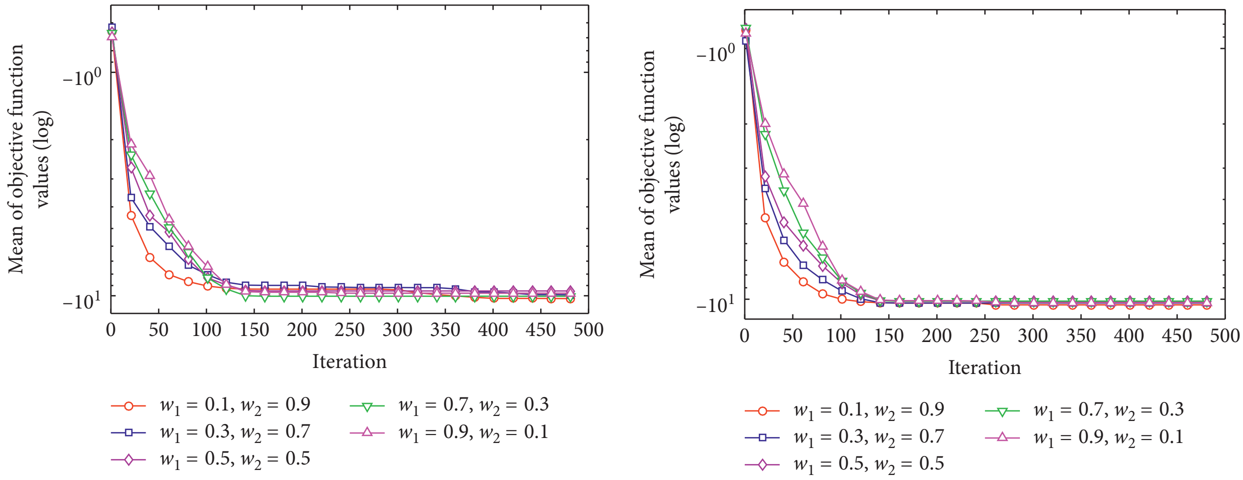
(e)

(f)

(g)

(h)

Figure 3: Continued.



(i)

(j)

FIGURE 3: Convergence curves of the DGWO with different weights w_1 and w_2 on 10 typical test functions. (a) f_1. (b) f_2. (c) f_3. (d) f_9. (e) f_{10}. (f) f_{15}. (g) f_{17}. (h) f_{21}. (i) f_{22}. (j) f_{23}.

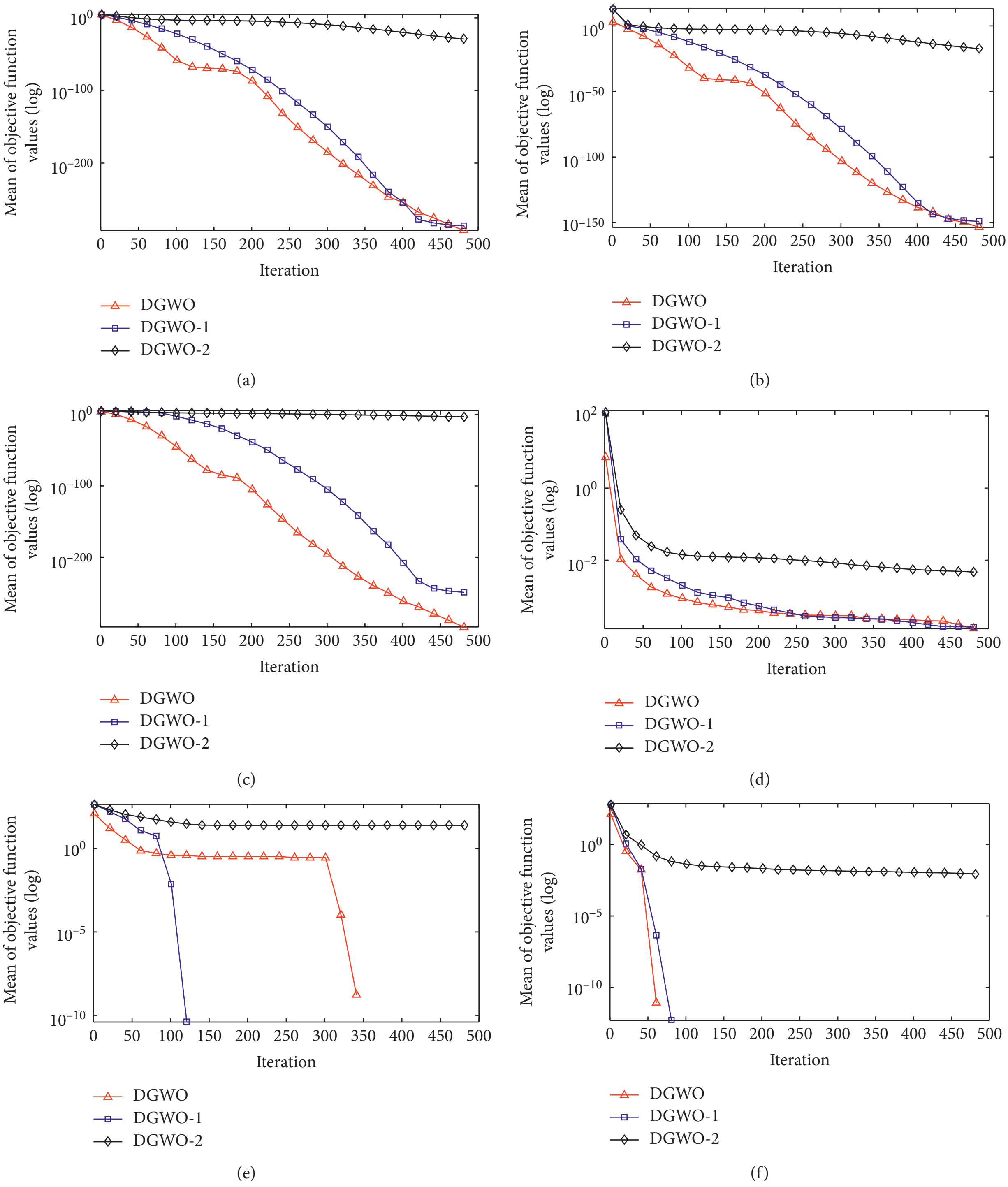
TABLE 4: Experimental comparison results of the DGWO, DGWO-1, and DGWO-2 on 23 functions ($m = 30$).

Function	DGWO-1 (Mean ± St. dev.)	DGWO-2 (Mean ± St. dev.)	DGWO (Mean ± St. dev.)
f_1	$1.43E-237 \pm 0.00E+00$	$8.73E-32 \pm 1.16E-31$	$2.38E-241 \pm 0.00E+00$
f_2	$5.60E-118 \pm 3.06E-149$	$3.84E-19 \pm 5.74E-19$	$1.39E-127 \pm 7.05E-127$
f_3	$6.40E-230 \pm 0.00E+00$	$2.74E-04 \pm 1.27E-03$	$9.45E-232 \pm 0.00E-00$
f_4	$1.10E-110 \pm 5.94E-110$	$3.34E-01 \pm 2.48E-01$	$3.11E-88 \pm 1.70E-87$
f_5	$2.85E+01 \pm 8.56E-02$	$2.84E+01 \pm 6.98E-01$	$2.85E+01 \pm 1.28E-01$
f_6	$7.66E-06 \pm 7.36E-06$	$2.42E+00 \pm 7.32E-01$	$9.70E-06 \pm 4.71E-06$
f_7	$1.74E-04 \pm 1.23E-04$	$4.35E-03 \pm 2.11E-03$	$1.66E-04 \pm 1.40E-04$
f_8	$-1.22E+04 \pm 1.08E+03$	$-5.97E+03 \pm 8.42E+02$	$-1.06E+04 \pm 2.43E+03$
f_9	$0.00E+00 \pm 0.00E+00$	$2.48E+01 \pm 1.24E+01$	$0.00E+00 \pm 0.00E+00$
f_{10}	$8.88E-16 \pm 0.00E+00$	$1.15E-14 \pm 3.73E-15$	$8.88E-16 \pm 0.00E+00$
f_{11}	$0.00E+00 \pm 0.00E+00$	$7.43E-03 \pm 1.11E-02$	$0.00E+00 \pm 0.00E+00$
f_{12}	$3.82E-06 \pm 4.75E-06$	$2.27E-01 \pm 1.85E-01$	$1.79E-06 \pm 2.19E-06$
f_{13}	$5.47E-05 \pm 5.82E-05$	$1.60E+00 \pm 4.21E-01$	$2.49E-05 \pm 2.75E-05$
f_{14}	$4.89E+00 \pm 5.60E+00$	$6.69E+00 \pm 5.01E+00$	$2.94E+00 \pm 4.43E+00$
f_{15}	$3.76E-04 \pm 3.04E-04$	$5.11E-03 \pm 8.56E-03$	$3.56E-04 \pm 2.31E-04$
f_{16}	$-1.0316 \pm 1.77E-05$	$-1.0316 \pm 2.83E-08$	$-1.0303 \pm 5.17E-03$
f_{17}	$0.3987 \pm 1.00E-03$	$0.3979 \pm 6.65E-07$	$0.3980 \pm 2.11E-04$
f_{18}	$3.0007 \pm 1.22E-03$	$3.0001 \pm 8.14E-05$	$3.0001 \pm 2.41E-04$
f_{19}	$-3.8566 \pm 5.02E-03$	$-3.8617 \pm 1.71E-03$	$-3.8590 \pm 5.87E-03$
f_{20}	$-3.12163 \pm 6.50E-02$	$-3.2543 \pm 8.34E-02$	$-3.1725 \pm 5.27E-02$
f_{21}	$-10.1433 \pm 1.22E-02$	$-8.1941 \pm 2.93E+00$	$-10.0984 \pm 1.56E-01$
f_{22}	$-10.3958 \pm 6.49E-03$	$-9.8783 \pm 1.73E+00$	$-10.2969 \pm 2.58E-01$
f_{23}	$-10.5221 \pm 2.25E-02$	$-10.0848 \pm 1.75E+00$	$-10.5334 \pm 2.94E-02$
Losses (−)	14	19	
Wins (+)	6	3	
Approximations (≈)	3	1	
Detected differences (α)	—	0.05	

on function f_{18}. To better understand this phenomenon, we need to know that the nonlinear control parameter strategy was specifically designed for the modified position-updated equation and is not suitable for independent use in the search process. Thus, the performance of the DGWO significantly outperformed that of the DGWO-2.

The convergence curves of the average objective function values of the DGWO, DGWO-1, and DGWO-2 on 10 typical test functions are plotted in Figure 4. From Table 4 and Figure 4, we can conclude that the two components of the DGWO are able to compensate for each other to improve the optimization performance of the GWO.

4.4. Performance Comparison with the Standard GWO Algorithm. We independently tested each problem 30 times



(a)

(b)

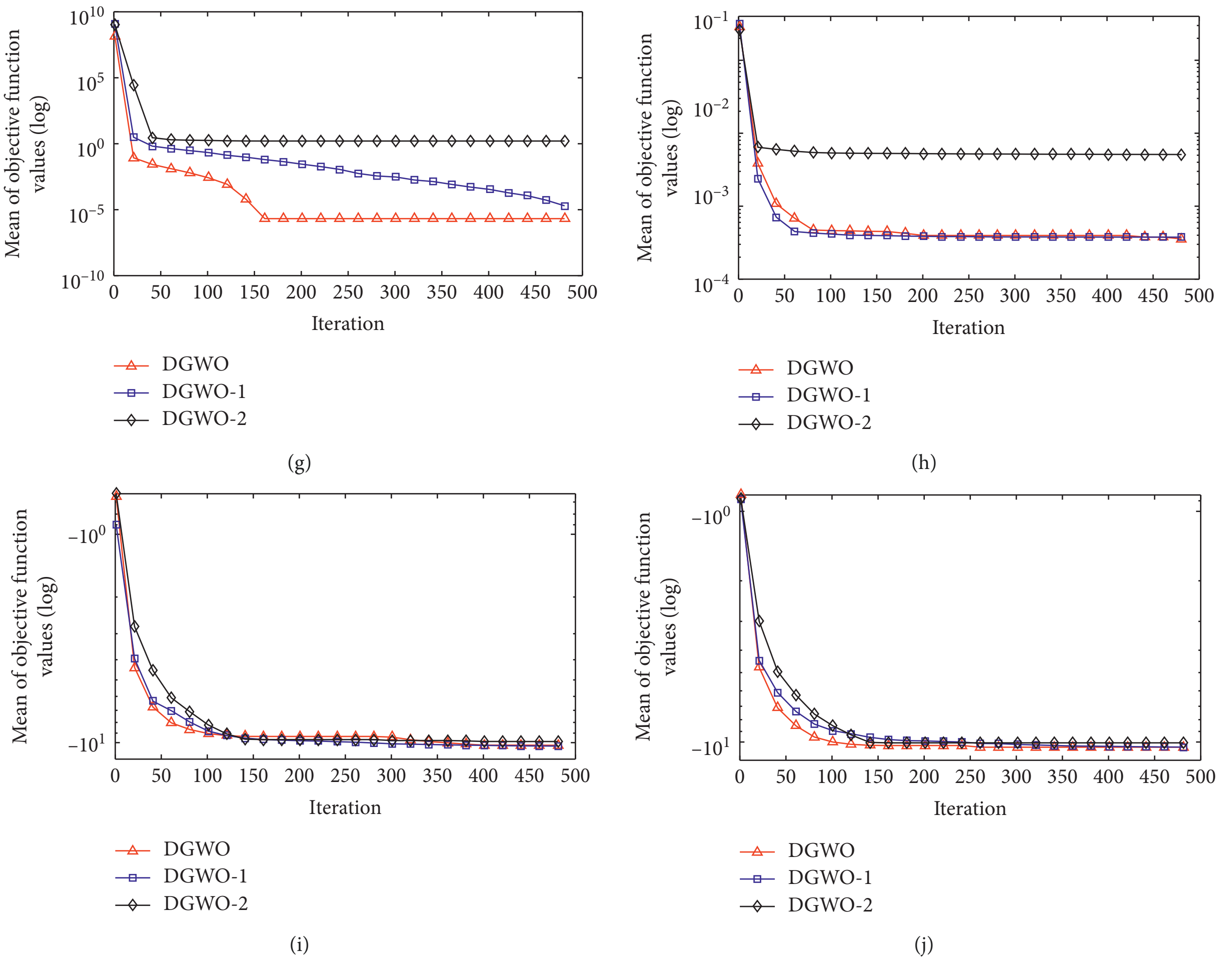
(c)

(d)

(e)

(f)

Figure 4: Continued.



(g)

(h)

(i)

(j)

FIGURE 4: Convergence curves of the DGWO, DGWO-1, and DGWO-2 on ten typical test functions. (a) f_1. (b) f_2. (c) f_3. (d) f_7. (e) f_9. (f) f_{11}. (g) f_{13}. (h) f_{15}. (i) f_{22}. (j) f_{23}.

to obtain four statistical criteria for comparing the performance of the algorithms; that is, “Best” indicates the best value, “Worst” represents the worst value, “Mean” denotes the average best values, and “St. dev.” indicates the standard deviation value. The simulation experimental results are described in Table 5.

As shown in Table 5, the DGWO has a better optimization performance for the seven unimodal benchmarks, with the exception of f_5 and compared with the standard versions of the GWO, since the DGWO provides the best “Best,” “Worst,” “Mean,” and “St. dev.” values for 6 of 7 unimodal benchmarks. For the six multimodal benchmarks (f_8–f_{13}) in Table 5, none of the standard versions of the GWO has a better optimization performance than the DGWO algorithm for all test problems using the “Mean” statistical criterion. As observed in Table 5, the DGWO algorithm achieved a better performance than the GWO for 5 fixed-dimension multimodal test functions (i.e., f_{14}, f_{20}–f_{23}) and provided slightly better results than the GWO for functions f_{16}, f_{18}, and f_{19}. For function f_{16}, however, the GWO obtained better results than the DGWO.

The percentage of problems solved by the GWO and DGWO is recorded in Table 6. It should be noted that when we use a certain algorithm to solve the 13 test functions (i.e., f_1–f_{13}) and 10 test functions (i.e., f_{14}–f_{23}) listed in Table 1, if the error between the actual value and its theoretical value is 10^{-5} and 10^{-3}, respectively, then this problem can be regarded as having been successfully solved. From Table 6, it can be seen that, for functions f_1, f_2, f_{10}, f_{16}, f_{18}, and f_{23}, the DGWO and GWO obtained the same percentage of solving problems. On 13 test problems (i.e., f_3, f_4, f_6–f_9, f_{11}–f_{15}, and f_{21}-f_{22}), the DGWO has shown a higher percentage than the GWO. However, the GWO has shown a higher percentage than the DGWO for functions f_{17}, f_{19}, and f_{20}.

To obtain an intuitive cognition of the convergence rate of the DGWO and GWO algorithms, Figure 5 shows the convergence curves of the DGWO and GWO for 12 typical test functions with $m = 10$, 30, 50, and 100. As shown in Figure 5, the DGWO algorithm achieved a faster convergence than the standard GWO algorithm for all 12 test problems. This finding verifies that the position-updated strategy and the nonlinear control parameter proposed in this paper can achieve faster search and excellent optimization performance of the DGWO algorithm for low- and high-dimensional problems.

Table 5: Results obtained by the DGWO and GWO algorithms on 23 test problems.

Key	m	GWO				DGWO			
		Best	Worst	Mean	St. dev.	Best	Worst	Mean	St. dev.
f_1	10	7.89E − 61	5.72E − 55	3.02E − 56	1.07E − 55	0.00E − 00	7.08E − 298	2.36E − 299	0.00E − 00
	30	1.48E − 29	4.31E − 27	8.83E − 28	1.09E − 27	1.63E − 287	5.44E − 240	2.38E − 241	0.00E − 00
	50	4.89E − 21	1.30E − 19	4.62E − 20	3.83E − 20	2.18E − 244	1.88E − 203	6.27E − 205	0.00E − 00
	100	1.97E − 13	2.88E − 12	1.05E − 12	6.38E − 13	1.43E − 183	1.01E − 163	8.19E − 169	0.00E − 00
f_2	10	7.44E − 35	2.11E − 32	3.62E − 33	4.99E − 33	7.09E − 190	4.34E − 154	1.45E − 155	7.92E − 155
	30	2.82E − 17	2.79E − 16	1.22E − 16	6.95E − 17	3.74E − 148	3.86E − 126	1.39E − 127	7.05E − 127
	50	8.15E − 13	5.92E − 12	2.64E − 12	1.31E − 12	4.61E − 132	5.38E − 109	1.80E − 110	9.81E − 110
	100	1.37E − 08	1.43E − 07	4.59E − 08	2.38E − 08	1.26E − 102	1.84E − 63	1.62E − 84	8.48E − 84
f_3	10	1.07E − 29	9.17E − 24	5.56E − 25	1.78E − 24	0.00E − 00	9.71E − 299	3.24E − 300	0.00E − 00
	30	5.46E − 11	3.20E − 04	2.06E − 05	5.95E − 05	1.67E − 260	2.83E − 230	9.45E − 232	0.00E − 00
	50	2.20E − 03	1.69E + 00	2.68E − 01	3.45E − 01	2.45E − 209	1.65E − 169	6.52E − 171	0.00E − 00
	100	4.99E + 01	1.80E + 03	5.60E + 02	5.34E + 02	7.91E − 144	3.02E − 18	1.01E − 19	5.52E − 19
f_4	10	6.43E − 20	2.03E − 17	2.64E − 18	4.31E − 18	1.06E − 150	3.41E − 129	1.25E − 130	6.22E − 130
	30	9.39E − 08	3.48E − 06	7.97E − 07	7.84E − 07	1.30E − 106	9.33E − 87	3.11E − 88	1.70E − 87
	50	4.84E − 05	1.55E − 03	3.28E − 04	3.51E − 04	9.12E − 88	1.21E − 76	8.25E − 78	2.36E − 77
	100	9.62E − 02	1.25E + 00	5.42E − 01	3.25E − 01	1.08E − 65	9.02E − 42	3.01E − 43	1.65E − 42
f_5	10	5.33E + 00	8.04E + 00	6.61E + 00	5.88E − 01	4.09E − 01	8.39E + 00	3.10E + 00	2.31E + 00
	30	2.55E + 01	2.87E + 01	2.69E + 01	7.47E + 01	2.82E + 01	2.87E + 01	2.85E + 01	1.28E − 01
	50	4.59E + 01	4.87E − 01	4.74E + 01	8.18E − 01	4.83E + 01	4.85E + 01	4.84E + 01	7.31E − 02
	100	9.61E + 01	9.85E + 01	9.77E + 01	7.81E − 01	9.80E + 01	9.80E + 01	9.80E + 01	9.99E − 03
f_6	10	9.48E − 07	2.53E − 01	2.47E − 02	7.55E − 02	2.96E − 08	4.66E − 06	5.84E − 07	8.37E − 07
	30	2.50E − 01	2.00E + 00	7.73E − 01	3.91E − 01	1.48E − 06	1.73E − 05	9.70E − 06	4.71E − 06
	50	1.25E + 00	3.70E + 00	2.58E + 00	6.23E + 00	9.62E − 06	2.05E − 04	5.16E − 05	4.87E − 05
	100	8.21E + 00	1.19E − 01	1.01E + 01	8.45E − 01	8.52E − 05	1.12E − 03	3.28E − 04	2.40E − 04
f_7	10	9.10E − 05	2.14E − 03	7.23E − 04	5.93E − 04	1.14E − 05	3.40E − 04	1.24E − 04	9.64E − 05
	30	8.25E − 04	3.97E − 03	1.98E − 03	9.59E − 04	2.88E − 05	6.20E − 04	1.66E − 04	1.40E − 04
	50	1.13E − 04	8.73E − 03	3.17E − 03	1.58E − 03	8.94E − 06	9.17E − 04	2.31E − 04	2.28E − 04
	100	3.02E − 03	2.18E − 02	8.20E − 03	4.02E − 03	2.51E − 06	1.93E − 03	2.62E − 04	3.68E − 04
f_8	10	−3.38E + 03	−2.05E + 03	−2.64E + 03	3.12E + 02	−4.19E + 03	−2.33E + 03	−3.85E + 03	5.85E + 02
	30	−7.81E + 03	−3.32E + 03	−6.02E + 03	9.48E + 02	−1.26E + 04	−5.66E + 03	−1.05E + 03	2.43E + 03
	50	−1.08E + 04	−4.16E + 03	−9.23E + 04	1.22E + 03	−2.10E + 04	−8.76E + 03	−1.59E + 04	3.94E + 03
	100	−1.99E + 04	−5.36E + 03	−1.57E + 04	3.05E + 03	−4.19E + 04	−1.56E + 04	−2.28E + 04	5.58E + 03
f_9	10	0.00E − 00	7.13E + 00	9.06E − 01	1.98E + 00	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	30	5.68E − 14	9.58E + 00	2.91E + 00	3.16E + 00	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	50	1.14E − 13	2.86E + 01	5.00E + 00	5.99E + 00	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	100	2.80E − 07	3.73E + 00	1.12E + 00	7.67E + 00	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
f_{10}	10	4.44E − 15	7.99E − 15	7.52E − 15	1.23E − 15	8.88E − 16	8.88E − 16	8.88E − 16	0.00E − 00
	30	6.84E − 14	1.47E − 13	1.07E − 13	1.45E − 14	8.88E − 16	8.88E − 16	8.88E − 16	0.00E − 00
	50	1.44E − 11	1.12E − 10	3.98E − 11	2.59E − 11	8.88E − 16	8.88E − 16	8.88E − 16	0.00E − 00
	100	4.91E − 08	1.96E − 07	1.10E − 07	4.08E − 08	8.88E − 16	8.88E − 16	8.88E − 16	0.00E − 00
f_{11}	10	0.00E − 00	9.22E − 02	1.96E − 02	2.00E − 02	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	30	0.00E − 00	5.14E − 02	5.48E − 03	1.13E − 02	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	50	0.00E − 00	3.17E − 02	2.59E − 03	7.27E − 03	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
	100	1.34E − 13	3.69E − 02	7.97E − 03	1.27E − 02	0.00E − 00	0.00E − 00	0.00E − 00	0.00E − 00
f_{12}	10	3.25E − 07	2.03E − 02	4.91E − 03	8.54E − 03	1.38E − 08	1.58E − 06	3.09E − 07	3.33E − 07
	30	1.96E − 02	1.05E − 01	4.80E − 02	2.31E − 02	1.34E − 07	8.29E − 06	1.79E − 06	2.19E − 06
	50	4.83E − 02	3.93E − 01	1.14E − 01	6.21E − 02	4.71E − 07	1.72E − 05	3.12E − 06	3.31E − 06
	100	1.87E − 01	4.65E − 01	2.93E − 01	6.46E − 02	4.04E − 06	7.82E − 05	1.61E − 05	1.43E − 05
f_{13}	10	2.99E − 06	1.01E − 01	9.94E − 03	3.03E − 02	3.31E − 08	9.79E − 06	2.09E − 06	2.46E − 06
	30	2.93E − 01	1.20E + 00	6.78E − 01	2.39E − 01	5.30E − 07	1.09E − 04	2.49E − 05	2.75E − 05
	50	1.34E + 00	2.93E + 00	2.15E + 00	4.24E − 01	1.75E − 05	4.30E − 04	1.12E − 04	8.81E − 05
	100	6.01E + 00	8.03E + 00	6.89E + 00	4.38E − 01	1.86E − 04	2.34E − 03	9.13E − 04	5.72E − 04
f_{14}	2	0.9980	1.08E + 01	3.2259	3.15E + 00	0.9980	1.27E + 00	2.9434	4.43E + 00
f_{15}	4	0.0003	2.04E − 02	0.0090	1.01E − 02	0.0003	1.58E − 03	0.0003	2.31E − 04
f_{16}	2	−1.0316	−1.0316	−1.0316	2.50E − 08	−1.0316	−1.0096	−1.0303	5.17E − 03
f_{17}	2	0.3979	0.3979	0.3979	1.22E − 06	0.3979	0.3990	0.3980	2.11E − 04
f_{18}	2	3.0000	3.0001	3.0000	2.58E − 05	3.0000	3.0013	3.0001	2.41E − 04
f_{19}	3	−3.8628	−3.8549	−3.8613	2.74E − 03	−3.8626	−3.8360	−3.8590	5.87E − 03
f_{20}	6	−3.3220	−2.8404	−3.2626	1.06E − 01	−3.2959	−3.1034	−3.1725	5.27E − 02

Table 5: Continued.

Key	m	GWO				DGWO			
		Best	Worst	Mean	St. dev.	Best	Worst	Mean	St. dev.
f_{21}	4	−10.1530	−5.0552	−9.4765	1.7505	−10.1532	−9.5340	−10.0984	$1.56E-01$
f_{22}	4	−10.4027	−5.0877	−10.2240	$9.70E-01$	−10.4028	−9.4206	−10.2969	$2.58E-01$
f_{23}	4	−10.5360	−2.4217	−10.2642	1.4812	−10.5364	−10.3753	−10.5334	$2.94E-02$

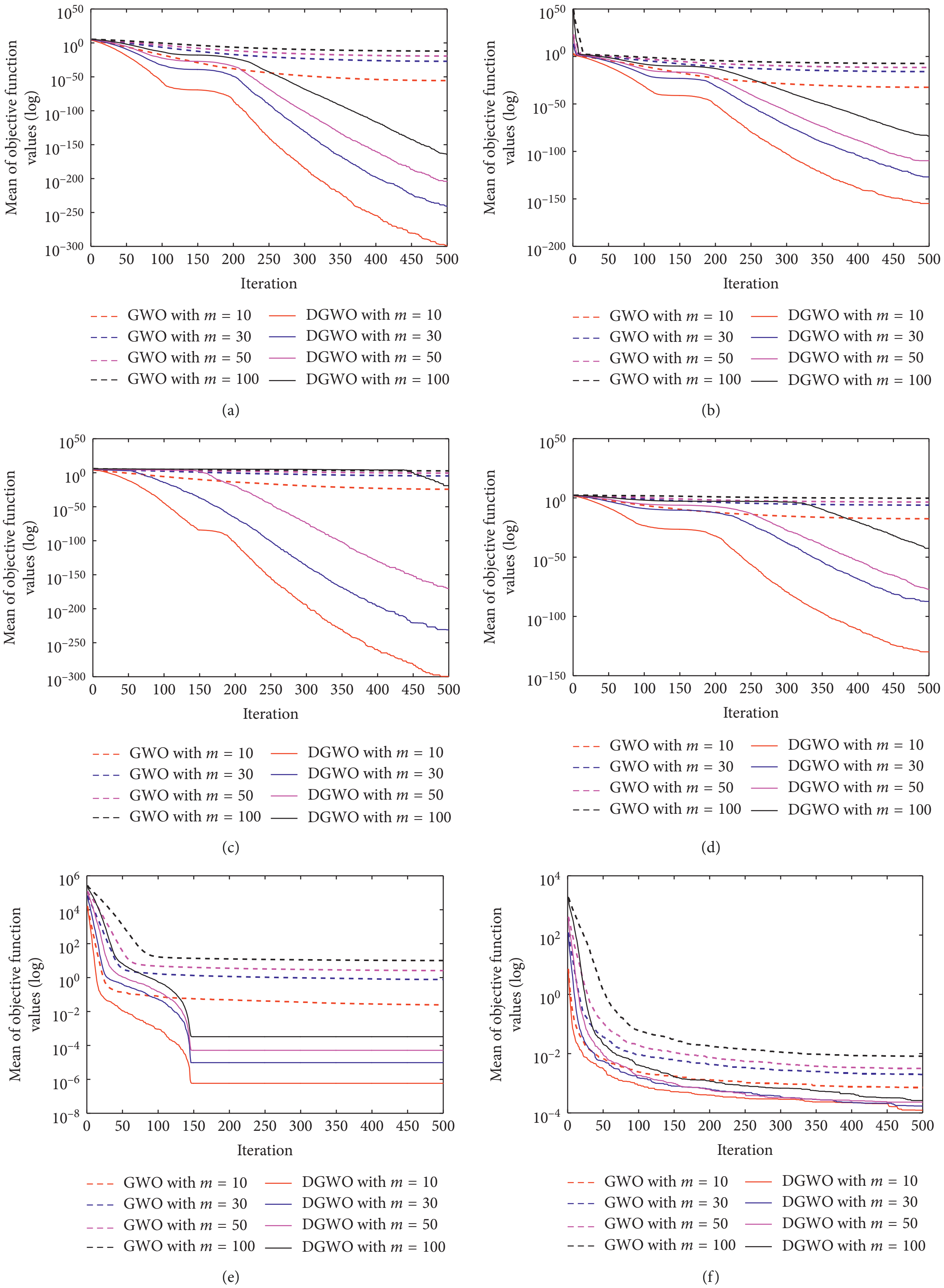
Table 6: Percentage of problems solved by the GWO and DGWO.

Function	GWO (%)				DGWO (%)			
	Dimension				Dimension			
	10	30	50	100	10	30	50	100
f_1	100	100	100	100	100	100	100	100
f_2	100	100	100	100	100	100	100	100
f_3	100	96	0	0	100	100	100	100
f_4	100	100	23	0	100	100	100	100
f_5	0	0	0	0	0	0	0	0
f_6	80	0	0	0	100	100	80	6
f_7	3	0	0	0	50	46	26	33
f_8	0	0	0	0	73	46	36	3
f_9	60	36	33	9	100	100	100	100
f_{10}	100	100	100	100	100	100	100	100
f_{11}	26	73	86	70	100	100	100	100
f_{12}	73	0	0	0	100	100	100	100
f_{13}	90	0	0	0	100	96	50	0
f_{14}	40				83			
f_{15}	43				96			
f_{16}	100				100			
f_{17}	100				96			
f_{18}	100				100			
f_{19}	100				90			
f_{20}	66				0			
f_{21}	0				66			
f_{22}	53				80			
f_{23}	96				96			

4.5. Performance Comparison with the Modified GWO Algorithm.

To further compare the optimization performance of the proposed DGWO algorithm with that of other improved GWO variants, i.e., the modified grey wolf optimizer (mGWO) algorithm [26], the grey wolf optimizer, which is based on the Powell local optimization (PGWO) method [45], and the exploration-enhanced grey wolf optimizer (EEGWO) algorithm [42], the parameters of the mGWO, PGWO, and EEGWO algorithms were established as follows: their population size was set to 30, and the maximum number of iterations was 500. The 23 benchmark test functions were selected from Table 1. The dimensions for 13 test functions (f_1–f_{13}) were set to 10, 30, 50, and 100. Each algorithm was independently run 30 times using each test function for their corresponding dimension. The mean (denoted by "Mean") and standard deviation (denoted by "St. dev.") of the fitness value are the two statistical criteria used to evaluate the performance of the algorithm. The simulation results of these four algorithms are recorded in Table 7.

As shown in Table 7, the DGWO obtained the best "Mean" and "St. dev." for functions f_1, f_2, f_3, f_8, and f_{13} with low dimensions ($m = 10$ and 30) and high dimensions ($m = 50$ and 100) compared with the mGWO, PGWO, and EEGWO. For the test problems f_4 and f_7 with $m = 30$, 50, and 100, the EEGWO achieved the best results among the four modified GWO algorithms, and the DGWO achieved slightly worse results than the EEGWO but better results than the mGWO and PGWO. For functions f_9, f_{10}, and f_{11}, the DGWO and EEGWO achieved the same results and are better than the mGWO and PGWO; note that the DGWO and EEGWO can obtain theoretical optima (0) for functions f_9 and f_{11}. The PGWO obtained the best results on test problems f_5, f_6, and f_{12} with all dimensions ($m = 10$, 30, 50, and 100) and attained the global theoretical optima (0) on problem f_6. However, the DGWO obtained the second best results for functions f_5, f_6, and f_{12}, which are similar to the results of the PGWO. For functions f_{14} to f_{23} with a fixed number of dimensions, the DGWO achieved the best results for 7 test functions (f_{14}, f_{15}, f_{17}, f_{19}, and f_{21}–f_{23}). Compared to the mGWO, the PGWO attained almost the same results for functions f_{16} and f_{20}, which are better than those of the DGWO and EEGWO. On test function f_{18}, the mGWO and PGWO obtained the best fitness values. In addition, the



(a)

(b)

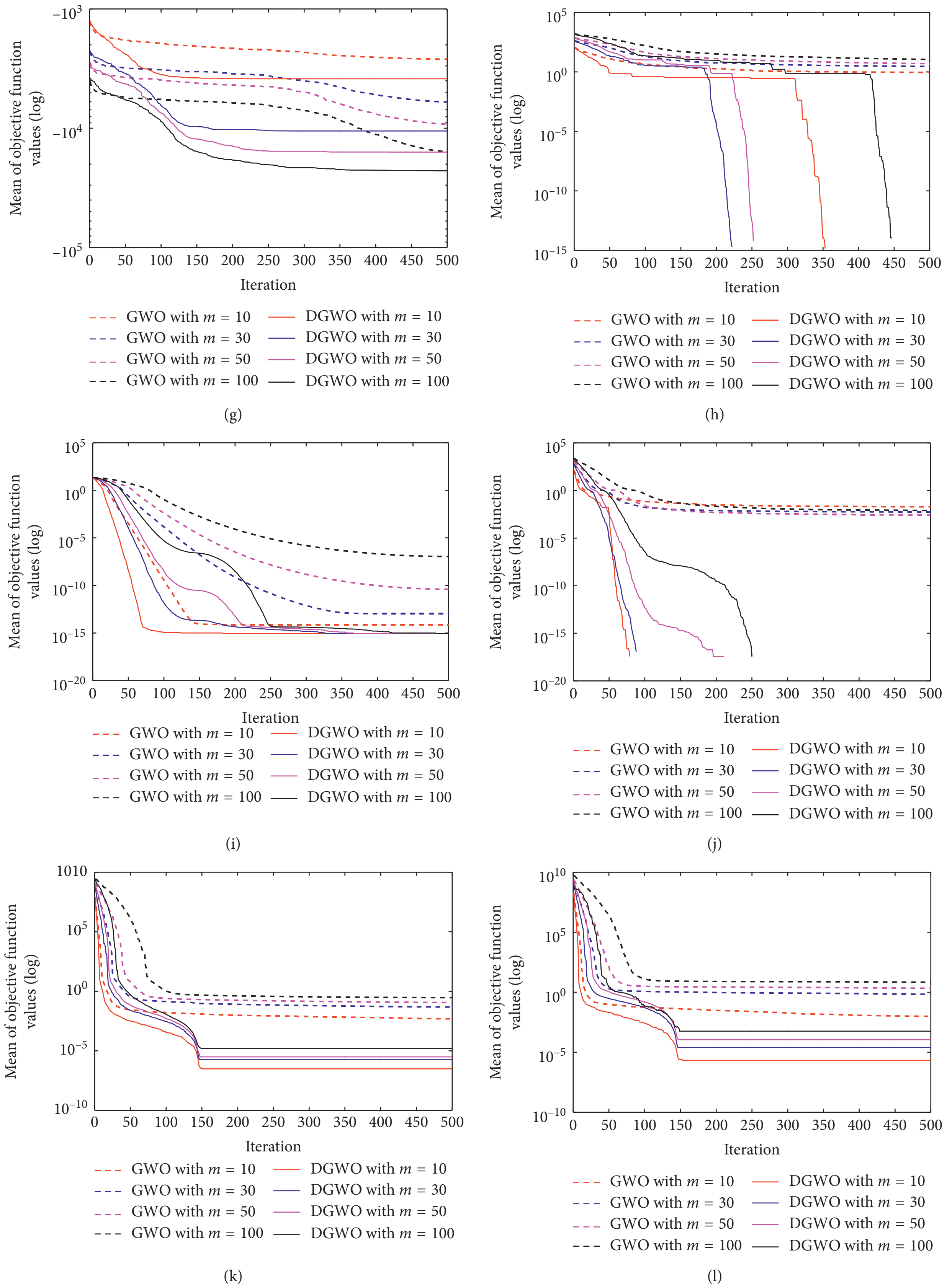
(c)

(d)

(e)

(f)

Figure 5: Continued.



(g)

(h)

(i)

(j)

(k)

(l)

Figure 5: Convergence curves of the GWO and DGWO with $m = 10$, 30, 50, and 100 on 12 typical test functions. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_6. (f) f_7. (g) f_8. (h) f_9. (i) f_{10}. (j) f_{11}. (k) f_{12}. (l) f_{13}.

Table 7: Results obtained by the three GWO variants and DGWO on 23 test problems.

Function	m	mGWO		PGWO		EEGWO		DGWO	
		Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.
f_1	10	$1.39E-74$	$6.61E-74$	$1.65E-75$	$3.98E-75$	$1.81E-245$	$0.00E-00$	$2.36E-299$	$0.00E-00$
	30	$3.90E-36$	$8.61E-36$	$1.01E-53$	$1.14E-53$	$3.23E-201$	$0.00E-00$	$2.38E-241$	$0.00E-00$
	50	$8.64E-26$	$9.26E-26$	$1.77E-48$	$1.78E-48$	$5.31E-192$	$0.00E-00$	$6.27E-205$	$0.00E-00$
	100	$2.40E-16$	$1.40E-16$	$5.99E-44$	$4.77E-44$	$2.87E-182$	$0.00E-00$	$8.19E-169$	$0.00E-00$
f_2	10	$3.49E-43$	$7.29E-43$	$6.20E-41$	$1.83E-40$	$1.51E-123$	$5.46E-123$	$1.45E-155$	$7.92E-155$
	30	$8.16E-22$	$4.82E-22$	$2.69E-28$	$1.53E-28$	$2.74E-102$	$2.95E-102$	$1.39E-127$	$7.05E-127$
	50	$6.72E-16$	$5.42E-16$	$3.26E-25$	$7.98E-26$	$6.15E-97$	$9.14E-97$	$1.80E-110$	$9.81E-110$
	100	$2.12E-10$	$9.47E-11$	$2.09E-22$	$6.37E-23$	$7.23E-92$	$9.38E-92$	$1.62E-84$	$8.48E-84$
f_3	10	$9.97E-33$	$2.68E-32$	$1.41E-40$	$3.29E-40$	$1.68E-244$	$0.00E-00$	$3.24E-300$	$0.00E-00$
	30	$1.82E-06$	$7.68E-06$	$5.16E-18$	$2.34E-17$	$1.62E-201$	$0.00E-00$	$9.45E-232$	$0.00E-00$
	50	$8.38E-02$	$2.12E-01$	$1.04E-08$	$2.32E-08$	$1.25E-190$	$0.00E-00$	$6.52E-171$	$0.00E-00$
	100	$1.16E+03$	$1.90E+03$	$1.37E-01$	$1.32E-01$	$4.18E-180$	$0.00E-00$	$1.01E-19$	$5.52E-19$
f_4	10	$1.59E-25$	$4.70E-25$	$1.24E-28$	$3.35E-28$	$3.31E-123$	$1.29E-122$	$1.25E-130$	$6.22E-130$
	30	$1.44E-09$	$2.07E-09$	$3.11E-21$	$2.06E-21$	$3.39E-102$	$5.25E-102$	$3.11E-88$	$1.70E-87$
	50	$9.12E-06$	$1.06E-05$	$6.59E-02$	$3.61E-01$	$4.73E-97$	$8.96E-97$	$8.25E-78$	$2.36E-77$
	100	$1.11E-00$	$1.00E-00$	$2.21E+01$	$1.40E+01$	$2.96E-92$	$2.25E-92$	$3.01E-43$	$1.65E-42$
f_5	10	$6.63E-00$	$6.38E-01$	$4.62E-01$	$1.39E-00$	$8.91E-00$	$5.51E-02$	$3.09E-00$	$2.31E-00$
	30	$2.68E+01$	$7.19E-01$	$1.18E-01$	$2.73E-00$	$2.89E+01$	$2.35E-02$	$2.85E-01$	$1.28E-01$
	50	$4.72E\ +\ 01$	$6.63E-01$	$2.96E+01$	$7.25E-00$	$4.89E+01$	$1.66E-02$	$4.84E+01$	$7.32E-02$
	100	$9.79E+01$	$4.48E-01$	$8.67E+01$	$2.19E+01$	$9.89E+01$	$1.79E-02$	$9.80E+01$	$9.99E-03$
f_6	10	$5.57E-06$	$2.37E-06$	$0.00E-00$	$0.00E-00$	$1.31E-00$	$2.45E-01$	$5.84E-07$	$8.37E-07$
	30	$5.97E-01$	$2.94E-01$	$0.00E-00$	$0.00E-00$	$6.11E-00$	$3.04E-01$	$9.70E-06$	$4.71E-06$
	50	$2.57E-00$	$6.05E-01$	$0.00E-00$	$0.00E-00$	$1.07E+01$	$6.03E-01$	$5.16E-05$	$4.87E-05$
	100	$9.53E-00$	$1.05E-00$	$0.00E-00$	$0.00E-00$	$2.31E+01$	$5.84E-01$	$3.28E-04$	$2.40E-05$
f_7	10	$4.48E-04$	$3.02E-04$	$7.13E-04$	$5.41E-04$	$6.13E-05$	$4.72E-05$	$1.24E-04$	$9.64E-05$
	30	$1.31E-03$	$7.25E-04$	$2.05E-03$	$9.42E-04$	$7.70E-05$	$7.05E-05$	$1.66E-04$	$1.40E-04$
	50	$2.19E-03$	$1.13E-03$	$3.59E-03$	$1.68E-03$	$5.67E-05$	$6.06E-05$	$2.31E-04$	$2.28E-04$
	100	$4.75E-03$	$1.88E-03$	$6.93E-03$	$2.34E-03$	$7.88E-05$	$5.60E-05$	$2.62E-04$	$3.68E-04$
f_8	10	$-2.60E+03$	$3.16E+02$	$-2.92E+03$	$3.05E+02$	$-1.18E+03$	$2.35E+02$	$-3.85E+03$	$5.86E+02$
	30	$-5.51+03E$	$1.47E+03$	$-7.79E+03$	$8.40E+02$	$-2.11E+03$	$4.82E+02$	$-1.06E+03$	$2.43E+03$
	50	$-8.51E+03$	$1.47E+03$	$-1.28E+04$	$1.38E+03$	$-2.85E+03$	$5.25E+02$	$-1.59E+04$	$3.94E+03$
	100	$-1.57E+04$	$2.45E+03$	$-2.49E+04$	$1.74E+03$	$-4.13E+03$	$9.09E+02$	$-2.28E+04$	$5.58E+03$
f_9	10	$9.06E-01$	$3.49E-00$	$1.00E+01$	$8.23E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	30	$7.58E-15$	$2.47E-14$	$1.71E+02$	$4.31E+01$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	50	$3.40E-02$	$1.86E-01$	$3.26E+02$	$5.89E+01$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	100	$5.35E-01$	$1.71E-00$	$6.95E+02$	$9.52E+01$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
f_{10}	10	$4.56E-15$	$6.49E-16$	$4.20E-15$	$9.01E-16$	$8.88E-16$	$0.00E-00$	$8.88E-16$	$0.00E-00$
	30	$2.32E-14$	$4.27E-15$	$4.56E-15$	$6.49E-16$	$8.88E-16$	$0.00E-00$	$8.88E-16$	$0.00E-00$
	50	$1.45E-13$	$5.01E-14$	$4.44E-15$	$0.00E-00$	$8.88E-16$	$0.00E-00$	$8.88E-16$	$0.00E-00$
	100	$1.45E-09$	$7.57E-10$	$1.40E+01$	$8.04E-00$	$8.88E-16$	$0.00E-00$	$8.88E-16$	$0.00E-00$
f_{11}	10	$1.31E-02$	$2.19E-02$	$6.67E-02$	$9.37E-02$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	30	$2.82E-03$	$7.94E-03$	$1.48E-02$	$3.65E-02$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	50	$4.56E-04$	$2.50E-03$	$1.19E-02$	$4.22E-02$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
	100	$1.32E-03$	$5.02E-03$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$	$0.00E-00$
f_{12}	10	$4.63E-03$	$9.95E-03$	$4.91E-32$	$2.04E-33$	$5.36E-01$	$1.73E-01$	$3.09E-07$	$3.33E-07$
	30	$4.89E-02$	$2.87E-02$	$1.65E-32$	$1.70E-33$	$5.57E-01$	$2.13E-01$	$1.79E-06$	$2.19E-06$
	50	$9.40E-02$	$3.20E-02$	$9.76E-33$	$6.89E-34$	$1.03E-00$	$7.70E-02$	$3.12E-06$	$3.31E-06$
	100	$2.68E-01$	$5.55E-02$	$1.04E-03$	$5.68E-03$	$1.11E-00$	$4.21E-02$	$1.61E-05$	$1.43E-05$
f_{13}	10	$1.24E-02$	$3.23E-02$	$3.25E-03$	$1.78E-02$	$7.83E-01$	$1.17E-01$	$2.09E-06$	$2.46E-06$
	30	$5.23E-01$	$2.00E-01$	$2.33E-02$	$5.10E-02$	$2.99E-00$	$3.12E-02$	$2.49E-05$	$2.75E-05$
	50	$1.79E-00$	$2.58E-00$	$7.32E-02$	$9.48E-02$	$4.99E-00$	$1.61E-03$	$1.12E-04$	$8.81E-05$
	100	$6.21E-00$	$4.17E-01$	$1.61E-01$	$2.40E-01$	$9.99E-00$	$2.12E-03$	$9.13E-04$	$5.72E-04$
f_{14}	2	4.3347	$3.70E+00$	4.3244	$4.35E+00$	10.2709	$2.74E+00$	2.9434	$4.43E+00$
f_{15}	4	0.0032	$6.86E-03$	0.0005	$4.16E-04$	0.0049	$2.97E-03$	0.0003	$2.31E-04$
f_{16}	2	−1.0316	$1.02E-07$	−1.0316	$1.42E-08$	−0.9661	$5.57E-02$	−1.0303	$5.17E-03$
f_{17}	2	0.3979	$1.18E-06$	0.3979	$5.87E-12$	1.7185	$1.23E+00$	0.3980	$2.11E-04$
f_{18}	2	3.0000	$5.43E-05$	3.0000	$4.61E-07$	20.0084	$2.05E+01$	3.0001	$2.41E-04$
f_{19}	3	−3.8619	$2.06E-03$	−3.8628	$8.57E-08$	−3.2895	$3.11E-01$	−3.8590	$5.80E-04$
f_{20}	6	−3.2436	$7.74E-02$	−3.2744	$5.92E-02$	−1.5884	$4.50E-01$	−3.1725	$5.27E-02$

Table 7: Continued.

Function	m	mGWO		PGWO		EEGWO		DGWO	
		Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.
f_{21}	4	−8.5891	$2.69E+00$	−8.9728	$2.18E+00$	−0.6512	$2.36E-01$	−10.0984	$1.56E-01$
f_{22}	4	−10.2238	$9.70E-01$	−10.0513	$1.34E+00$	−0.6848	$2.16E-01$	−10.2969	$2.58E-01$
f_{23}	4	−10.5334	$1.79E-03$	−9.9085	$1.96E+01$	−0.9298	$2.72E-01$	−10.5334	$2.94E-02$

EEGWO algorithm exhibits poor optimization performance on functions f_{14} to f_{23}.

From Table 7, we can see that the EEGWO provides very competitive results compared to the DGWO, and it is challenging to determine which algorithm is better. Therefore, it is necessary to conduct an appropriate statistical analysis to see whether the obtained results of the employed algorithms are significant at a given confidence interval. In this paper, the sign test is adopted, which is obtained from references [11, 46]. The statistical results are recorded in Table 8. It should be noted that these statistical analysis results are conducted based on the average results of the 20 independently obtained best results. As seen from Table 8, the DGWO is significantly better than the GWO, mGWO, and PGWO on unimodal and multimodal test functions at a significance level of 0.05 but shows a nonsignificant performance difference on 10 fixed-dimension multimodal benchmark functions. In addition, when compared to the EEGWO, the DGWO provides a nonsignificant performance difference on 13 unimodal and multimodal test functions but obtains significantly better results on 10 fixed-dimension multimodal benchmark functions at a significance level of 0.1.

The percentages of problems solved by the mGWO, PGWO, and EEGWO are recorded in Table 9. Compared to the mGWO, the DGWO obtained the same percentage on six functions (i.e., f_1, f_2, f_5, f_{16}, f_{18}, and f_{23}) and a higher percentage on thirteen functions (i.e., f_3-f_4, f_6–f_9, f_{11}–f_{13}, f_{15}, and f_{21}-f_{22}), while the DGWO showed a lower percentage on three functions (i.e., f_{17}, f_{19}, and f_{20}). Compared to the PGWO, the DGWO provided the same and higher percentage on five functions (i.e., f_1, f_2, f_6, f_{11}, and f_{18}) and eleven functions (i.e., f_3-f_4, f_7–f_{12}, f_{14}-f_{15}, and f_{23}), respectively; on the contrary, the PGWO showed a higher percentage than the DGWO on six functions (i.e., f_5, f_{17}, f_{19}, and f_{20}–f_{22}). For function f_{13}, the DGWO showed a higher percentage than the PGWO when the dimensions were 10, 30, and 50 but obtained a lower percentage when the dimension was 100. Compared to the EEGWO, the DGWO achieved the same and higher percentage on nine functions (i.e., f_1–f_5, f_9–f_{11}, and f_{20}) and twelve functions (i.e., f_8, f_{12}-f_{13}, f_{14}–f_{19}, and f_{21}–f_{23}), respectively. For function f_7, however, the EEGWO obtained a higher percentage than the DGWO.

To investigate the convergence speed of the three modified versions of the GWO mentioned in this paper and the proposed DGWO algorithm for low-dimensional and high-dimensional problems, Figure 6 plots the convergence curves of 10 typical functions (f_1–f_4, f_6-f_7, f_9-f_{10}, and f_{12}-f_{13}) with dimensions of 30 and 100. For functions f_1–f_4, f_7, and f_9, the DGWO and EEGWO achieve the fastest convergence speed, whereas the EEGWO achieves a faster convergence speed for high-dimensional functions; however, the DGWO attains a better convergence speed for low-dimensional problems. The PGWO has a fast convergence speed for functions f_6 and f_{12}, and the DGWO ranked second. The DGWO exhibits the fastest convergence speed for functions f_{10} and f_{13}, and the EEGWO shows the same convergence speed for function f_{10}. These analysis results verify that the proposed DGWO achieves excellent convergence performance for both low-dimensional problems and high-dimensional problems.

In addition to the abovementioned GWO versions, an interesting GWO variant named "GWO-EPD" [47] has successfully caught our attention because it exhibits some similarities and differences compared with our proposed DGWO algorithm. The GWO-EPD algorithm has some features that are similar to those of the DGWO algorithm, such as dynamically removing some inferior solutions and repositioning them by adopting alpha, beta, and delta wolves. However, the differences between those two algorithms are also easy to distinguish. For example, in the DGWO, some variables of the current best solution are removed and repositioned by using probability that was modeled as equation (12), while in GWO-EPD, half of the worst search agents are eliminated and reinitialized with equal probability. In addition, in the DGWO, the variables are repositioned by employing the modified position-updated equation (see equation (26)); however, in GWO-EDP, the mechanism of EDP is applied in the GWO algorithm to randomly reinitialize its worst search agents. To further verify the scalability of the DGWO, we compared it with GWO-EPD on 13 test functions (i.e., f_1–f_{13}), and the results are recorded in Table 1, with dimensions from 30 to 100. All of the DGWO parameters were kept the same as those defined in the above section. The parameter values of GWO-EPD were kept the same as in its original papers. In addition, the maximum number of iterations and population size were set as 500 and 30, respectively, and 30 independent runs were executed for each test function. The experimental results are presented in Table 10.

As seen from Table 10, for $m = 30$, compared to the GWO-EPD algorithm, the DGWO provided better results on eleven functions (i.e., f_1–f_4, f_6-f_7, and f_9–f_{13}). Similarly, for $m = 100$, when compared to GWO-EPD, the DGWO also offered better results on eleven functions (i.e., f_1–f_4, f_6-f_7, and f_9–f_{13}). However, better results for f_5 and f_8 were obtained by the GWO-EPD algorithm. In summary, the increase in dimensions has little impact on the performance of the DGWO algorithm. Even when suffering from large-scale optimization problems, the DGWO still worked well and obtained promising results.

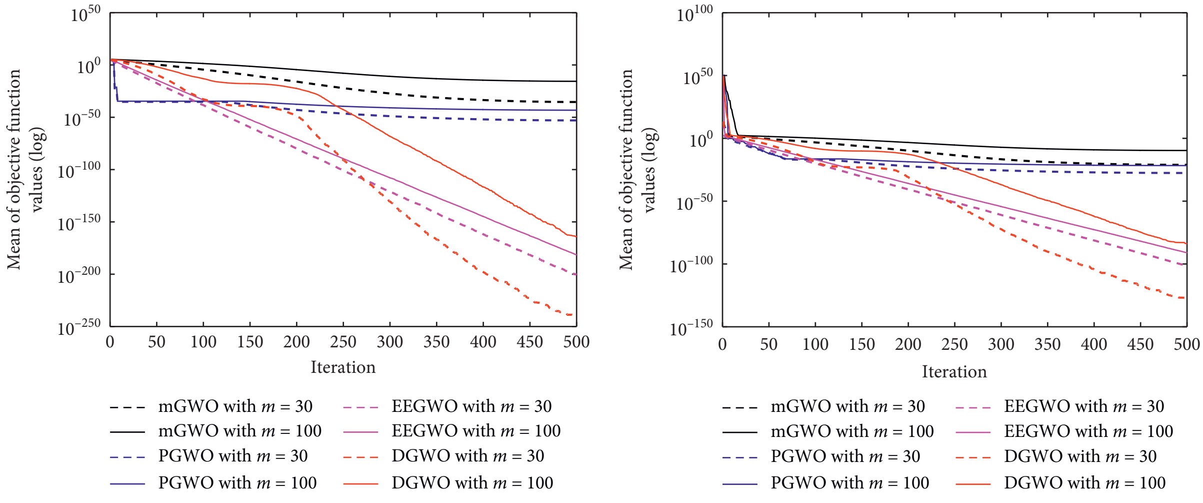
Table 8: Sign test results of the five different GWO algorithms.

DGWO	GWO					mGWO					PGWO					EEGWO				
	f_1–f_{13}				f_{14}–f_{23}	f_1–f_{13}				f_{14}–f_{23}	f_1–f_{13}				f_{14}–f_{23}	f_1–f_{13}				f_{14}–f_{23}
	Dimension					Dimension					Dimension					Dimension				
	10	30	50	100		10	30	50	100		10	30	50	100		10	30	50	100	
Wins (+)	13	12	12	12	6	11	11	11	11	5	13	12	12	12	6	8	8	7	5	10
Approximations (≈)	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	3	3	3	3	0
Losses (−)	0	1	1	1	4	2	2	2	1	4	0	1	1	1	4	2	2	3	5	0
Detected differences (α)	0.05	0.05	0.05	0.05	n	0.05	0.05	0.05	0.05	n	0.05	0.05	0.05	0.05	n	n	n	n	n	0.1

Note: "n" refers to nonsignificant at the significance levels of 0.05 and 0.1.

Table 9: Percentage of problems solved by the mGWO, PGWO, and EEGWO.

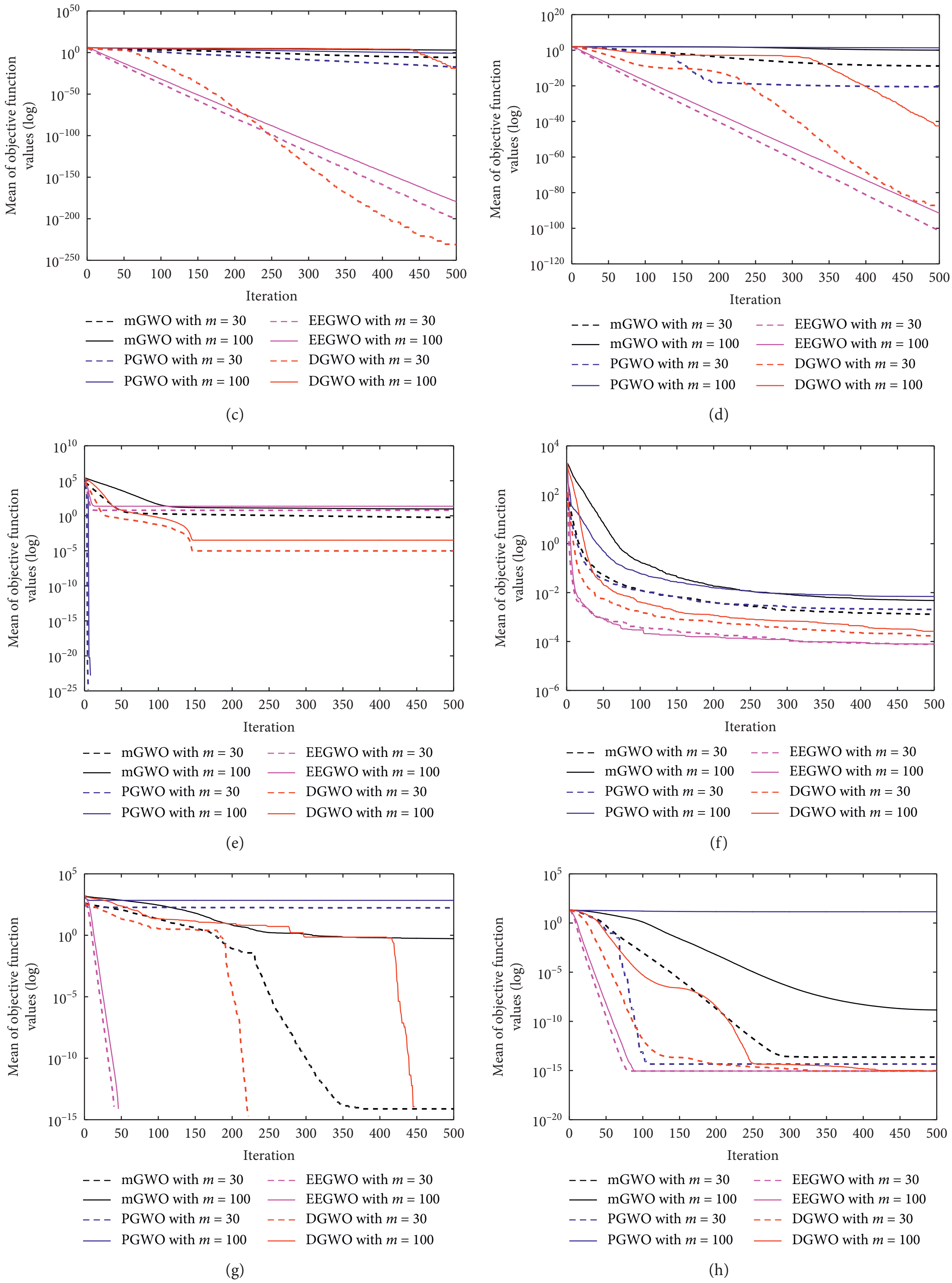
Function	mGWO (%)				PGWO (%)				EEGWO (%)			
	Dimension				Dimension				Dimension			
	10	30	50	100	10	30	50	100	10	30	50	100
f_1	100	100	100	100	100	100	100	100	100	100	100	100
f_2	100	100	100	100	100	100	100	100	100	100	100	100
f_3	100	100	6	0	100	100	100	0	100	100	100	100
f_4	100	100	100	0	100	100	96	0	100	100	100	100
f_5	0	0	0	0	6	0	0	0	0	0	0	0
f_6	100	0	0	0	100	100	100	100	0	0	0	0
f_7	13	0	0	0	0	0	0	0	83	70	73	70
f_8	0	0	0	0	0	0	0	0	0	0	0	0
f_9	90	96	96	83	6	0	0	0	100	100	100	100
f_{10}	100	100	100	100	100	100	100	10	100	100	100	100
f_{11}	53	86	96	93	33	80	90	100	100	100	100	100
f_{12}	80	0	0	0	100	100	100	96	0	0	0	0
f_{13}	86	0	0	0	96	66	30	13	0	0	0	0
f_{14}	23				43				0			
f_{15}	16				66				0			
f_{16}	100				100				0			
f_{17}	100				100				0			
f_{18}	100				100				0			
f_{19}	100				100				0			
f_{20}	46				60				0			
f_{21}	0				76				0			
f_{22}	26				93				0			
f_{23}	96				90				0			



(a)

(b)

Figure 6: Continued.



(c)

(d)

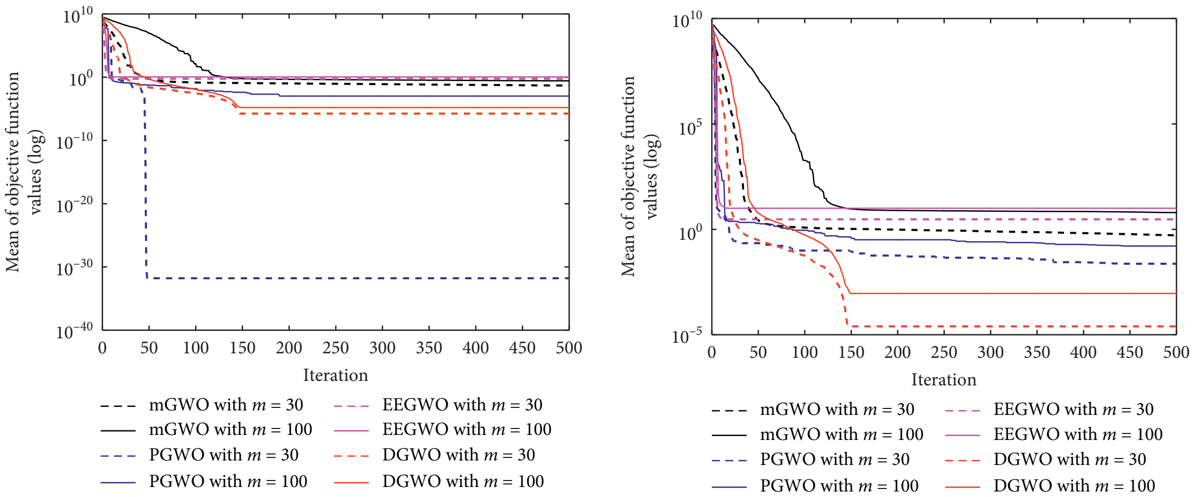
(e)

(f)

(g)

(h)

Figure 6: Continued.



(i) (j)

Figure 6: Convergence curves of the four different GWO variants with $m = 30$ and 100 on 10 typical test functions. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_6. (f) f_7. (g) f_9. (h) f_{10}. (i) f_{12}. (j) f_{13}.

Table 10: Mean and St. dev. results of the DGWO and GWO-EPD for thirteen functions ($m = 30$ and 100).

Function	GWO-EPD with $m = 30$ Mean ± St. dev.	DGWO with $m = 30$ Mean ± St. dev.	GWO-EPD with $m = 100$ Mean ± St. dev.	DGWO with $m = 100$ Mean ± St. dev.
f_1	6.75E − 25 ± 1.16E − 24	**2.36E − 299 ± 0.00E + 00**	3.64E − 01 ± 5.28E − 01	**8.19E − 165 ± 0.00E + 00**
f_2	4.52E − 15 ± 3.98E − 15	**1.45E − 155 ± 7.92E − 155**	3.00E − 02 ± 3.04E − 02	**1.62E − 84 ± 8.48E − 84**
f_3	2.89E − 04 ± 6.38E − 04	**3.24E − 300 ± 0.00E + 00**	2.50E + 00 ± 1.99E + 00	**1.01E − 19 ± 5.52E − 19**
f_4	5.41E − 02 ± 5.16E − 02	**1.25E − 130 ± 6.22E − 130**	4.87E − 02 ± 4.92E − 02	**3.01E − 43 ± 1.65E − 42**
f_5	**2.02E − 01 ± 3.50E − 01**	3.10E + 00 ± 2.85E + 00	**2.10E + 01 ± 4.34E + 01**	9.80E + 01 ± 9.99E − 03
f_6	2.34E − 01 ± 1.93E − 01	**5.84E − 07 ± 8.37E − 07**	2.24E + 02 ± 1.29E + 01	**3.28E − 04 ± 2.40E − 04**
f_7	3.87E − 03 ± 2.59E − 03	**1.24E − 04 ± 9.64E − 05**	2.31E − 02 ± 2.20E − 02	**2.62E − 04 ± 3.68E − 04**
f_8	**−1.26E + 04 ± 1.63E − 01**	−3.85E + 03 ± 5.86E + 02	**−4.19E + 05 ± 1.50E + 01**	−2.28E + 04 ± 5.58E + 03
f_9	1.57E − 08 ± 8.57E − 08	**0.00E + 00 ± 0.00E + 00**	2.80E − 01 ± 4.20E − 01	**0.00E + 00 ± 0.00E + 00**
f_{10}	2.46E − 11 ± 1.32E − 10	**8.88E − 16 ± 0.00E + 00**	1.33E − 02 ± 1.33E − 02	**8.88E − 16 ± 0.00E + 00**
f_{11}	1.53E − 03 ± 5.05E − 03	**0.00E + 00 ± 0.00E + 00**	3.22E − 02 ± 1.23E − 01	**0.00E + 00 ± 0.00E + 00**
f_{12}	3.45E − 02 ± 1.35E − 02	**3.09E − 07 ± 3.33E − 07**	7.70E − 01 ± 7.58E − 02	**1.61E − 05 ± 1.43E − 05**
f_{13}	3.20E − 04 ± 1.94E − 04	**9.13E − 04 ± 5.72E − 04**	7.07E − 02 ± 2.07E − 01	**9.13E − 04 ± 5.72E − 04**

The bold values in GWO-EPD with m = 30 are losses (−): 10, wins (+): 2, approximations (=): 1, and detected differences (α): 0.05. The bold values in GWO-EPD with m = 100 are losses (−): 11, wins (+): 2, approximations (=): 0, and detected differences (α): 0.05.

4.6. Performance Comparison with Other State-of-the-Art Algorithms ($m = 30$). In this section, we compared the DGWO to seven recently proposed state-of-the-art population-based optimization methods, such as the autonomous particles groups for particle swarm optimization (AGPSO) [48], the improved PSO with time-varying accelerator coefficients (IPSO) [49], the improved PSO algorithm based on asymmetric time-varying acceleration coefficients (MPSO) [50], the time-varying acceleration coefficients particle swarm optimization (TACPSO) [51], the hybrid differential evolution with biogeography-based optimization (DEBBO) [52], the hybrid whale optimization algorithm with simulated annealing (WOA-SA) [53], and the salp swarm algorithm (SSA) [54]. All DGWO parameters were kept the same as those listed in Section 4.1. The parameter settings of the seven algorithms are listed as follows: the population size is 30, the maximum number of iterations is 500, and the other algorithm parameters are the same as in their original papers.

To compare the optimization performance of the seven algorithms, the results are achieved over 30 independent runs. The best (denoted by "Best"), average (denoted by "Mean"), and standard deviation (denoted by "St. dev.") of the best solution in the last iteration are collected in Table 11. The best obtained results are highlighted in boldface type.

Table 11 shows the results for 23 test functions. As presented in this table, the DGWO had the best results for three of seven unimodal benchmark problems (i.e., f_3, f_4, and f_7). For function f_6, the DGWO performed slightly worse than the SSA and obtained the second best result. For functions f_1 and f_2, the WOA-SA achieved the global optimal values (0), and the DGWO provided solutions near 0. For the multimodal benchmark functions f_8–f_{13}, the DGWO presented the best results, with the exception of functions f_{12} and f_{13}. For functions f_8, f_{12}, and f_{13}, the DEBBO obtained almost the same results as the DGWO. Compared to the WOA-SA, the DGWO obtained similar and worse results for

Table 11: Comparison results of the DGWO and other seven algorithms on 23 test functions ($m = 30$).

Test function		AGPSO	IPSO	MPSO	TACPSO	DEBBO	WOA-SA	SSA	DGWO
f_1	Best	1.04E − 03	1.75E − 04	9.29E − 03	8.09E − 04	1.04E − 05	0.00E + 00	3.49E − 08	1.63E − 287
	Mean	2.42E − 01	1.12E − 01	1.00E + 03	2.35E − 01	3.48E − 05	0.00E + 00	2.03E − 07	2.38E − 241
	St. dev.	1.05E + 00	4.89E − 01	3.05E + 03	8.70E − 01	1.93E − 05	0.00E + 00	2.52E − 07	0.00E + 00
f_2	Best	1.78E − 02	3.78E − 02	1.00E + 01	6.65E − 02	2.11E − 04	0.00E + 00	2.78E − 01	3.74E − 148
	Mean	5.99E + 00	6.04E − 00	3.84E + 01	8.35E − 01	3.63E − 04	0.00E + 00	2.13E + 00	1.39E − 127
	St. dev.	7.61E + 00	7.69E − 00	1.99E + 01	1.84E + 00	9.71E − 05	0.00E + 00	1.56E + 00	7.05E − 127
f_3	Best	2.64E + 02	6.23E + 02	1.09E + 03	1.66E + 02	6.45E + 03	1.70E − 02	3.81E + 02	1.67E − 260
	Mean	4.26E + 03	6.65E + 03	1.95E + 04	1.09E + 03	1.59E + 04	5.46E − 01	1.59E + 03	9.45E − 232
	St. dev.	4.97E + 03	5.40E + 03	9.13E + 03	1.47E + 03	3.40E + 03	3.06E − 01	1.02E + 03	0.00E + 00
f_4	Best	7.93E + 00	4.34E + 00	7.86E + 00	2.78E + 00	4.25E + 00	2.53E − 02	6.69E + 00	1.30E − 106
	Mean	1.63E + 01	1.03E + 01	2.00E + 01	9.64E + 00	6.76E + 00	7.82E − 01	1.09E + 01	3.11E − 88
	St. dev.	4.77E + 00	3.58E + 00	5.87E + 00	3.44E + 00	1.64E + 00	4.02E − 01	3.71E + 00	1.70E − 87
f_5	Best	2.47E + 01	7.40E + 01	7.83E + 01	2.91E + 01	2.53E + 01	0.00E + 00	4.51E + 00	2.82E + 01
	Mean	3.29E + 02	3.45E + 03	1.26E + 04	3.20E + 03	5.09E + 01	3.03E + 01	1.49E + 02	2.85E + 01
	St. dev.	7.38E + 02	1.64E + 04	3.09E + 04	1.64E + 04	3.13E + 01	2.82E + 01	3.48E + 02	1.28E − 01
f_6	Best	1.51E − 03	1.64E − 04	3.74E − 03	4.87E − 04	1.09E − 05	3.39E − 04	3.72E − 08	1.48E − 06
	Mean	7.83E − 02	7.14E − 02	3.30E + 02	1.73E − 01	3.32E − 05	8.44E − 04	2.09E − 07	9.70E − 06
	St. dev.	9.30E − 02	2.78E − 01	1.81E + 03	3.42E − 01	1.93E − 05	3.07E − 04	3.00E − 07	4.71E − 06
f_7	Best	4.63E − 02	3.38E − 02	3.89E − 02	4.32E − 02	2.04E − 02	1.50E − 02	4.64E − 02	2.88E − 05
	Mean	1.14E − 01	7.41E − 02	1.25E + 00	8.62E − 02	4.22E − 02	4.93E − 02	1.78E − 01	1.66E − 04
	St. dev.	5.01E − 02	3.05E − 02	2.89E + 00	3.67E − 02	1.41E − 02	2.39E − 02	7.96E − 02	1.40E − 04
f_8	Best	−7.83E + 03	−7.77E + 03	−7.44E + 03	−7.20E + 03	−1.22E + 04	−8.86E + 03	−6.31E + 03	−1.24E + 04
	Mean	−9.30E + 03	−9.23E + 03	−8.85E + 03	−8.52E + 03	−1.25E + 04	−1.03E + 04	−7.36E + 03	−1.26E + 04
	St. dev.	7.09E + 02	7.04E + 02	7.51E + 02	7.72E + 02	1.09E + 02	8.47E + 02	6.76E + 02	2.43E + 03
f_9	Best	4.38E + 01	3.68E + 01	7.76E + 01	2.81E + 01	2.36E + 01	0.00E + 00	2.09E + 01	0.00E + 00
	Mean	8.39E + 01	9.73E + 01	1.34E + 02	7.11E + 01	3.27E + 01	0.00E + 00	5.13E + 01	0.00E + 00
	St. dev.	2.65E + 01	2.84E + 01	3.37E + 01	2.27E + 01	5.66E + 00	0.00E + 00	1.68E + 01	0.00E + 00
f_{10}	Best	2.26E + 00	6.08E − 03	6.78E − 01	9.33E − 01	7.68E − 04	8.88E − 16	1.16E + 00	8.88E − 16
	Mean	3.69E + 00	2.23E + 00	7.37E + 00	2.34E + 00	1.34E − 03	8.88E − 16	2.78E + 00	8.88E − 16
	St. dev.	8.02E − 01	9.89E − 01	6.23E + 00	9.60E − 01	5.11E − 04	0.00E + 00	7.52E − 01	0.00E + 00
f_{11}	Best	1.00E − 02	2.46E − 03	1.16E − 02	2.84E − 03	1.53E − 05	0.00E + 00	9.26E − 04	0.00E + 00
	Mean	1.50E − 01	9.81E − 02	6.31E + 00	1.12E − 01	1.20E − 03	0.00E + 00	1.47E − 02	0.00E + 00
	St. dev.	1.22E − 01	1.26E − 01	2.29E + 01	1.74E − 01	2.76E − 03	0.00E + 00	1.34E − 02	0.00E + 00
f_{12}	Best	9.88E − 01	1.09E − 01	5.83E − 01	6.81E − 04	1.22E − 06	2.04E − 05	2.81E + 00	4.04E − 06
	Mean	4.18E − 00	1.47E + 00	3.59E + 00	1.92E + 00	9.46E − 06	6.36E − 05	6.75E + 00	1.61E − 05
	St. dev.	2.48E − 00	1.28E + 00	1.77E + 00	1.55E + 00	9.45E − 06	5.37E − 05	3.10E + 00	1.43E − 05
f_{13}	Best	1.30E + 00	1.89E − 03	2.19E − 01	3.10E − 02	8.50E − 06	1.35E − 32	1.92E − 01	1.86E − 04
	Mean	9.78E + 00	4.00E + 00	9.39E + 00	4.62E + 00	4.46E − 05	1.35E − 32	1.63E + 01	9.13E − 04
	St. dev.	7.45E + 00	6.42E + 00	6.58E + 00	4.26E + 00	2.39E − 05	5.57E − 48	1.43E + 01	5.72E − 04
f_{14}	Best	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
	Mean	0.9980	0.9980	0.9980	0.9980	1.0900	7.4300	1.3900	2.9400
	St. dev.	1.70E − 16	1.60E − 16	7.14E − 17	2.30E − 16	3.03E − 01	5.13E + 00	8.86E − 01	4.43E + 00
f_{15}	Best	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0004	0.0003
	Mean	0.0021	0.0013	0.0043	0.0012	0.0020	0.0032	0.0015	0.0003
	St. dev.	4.98E − 03	3.63E − 03	7.34E − 03	3.64E − 03	5.00E − 03	1.19E − 02	3.57E − 03	2.31E − 04
f_{16}	Best	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Mean	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	St. dev.	6.12E − 16	6.05E − 16	6.25E − 16	5.83E − 16	6.65E − 16	1.80E − 10	4.87E − 14	5.17E − 03
f_{17}	Best	0.3979	0.3979	3.9789	0.3979	0.3979	0.3979	0.3979	0.3979
	Mean	0.3989	0.3979	3.9789	0.3979	0.3979	0.3979	0.3979	0.3980
	St. dev.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	5.82E − 06	7.11E − 14	2.11E − 04
f_{18}	Best	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	Mean	3.0000	3.0000	3.0000	3.0000	3.0000	4.8000	3.0000	3.0000
	St. dev.	2.27E − 15	2.46E − 15	2.48E − 15	1.59E − 15	1.36E − 15	6.85E + 01	2.64E − 13	2.41E − 04
f_{19}	Best	−3.8629	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8626
	Mean	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8590
	St. dev.	2.61E − 15	2.61E − 15	2.64E − 15	2.63E − 15	2.71E − 15	4.17E − 09	1.23E − 10	5.87E − 03
f_{20}	Best	−3.3220	−3.3220	−3.3220	−3.3220	−3.3220	−3.3220	−3.3220	−3.2959
	Mean	−3.2643	3.2600	−3.2691	−3.2467	−3.2903	−3.2903	−3.2136	−3.1725
	St. dev.	6.38E − 02	6.41E − 02	6.78E − 02	5.83E − 02	5.35E − 02	5.35E − 02	5.03E − 02	5.27E − 02

Table 11: Continued.

Test function		AGPSO	IPSO	MPSO	TACPSO	DEBBO	WOA-SA	SSA	DGWO
f_{21}	Best	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532
	Mean	−6.1332	−6.6316	−6.3807	−5.8073	−8.3101	−5.2251	−6.8967	−10.0984
	St. dev.	3.06E + 00	3.06E + 00	3.07E + 00	3.27E + 00	2.94E + 00	9.31E − 01	3.42E + 00	1.56E − 01
f_{22}	Best	−10.4029	−10.4029	−10.4029	−10.4029	−10.4029	−5.0877	−10.4029	−10.4029
	Mean	−7.5123	−8.9999	−8.4546	−8.3648	−9.2703	−5.0877	−9.0951	−10.2969
	St. dev.	3.26E + 00	2.63E + 00	2.88E + 00	3.21E + 00	2.59E + 00	3.18E − 07	2.70E + 00	2.58E − 01
f_{23}	Best	−10.5364	−10.5364	−10.5364	−10.5364	−10.5364	−5.1285	−10.5364	−10.5364
	Mean	−9.0557	−9.8342	−9.1913	−8.4994	−10.0064	−5.1285	−8.9619	−10.5307
	St. dev.	2.79E + 00	2.15E + 00	2.78E + 00	3.22E + 00	2.00E + 00	1.86E − 12	2.96E + 00	2.94E − 02

Table 12: Percentage of problems solved by the GWO and DGWO.

Function	AGPSO (%)	IPSO (%)	MPSO (%)	TACPSO (%)	DEBBO (%)	WOA-SA (%)	SSA (%)	DGWO (%)
f_1	0	0	0	0	100	100	100	100
f_2	0	0	0	0	0	100	0	100
f_3	0	0	0	0	0	0	0	100
f_4	0	0	0	0	0	0	0	100
f_5	0	0	0	0	0	3	0	0
f_6	0	0	0	0	100	0	100	100
f_7	0	0	0	0	0	0	0	46
f_8	0	0	0	0	60	0	0	46
f_9	0	0	0	0	0	100	0	100
f_{10}	0	0	0	0	0	100	0	100
f_{11}	0	0	0	0	50	100	0	100
f_{12}	0	0	0	0	100	80	0	100
f_{13}	0	0	0	0	96	100	0	96
f_{14}	100	100	100	100	90	20	80	83
f_{15}	10	53	3	70	10	60	0	96
f_{16}	100	100	100	100	100	100	100	100
f_{17}	100	100	100	100	100	100	100	96
f_{18}	100	100	100	100	100	90	100	100
f_{19}	100	100	100	100	100	100	100	90
f_{20}	53	50	60	36	70	70	16	0
f_{21}	33	40	36	30	70	3	50	66
f_{22}	53	76	66	70	80	0	80	80
f_{23}	76	90	80	70	90	0	76	96

three functions (f_9–f_{11}) and one function (f_{13}), respectively. Table 11 also shows the results for 10 fixed-dimension multimodal benchmark functions (f_{14}–f_{23}). As shown in Table 11, the results of the AGPSO, IPSO, MPSO, and TACPSO are equal for four functions (f_{14}, f_{16}, f_{18}, and f_{20}) and better than those of the DGWO. However, the DGWO achieved the best results of all algorithms for six fixed-dimension unimodal benchmark problems (i.e., f_{15}, f_{17}, f_{19}, and f_{21}–f_{23}). The WOA-SA and SSA obtained similar results for function f_{20} and are better than the other algorithms.

The percentages of problems solved by the seven state-of-the-art algorithms are listed in Table 12. As seen from this table, the AGPSO, IPSO, MPSO, and TACPSO have all failed to solve thirteen test functions (i.e., f_1–f_{13}) but completely solved all five test functions (i.e., f_{14} and f_{16}–f_{19}). Of the thirteen functions f_1–f_{13}, nine functions are completely solved by the DGWO, three functions are fully solved by the DEBBO, six functions are fully solved by the WOA-SA, and two functions are fully solved by the SSA. Of the ten functions f_{14}–f_{23}, four functions are completely solved by the DEBBO and SSA, three functions are fully solved by the WOA-SA, and two functions are fully solved by the DGWO. However, for functions f_{15} and f_{22}-f_{23}, the DGWO has achieved the highest percentage.

Figure 7 plots the convergence curves of the average objective function values of the algorithms for some typical test problems, where f_1, f_3, f_4, and f_7 are unimodal functions, f_9, f_{10}, and f_{11} are multimodal benchmark functions, and f_{15}, f_{21}, and f_{23} are fixed-dimension multimodal benchmark functions. As observed from these curves, the DGWO has the best convergence rates for all 10 classic benchmark functions. Note that unimodal test problems are suitable for benchmarking the convergence ability of algorithms since they have only one global minimum and do not have local minima in the search space [48]. Since multimodal and fixed-dimension multimodal benchmark functions have more than one local optimal solution, they are suitable for benchmarking the capability of algorithms in avoiding local minima [48]. As indicated by the results, the DGWO performs better than the seven compared algorithms on both the unimodal and multimodal benchmark functions. The DGWO achieves superior results because the candidate

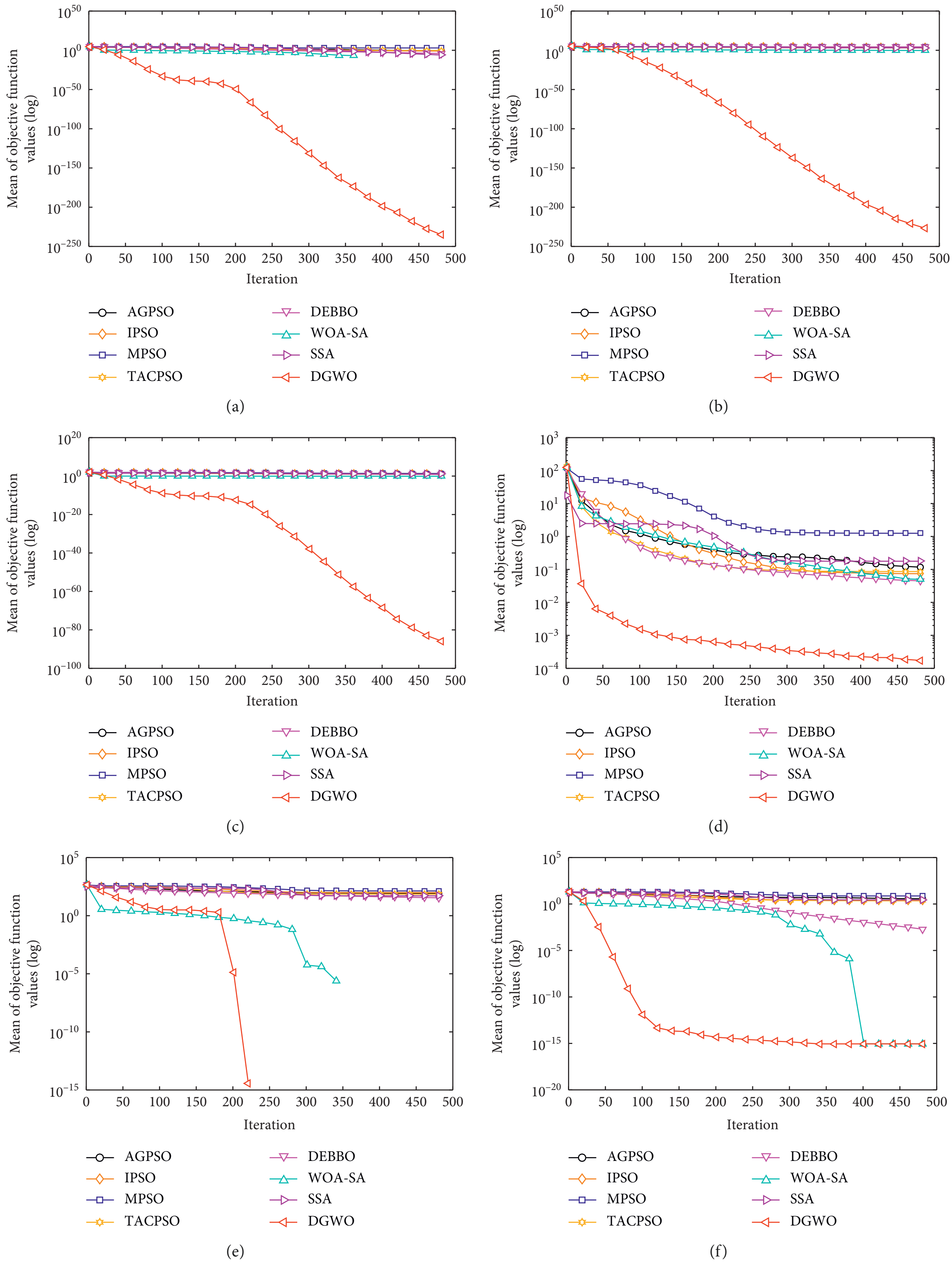
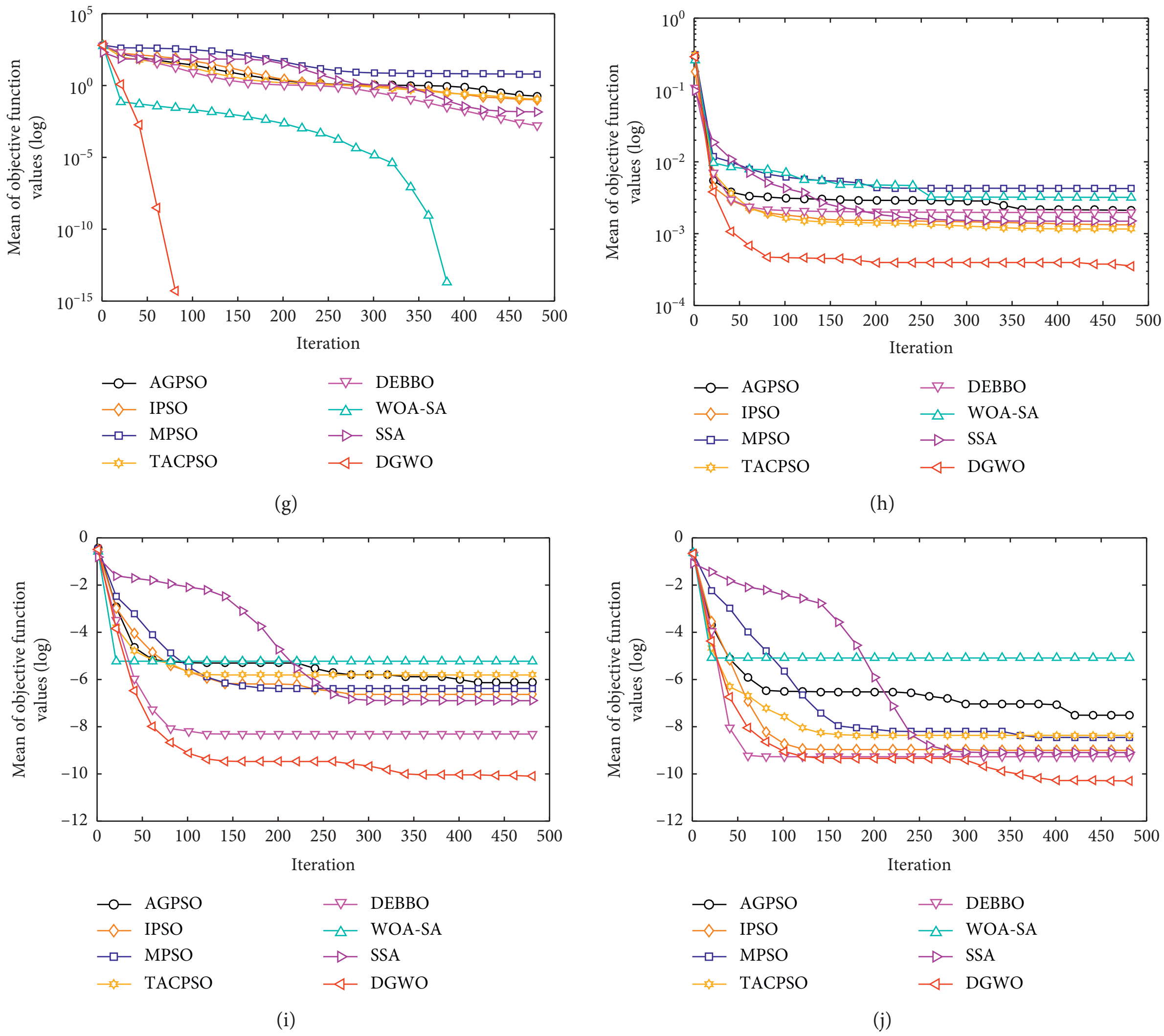


Figure 7: Continued.



(g)

(h)

(i)

(j)

FIGURE 7: Convergence curves of the algorithms on the 10 classic benchmark functions. (a) f_1. (b) f_3. (c) f_4. (d) f_7. (e) f_9. (f) f_{10}. (g) f_{11}. (h) f_{15}. (i) f_{21}. (j) f_{23}.

particles have diversity in the population and the balance between exploration and exploitation during the iteration is achieved by the strategies of the modified position-updated equation (i.e., equation (26)) and nonlinear control parameter (i.e., equation (27)).

To further investigate the optimization performance of the DGWO on some standard and complex benchmark problems, we compared it with the TACPSO, IPSO, and GWO on a CEC2014 benchmark test suite with the dimension 30. The parameter settings of the DGWO and other selected algorithms were the same as mentioned above. The maximum number of iterations for the DGWO was 5×10^4, and for each problem, 20 independent runs were implemented. The experiment results are shown in Table 13.

From Table 13, it can be seen that, of the unimodal test functions (f_1–f_3), the proposed DGWO achieves the best performance on f_1 and f_2. Of the 13 multimodal functions (f_4–f_{16}), the DGWO shows better results on 11 benchmark test functions and similar results on two test functions (i.e., f_{13} and f_{16}). Of the 6 hybrid functions (f_{17}–f_{22}), the DGWO gives the best results on four functions (f_{17}, f_{19}, f_{21}, and f_{22}), while it becomes the second best algorithm for function f_{18}. Of the 8 composition functions (f_{23}–f_{30}), the proposed DGWO gives the best results on all test functions except for function f_{26} but provides the worst result on function f_{26}.

From the statistical analysis listed in Table 13, the DGWO performs better than the TACPSO at a significance level of 0.05 and better than the IPSO and GWO at a significance level of 0.1.

4.7. Experiment on Real-World Engineering Problems.

In this section, several classic real-world engineering optimization problems were selected to validate the practical optimization performance of our proposed algorithm. The DGWO and GWO methods were applied to solve three well-known constrained engineering design problems, including Himmelblau's problem, the gear train design, and the pressure vessel design. It is noted that the penalty function methods were employed to address the constrained optimization

Table 13: Comparison between the DGWO and other algorithms on CEC2014 benchmark functions.

Function	TACPSO Mean ± St. dev.	IPSO Mean ± St. dev.	GWO Mean ± St. dev.	DGWO Mean ± St. dev.
f_1	1.30E + 08 ± 9.99E + 07	7.75E + 07 ± 2.13E + 07	8.59E + 07 ± 6.67E + 07	5.79E + 07 ± 2.74E + 07
f_2	1.64E + 10 ± 1.10E + 10	2.55E + 09 ± 2.69 E + 09	2.31E + 09 ± 2.32E + 09	2.18E + 09 ± 2.02E±09
f_3	5.18E + 04 ± 2.89E + 04	8.77E + 03 ± 7.99E + 03	3.39E + 04 ± 8.81E + 03	3.56E + 04 ± 9.79E + 03
f_4	1.98E + 03 ± 1.49E + 03	6.64E + 02 ± 8.69E + 01	6.55E + 02 ± 1.07E + 02	6.63E + 02 ± 1.17E + 02
f_5	5.21E + 02 ± 1.91E − 01	5.20E + 02 ± 2.83E − 01	5.21E + 02 ± 5.65E − 02	5.20E + 02 ± 5.74E − 02
f_6	6.24E + 02 ± 2.79E + 00	6.19E + 02 ± 3.24E + 00	6.14E + 02 ± 3.42E + 00	6.13E + 02 ± 2.57E + 00
f_7	8.79E + 02 ± 7.77E + 01	7.26E + 02 ± 1.95E + 01	7.17E + 02 ± 1.27E + 01	7.10E + 02 ± 1.21E + 01
f_8	9.23E + 02 ± 2.23E + 01	8.76E + 02 ± 1.62E + 01	8.77E + 02 ± 2.04E + 01	8.67E + 02 ± 2.37E + 01
f_9	1.07E + 03 ± 2.66E + 01	1.02E + 03 ± 3.08E + 01	9.98E + 02 ± 2.24E + 01	9.76E + 02 ± 2.08E + 01
f_{10}	4.49E + 03 ± 5.73E + 02	3.68E + 03 ± 6.16E + 02	3.18E + 03 ± 6.01E + 02	3.11E + 03 ± 4.96E + 02
f_{11}	5.22E + 03 ± 6.52E + 02	4.52E + 03 ± 5.19E + 02	3.94E + 03 ± 7.09E + 02	3.79E + 03 ± 5.93E + 02
f_{12}	1.21E + 03 ± 3.09E − 01	1.20E + 03 ± 3.13E − 01	1.20E + 03 ± 1.06E + 00	1.20E + 03 ± 1.15E + 00
f_{13}	1.30E + 03 ± 1.29E + 00	1.30E + 03 ± 8.80E − 01	1.30E + 03 ± 1.08E − 01	1.30E + 03 ± 1.41E − 01
f_{14}	1.45E + 03 ± 2.09E + 01	1.41E + 03 ± 1.17E + 01	1.40E + 03 ± 5.06E + 00	1.40E + 03 ± 4.99E + 00
f_{15}	4.17E + 04 ± 7.37E + 04	1.64E + 03 ± 2.10E + 02	1.67E + 03 ± 4.72E + 02	1.57E + 03 ± 4.15E + 02
f_{16}	1.61E + 03 ± 5.48E − 01	1.61E + 03 ± 5.32E − 01	1.61E + 03 ± 7.61E − 01	1.61E + 03 ± 6.28E − 01
f_{17}	4.14E + 06 ± 2.87E + 06	5.17E + 06 ± 7.06E + 05	2.31E + 06 ± 3.48E + 05	2.13E + 06 ± 3.93E + 05
f_{18}	9.93E + 07 ± 2.70E + 08	6.91E + 06 ± 2.61E + 05	9.28E + 07 ± 2.00E + 07	2.14E + 07 ± 3.13E + 06
f_{19}	2.01E + 03 ± 6.35E + 01	1.94E + 03 ± 3.13E + 01	1.94E + 03 ± 2.46E + 01	1.94E + 03 ± 3.00E + 01
f_{20}	1.74E + 04 ± 1.73E + 04	5.26E + 03 ± 2.62E + 03	2.01E + 04 ± 7.77E + 03	1.95E + 04 ± 9.63E + 03
f_{21}	7.15E + 05 ± 8.92E + 05	1.68E + 05 ± 1.21E + 05	7.15E + 05 ± 9.20E + 04	1.42E + 05 ± 2.92E + 04
f_{22}	2.86E + 03 ± 1.96E + 02	2.71E + 03 ± 1.53E + 02	2.60E + 03 ± 1.35E + 02	2.56E + 03 ± 1.76E + 02
f_{23}	2.68E + 03 ± 3.88E + 01	2.63E + 03 ± 1.21E + 01	2.64E + 03 ± 1.18E + 01	2.63E + 03 ± 1.18E + 01
f_{24}	2.68E + 03 ± 1.64E + 01	2.64E + 03 ± 1.19E + 01	2.60E + 03 ± 1.68E − 03	2.60E + 03 ± 1.53E − 03
f_{25}	2.72E + 03 ± 5.08E + 00	2.71E + 03 ± 5.77E + 00	2.71E + 03 ± 4.82E + 00	2.71E + 03 ± 5.24E + 00
f_{26}	2.72E + 03 ± 6.35E + 01	2.73E + 03 ± 6.23E + 01	2.74E + 03 ± 4.87E + 01	2.75E + 03 ± 5.08E + 01
f_{27}	3.78E + 03 ± 8.68E + 01	3.53E + 03 ± 2.48E + 02	3.35E + 03 ± 1.06E + 02	3.35E + 03 ± 1.19E + 02
f_{28}	4.50E + 03 ± 6.12E + 02	4.36E + 03 ± 3.87E + 02	4.03E + 03 ± 3.52E + 02	3.96E + 03 ± 2.93E + 02
f_{29}	1.64E + 07 ± 1.47E + 07	2.18E + 07 ± 1.68E + 07	6.46E + 06 ± 2.21E + 06	1.01E + 06 ± 2.13E + 06
f_{30}	1.14E + 05 ± 7.79E + 04	6.04E + 04 ± 6.63E + 04	4.09E + 04 ± 1.72E + 04	4.07E + 04 ± 2.37E + 04
Losses (−)	26	19	20	
Wins (+)	2	4	2	
Approximations (≈)	2	7	8	
Detected differences (α)	0.05	0.1	0.1	

problems [55]. DGWO and GWO parameters for these three real-world engineering optimization applications were provided as follows: the population size was 30, the maximum number of iterations was 1000, and each problem was run independently 30 times.

4.7.1. Himmelblau's Nonlinear Optimization Problem. Himmelblau's problem is a well-known benchmark nonlinear constrained optimization problem that was developed by Himmelblau [56]. This type of optimization is performed to find the decision vector $Y = [y_1, y_2, y_3, y_4, y_5]$. To obtain the minimize function *f*, the objective function *f* is modeled as

$$\begin{aligned} \text{minimize} \quad & f(Y) = 5.3578547 y_3^2 + 0.8356891 y_1 y_5 \\ & \qquad + 37.293239 y_1 - 40792.141 \\ \text{s.t.} \quad & 0 \le g_1(Y) \le 92 \\ & 90 \le g_2(Y) \le 110 \\ & 20 \le g_3(Y) \le 25, \end{aligned} \tag{28}$$

where

$$\begin{aligned} g_1(Y) &= 85.334407 + 0.0056858 y_2 y_5 + 0.0006262 y_1 y_4 \\ &\quad - 0.0022053 y_3 y_5, \\ g_2(Y) &= 80.51249 + 0.0071317 y_2 y_5 + 0.0029955 y_1 y_2 \\ &\quad - 0.0021813 y_3^2, \\ g_3(Y) &= 9.300961 + 0.0047026 y_3 y_5 + 0.0012547 y_1 y_3 \\ &\quad + 0.0019085 y_3 y_4, \\ & 78 \le y_1 \le 102,\ 33 \le y_2 \le 45,\ 27 \le y_3, y_4, y_5 \le 45. \end{aligned} \tag{29}$$

Several researchers have employed different algorithms to solve this problem, such as the generalized reduced gradient (GRG) [56], the genetic algorithm (GA) [57], GA solution based on a global reference (GA-G) [58], and GA solution based on a local reference (GA-L) [58]. Table 14 illustrates the results of the best run obtained by the DGWO and the previously mentioned methods. Table 14 reveals that the results achieved by employing the DGWO algorithm are better than those of the previously reported best feasible

Table 14: Comparison results of the best Himmelblau's nonlinear optimization problem obtained by different algorithms.

Methods	Design variables					$f(Y)$	Constraints		
	y_1	y_2	y_3	y_4	y_5		$0 \leq g_1 \leq 92$	$90 \leq g_2 \leq 110$	$20 \leq g_1 \leq 25$
GRG [8]	78.62	33.44	31.07	44.18	35.22	−30373.95	90.52	NA	20.13
GA [9]	80.39	35.07	32.05	40.33	33.34	−30005.70	91.66	99.53690	20.03
GA-G [10]	80.61	34.21	31.34	42.05	34.85	−30175.80	92.00	NA	20.00
GA-L [10]	81.49	34.09	31.24	42.20	34.37	−30182.27	90.57	NA	20.12
GWO	78.01	33.00	30.0063	45.00	36.7570	−30662.42	92.00	94.91	20.00
Present study	78.00	33.00	30.0468	45.00	36.71	−30653.30	91.99	94.89	20.01

NA: not available.

solution and that the DGWO could provide very competitive results compared to the GWO.

4.7.2. Gear Train Design Problem. The gear train design problem has four integer variables and was initially introduced by Sandgran [59]. The task of solving this problem is to determine the optimal number of teeth of the gearwheel between 12 and 60 to minimize the gear ratio of the gear train displayed in Figure 8. The optimization model of this problem, with the decision vector $Y = [T_d, T_b, T_a, T_f] = [y_1, y_2, y_3, y_4]$, is formulated as follows:

$$\begin{aligned} \text{minimize} \quad f(Y) &= \left(\frac{1}{6.931} - \frac{y_1 y_2}{y_3 y_4}\right)^2, \\ 12 \leq y_1, y_2, y_3, y_4 &\leq 60, y_i \in \mathbb{Z}^+, \end{aligned} \tag{30}$$

where the gear ratio $= y_1 y_2 / y_3 y_4$.

Table 15 shows the optimization results of the best run of the gear train design problem, which is solved by different algorithms and the proposed DGWO algorithm. The statistical results of these algorithms and the results of the GSA-GA and CS algorithms proposed by Gandomi et al. [60] and Garg [61], which are shown in Table 16 with studies [60, 61], conclude that the result proposed by the DGWO algorithm is superior to the results of the two algorithms, and the worst (Worst), mean (Mean), and standard deviation (St. dev.) are low. The results obtained by the DGWO are slightly better than those by the GWO and are significantly better than those reported by different methods in [59–62].

4.7.3. Pressure Vessel Design Problem. In this problem, a cylindrical pressure vessel is mounted on both ends by hemispherical balls, and its cylinder is formulated by combining two longitudinal welds, as described in Figure 9 [63]. The four decision variables, which include the thickness of the pressure vessel (T_s), thickness of the head (T_h), inner radius of the vessel (R), and length of the cylindrical section of the vessel (L), are selected to be optimized to achieve the minimized total cost of the pressure vessel. Therefore, the formulation for this problem consists of four variables $Y = [T_s, T_h, R, L]$, which are modeled as follows:

$$\begin{aligned} \text{minimize} \quad f(Y) &= 0.6224 y_1 y_3 y_4 + 1.7781 y_2 y_3^2 \\ &\quad + 3.1661 y_1^2 y_4 + 19.84 y_1^2 y_3 \\ \text{s.t.} \quad & g_1(Y) = -y_1 + 0.0193 y_3 \leq 0 \\ & g_2(Y) = -y_2 + 0.00954 y_3 \leq 0 \\ & g_3(Y) = -\pi y_3^2 y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \leq 0 \\ & g_4(Y) = y_4 - 240 \leq 0 \\ & 1 \times 0.0625 \leq y_1, y_2 \leq 99 \times 0.0625 \quad 10 \leq y_3, y_4 \leq 200. \end{aligned} \tag{31}$$

The results obtained for this problem are computed by the proposed DGWO method and are compared with the results of the best run achieved by other algorithms in Table 17. The practical optimization performance of the DGWO algorithm is superior to that of existing approaches but slightly worse than that of the GWO. The statistical results after 30 independent runs are recorded in Table 18, which further validates the finding that the standard deviation of the proposed DGWO method is less than that of other algorithms except for the result reported in [64] and the worst result is better than that of the compared algorithms. In addition, the DGWO could obtain very close best and average results to the GWO and is better than other compared algorithms.

4.8. Several Insights for Applying the DGWO Algorithm. As discussed above, the optimization performance of the DGWO algorithm has been validated on several classical well-known benchmark functions. As seen from Table 3, the different position weight values w_1 and w_2 play an important role in improving the optimization performance of different types of problems. If the objective problems are a kind of unimodal or multimodal problem (such as f_1–f_{13}), the values of w_1 and w_2 can be 0.1 and 0.9 or 0.3 and 0.7, respectively, and both of these two strategies can obtain relatively high-quality solutions. If the objective problems are a

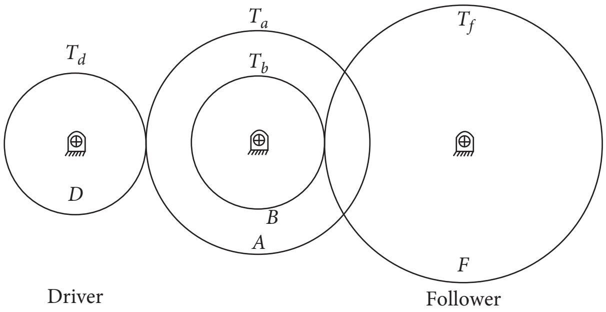


Figure 8: Structure of the gear train design problem.

Table 15: Comparison results of the best gear train design problem obtained by different algorithms.

Variables	Sandgren [59]	Deb and Goyal [62]	Gandomi et al. [60]	Garg [61]	GWO	Present study
$T_d(y_1)$	18	19	19	19	19	19
$T_b(y_2)$	22	16	16	16	16	16
$T_a(y_3)$	45	49	43	43	43	43
$T_f(y_4)$	60	43	49	49	49	49
Gear ration	0.15	0.14	0.14	0.14	0.14	0.14
$f(Y)$	5.71×10^{-6}	2.70×10^{-12}	2.70×10^{-12}	2.70×10^{-12}	2.70×10^{-12}	2.70×10^{-12}

Table 16: Statistical results of different algorithms for the gear train design problem.

Algorithms	Best	Worst	Mean	St. dev.
Gandomi et al. [60]	2.70×10^{-12}	2.36×10^{-9}	1.98×10^{-9}	3.56×10^{-9}
Garg [61]	2.70×10^{-12}	$3.\ 30\times10^{-9}$	1.22×10^{-9}	8.77×10^{-10}
GWO	2.70×10^{-12}	1.36×10^{-9}	6.25×10^{-9}	6.22×10^{-10}
Present study	2.70×10^{-12}	1.36×10^{-9}	5.65×10^{-10}	5.72×10^{-10}

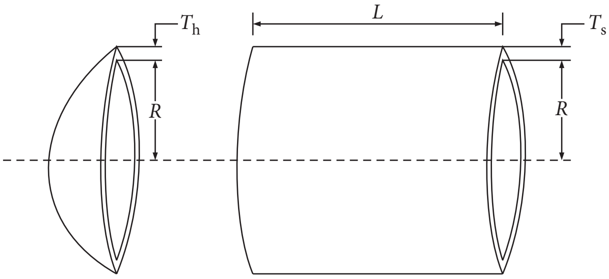


Figure 9: Structure of the pressure vessel design problem.

Table 17: Comparison results of the best pressure vessel design problem obtained by different algorithms.

Algorithms	Decision variables				Cost
	y_1	y_2	y_3	y_4	$f(Y)$
Sandgren [59]	1.13	0.63	47.70	117.70	8129.10
Kannan and Kramer [63]	1.13	0.63	58.29	43.69	7198.04
Coello [64]	0.81	0.44	40.32	200.00	6288.75
He and Wang [65]	0.81	0.44	42.09	176.75	6061.08
Kaveh and Talatahari [66]	0.81	0.44	42.10	176.57	6059.09
Gandomi et al. [60]	0.81	0.44	42.10	176.64	6059.71
GWO	0.78	0.39	40.36	199.46	5890.18
Present study	0.80	0.40	41.58	183.25	5938.78

kind of fixed-dimension multimodal problem (such as f_{14}–f_{23}), $w_1 = 0.1$ and $w_2 = 0.9$ can achieve better results than other position weight values. In addition, from Table 4, we can observe that the DGWO-1 algorithm, which employed the modified position-updated equation (i.e., equation (26)) and the linear control parameter $\overrightarrow{a}$, can

Table 18: Statistical results of different algorithms for the pressure vessel design problem.

Algorithms	Best	Worst	Mean	St. dev.
Sandgren [59]	8129.10	N/A	N/A	N/A
Kannan and Kramer [63]	7198.04	N/A	N/A	N/A
Coello [64]	6288.75	6308.15	6293.84	7.41
He and Wang [65]	6061.08	6363.80	6147.13	86.46
Kaveh and Talatahari [66]	6059.09	6135.33	6075.26	41.68
Gandomi et al. [60]	6059.71	6495.35	6447.74	502.69
GWO	5890.18	6674.46	5938.88	141.64
Present study	5938.78	6113.47	6013.93	85.53

N/A: not available.

provide very competitive results for unimodal and multimodal problems and slightly poorer results on fixed-dimension multimodal problems. Therefore, if the objective problems are unimodal or multimodal problems, the control parameter of the DGWO can adopt both $\vec{a}$ and $\vec{a}'$; otherwise, if the objective problems are fixed-dimension multimodal problems, we recommend that the practitioners use the nonlinear control parameter strategy proposed in this paper (i.e., equation (27)).

5. Conclusions

In this paper, an improved version of the GWO (referred to as the DGWO) algorithm is proposed to solve continuous numerical optimization problems. First, the DDS method is introduced into the GWO algorithm to perturb a set of dimensions of the first three best solutions to increase the diversity of a particle and to enhance the exploration ability of the GWO algorithm. This method realizes the predation mode of freely switching between direct encirclement and spiral walking. Second, the position interaction information about the three leaders (i.e., α, β, and δ) in the predation process is further considered, and the position-updated equation that is based on this information is proposed to increase the ability of the GWO algorithm to jump out of the local optimum. Finally, the proposed nonlinear control parameter strategy is designed to enhance the exploitation ability of the GWO algorithm, as well as the convergence precision and convergence rate. Based on the three improvements to the GWO algorithm, the balance between exploration and exploitation, convergence precision, and convergence rate have been enhanced. Twenty-three benchmark test problems, the CEC2014 benchmark suite, and three classic real-world engineering design applications were employed to verify the practical optimization performance of the proposed DGWO technique. First, the experimental results on unimodal functions show the exploitation ability of the DGWO, which helps accelerate the convergence speed and enhance the solution accuracy. Second, the exploration capability of the DGWO was demonstrated by the results on the multimodal functions. Third, the results from fixed-dimension multimodal functions and composite functions show that the DGWO succeeds in jumping out of local optima by balancing the exploration and exploitation. The simulations confirmed that the DGWO could find very competitive optimization results compared to recent GWO variants and state-of-the-art heuristic algorithms. However, the optimization performance of the DGWO algorithm for Himmelblau's nonlinear engineering design problem is not very competitive but shows excellent results for the gear train design problem and the pressure vessel design problem. Although several experiments have demonstrated that the DGWO is efficient, effective, and robust, it also has several obvious shortcomings, such as the greater number of parameters that need to be adjusted compared with the original GWO algorithm, the poor optimization performance on the complex problems that are included in the CEC2014 suite, and the percentage of problems solved is not 100% guaranteed.

In future works, there are two main aspects that need to be implemented. An interesting research point is to further simplify the spiral walking predation technique and to improve the positional interaction information strategy to propose a variant of the GWO with a simpler algorithm structure and higher optimization performance. In addition, we intend to utilize the proposed DGWO algorithm for solving multiobjective optimization problems and economic load dispatch problems and training neural networks in our future research.

Data Availability

The MATLAB code used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded in part by the National Natural Science Foundation of China (grant no. 71841054), the Young Innovative Talents Training Program for Universities in Heilongjiang Province (grant no. UNPYSCT-2018151), the Fundamental Research Funds for the Central Universities (grant no. 2572018BM07), the Natural Science Foundation of Heilongjiang Province of China (grant no. LH2019G014), the Postdoctoral Foundation of Heilongjiang Province of China (grant no. LBH-Z18015), and the 2019 Annual Basic Project of the Party's Political Construction Research Center of the Ministry of Industry and Information Technology (grant no. 19GZY411).

References

[1] I. Boussaïd, J. Lepagnot, and P. Siarry, "A Survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.

[2] G. M. Portal, M. Ritt, L. M. Borba, and L. S. Buriol, "Simulated annealing for the machine reassignment problem," *Annals of Operations Research*, vol. 242, no. 1, pp. 93–114, 2016.

[3] C. D. Tarantilis and C. T. Kiranoudis, "A threshold-accepting metaheuristic method for scheduling the operations of dehydration plants," *Drying Technology*, vol. 20, no. 6, pp. 1143–1160, 2002.

[4] Z. H. Jia and Y. Liao, "Based on micro canonical annealing and improved FCM algorithm for remote sensing image segmentation," *Laser Journal*, vol. 32, no. 2, pp. 20-21, 2011.

[5] X. Li, C. Yue, Y. P. Aneja, S. Chen, and Y. Cui, "An iterated tabu search metaheuristic for the regenerator location problem," *Applied Soft Computing*, vol. 70, pp. 182–194, 2018.

[6] A. Naanaa and S. Belghith, "An efficient guided local search approach for multiuser detection in UWB systems," *Physical Communication*, vol. 26, pp. 141–148, 2018.

[7] B. A. Tolson and C. A. Shoemaker, "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration," *Water Resources Research*, vol. 43, no. 1, pp. 208–214, 2007.

[8] S. Kang, S. Lee, and T. Kang, "Development and application of storage-zone decision method for long-term reservoir operation using the dynamically dimensioned search algorithm," *Water Resources Management*, vol. 31, no. 1, pp. 219–232, 2017.

[9] F.-R. Lin, N.-J. Wu, C.-H. Tu, and T.-K. Tsay, "Automatic calibration of an unsteady river flow model by using dynamically dimensioned search algorithm," *Mathematical Problems in Engineering*, vol. 2017, Article ID 7919324, 19 pages, 2017.

[10] B. A. Tolson, M. Asadzadeh, H. R. Maier, and A. Zecchin, "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization," *Water Resources Research*, vol. 45, no. 12, 2009.

[11] 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," *Scientific Programming*, vol. 2019, Article ID 2401818, 20 pages, 2019.

[12] C. Lu, L. Gao, and J. Yi, "Grey wolf optimizer with cellular topological structure," *Expert Systems With Applications*, vol. 107, pp. 89–114, 2018.

[13] J. R. Fernández, J. A. López-Campos, A. Segade, and J. A. Vilán, "A genetic algorithm for the characterization of hyperelastic materials," *Applied Mathematics and Computation*, vol. 329, pp. 239–250, 2018.

[14] F. Vesting and R. E. Bensow, "Particle swarm optimization: an alternative in marine propeller optimization?," *Engineering Optimization*, vol. 50, no. 1, pp. 70–88, 2018.

[15] E. G. Martinez-Soltero and J. Hernandez-Barragan, "Robot navigation based on differential evolution," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 350–354, 2018.

[16] J. Xu and F. Yan, "Hybrid Nelder–Mead algorithm and Dragonfly algorithm for function optimization and the training of a multilayer perceptron," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3473–3487, 2019.

[17] D. R. Edla, A. Lipare, and R. Cheruku, "Shuffled complex evolution approach for load balancing of gateways in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 4, pp. 3455–3476, 2018.

[18] S. Gholizadeh and H. Poorhoseini, "Optimum design of steel frame structures by a modified dolphin echolocation algorithm," *Structural Engineering and Mechanics*, vol. 55, no. 3, pp. 535–554, 2015.

[19] S. Gholizadeh and A. M. Shahrezaei, "Optimal placement of steel plate shear walls for steel frames by bat algorithm," *The Structural Design of Tall and Special Buildings*, vol. 24, no. 1, pp. 1–18, 2015.

[20] H. Sajedi, "Image steganalysis using artificial bee colony algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 29, no. 5, pp. 949–966, 2017.

[21] X. Song, L. Tang, S. Zhao et al., "Grey wolf optimizer for parameter estimation in surface waves," *Soil Dynamics and Earthquake Engineering*, vol. 75, pp. 147–157, 2015.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[23] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.

[24] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for grey wolf optimizer," *Expert Systems With Applications*, vol. 113, pp. 481–498, 2018.

[25] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, 2018.

[26] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, Article ID 7950348, 16 pages, 2016.

[27] A. Kishor and P. K. Singh, "Empirical study of grey wolf optimizer,," Edited by M. Pant, K. Deep, J. Bansal, A. Nagar, and K. Das, Eds., in *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, vol. 436, pp. 1037–1049, Springer, Singapore, 2015.

[28] Q. Luo, S. Zhang, Z. Li, and Y. Zhou, "A novel complex-valued encoding grey wolf optimization algorithm," *Algorithms*, vol. 9, no. 1, pp. 1–23, 2015.

[29] B. Yang, X. Zhang, T. Yu, H. Shu, and Z. Fang, "Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine," *Energy Conversion and Management*, vol. 133, pp. 427–443, 2017.

[30] J. Xu, F. Yan, O. Grace Ala, L. Su, and F. Li, "Chaotic dynamic weight grey wolf optimizer for numerical function optimization," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 2, pp. 2367–2384, 2019.

[31] Z.-M. Gao and J. Zhao, "An improved grey wolf optimization algorithm with variable weights," *Computational Intelligence & Neuroscience*, vol. 2019, Article ID 2981282, 13 pages, 2019.

[32] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.

[33] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, "Feature subset selection approach by gray-wolf optimization," *Advances in Intelligent Systems and Computing*, vol. 334, pp. 1–13, 2015.

[34] S. Mirjalili, "How effective is the grey wolf optimizer in training multi-layer perceptrons," *Applied Intelligence*, vol. 43, no. 1, pp. 150–161, 2015.

[35] S. A. Bessedik, R. Djekidel, and A. Ameur, "Performance of different kernel functions for LS-SVM-GWO to estimate

flashover voltage of polluted insulators," *IET Science, Measurement & Technology*, vol. 12, no. 6, pp. 739–745, 2018.

[36] V. Kumar, J. K. Chhabra, and D. Kumar, "Grey wolf algorithm based clustering technique," *Journal of Intelligent Systems*, vol. 26, no. 1, pp. 153–168, 2017.

[37] 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, 2019.

[38] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Advances in Engineering Software*, vol. 99, pp. 121–136, 2016.

[39] P.-W. Tsai, T.-T. Nguyen, and T.-K. Dao, "Robot path planning optimization based on multiobjective grey wolf optimizer," *Advances in Intelligent Systems and Computing*, vol. 536, pp. 166–173, 2017.

[40] J. Sujitha and K. Baskaran, "Genetic grey wolf optimizer based channel estimation in wireless communication system," *Wireless Personal Communications*, vol. 99, no. 2, pp. 965–984, 2018.

[41] B. Y. Qu, J. J. Liang, Z. Y. Wang, Q. Chen, and P. N. Suganthan, "Novel benchmark functions for continuous multimodal optimization with comparative results," *Swarm and Evolutionary Computation*, vol. 26, pp. 23–34, 2016.

[42] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80, 2018.

[43] M. J. F. G. Macêdo, M. F. P. Costa, A. M. A. C. Rocha, and E. W. Karas, "Combining filter method and dynamically dimensioned search for constrained global optimization," in *Computational Science and Its Applications—ICCSA*, pp. 119–134, 2017.

[44] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[45] J. Zhang, "Optimal control problem of converter steelmaking production process based on operation optimization method," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 483674, 13 pages, 2015.

[46] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[47] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263, 2015.

[48] S. Mirjalili, A. Lewis, and A. S. Sadiq, "Autonomous particles groups for particle swarm optimization," *Arabian Journal for Science and Engineering*, vol. 39, no. 6, pp. 4683–4697, 2014.

[49] Z. Cui, J. Zeng, and Y. Yin, "An improved PSO with time-varying accelerator coefficients," in *Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications*, pp. 638–643, Kaohsiung, Taiwan, November 2008.

[50] G. Q. Bao and K. F. Mao, "Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 2134–2139, Guilin, China, December 2009.

[51] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *Proceedings of the Asia-Pacific Conference on Information Processing*, pp. 330–332, Shenzhen, China, July 2009.

[52] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: A hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.

[53] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[54] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[55] A.-q. Xing, "Applications of the penalty function method in constrained optimal control problems," *Journal of Applied Mathematics and Simulation*, vol. 2, no. 4, pp. 251–265, 1989.

[56] D. M. Himmelblau, *Applied Nonlinear Programming*, McGraw-Hill, New York, NY, USA, 1972.

[57] H. Garg, "Solving structural engineering design optimization problems using an artificial bee colony algorithm," *Journal of Industrial and Management Optimization*, vol. 10, no. 3, pp. 777–794, 2013.

[58] H. Abdollah, X. Q. Charlene, and H. L. Steven, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.

[59] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.

[60] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[61] H. Garg, "A hybrid GSA-GA algorithm for constrained optimization problems," *Information Sciences*, vol. 478, pp. 499–523, 2019.

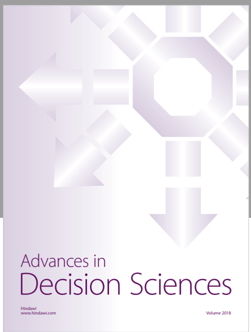
[62] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

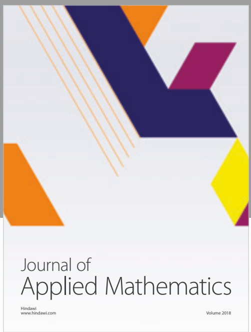
[63] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, no. 2, pp. 318–320, 1994.

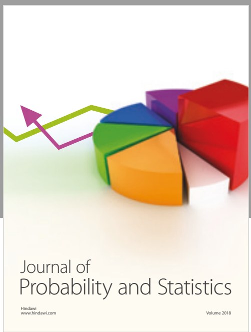
[64] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

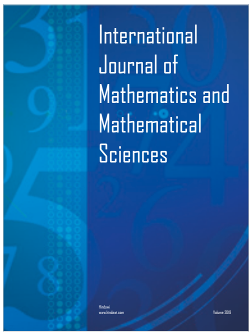
[65] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.

[66] A. Kaveh and S. Talatahari, "Engineering optimization with hybrid particle swarm and ant colony optimization," *Asian Journal of Civil Engineering*, vol. 10, pp. 611–628, 2009.









Hindawi

Submit your manuscripts at
www.hindawi.com