

Research Article

Equalized Grey Wolf Optimizer with Refraction Opposite Learning

Lijun Sun ^{1,2,3} Binbin Feng ^{1,2,3} Tianfei Chen ^{1,2,3} Dongliang Zhao ^{1,2,3}
and Yan Xin ^{1,2}

¹Key Laboratory of Grain Information Processing and Control, Ministry of Education, Henan University of Technology, Zhengzhou 450001, China

²Zhengzhou Key Laboratory of Machine Perception and Intelligent System, Henan University of Technology, Zhengzhou 450001, China

³College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

Correspondence should be addressed to Tianfei Chen; chen_tianfei@163.com

Received 2 December 2021; Revised 14 March 2022; Accepted 1 April 2022; Published 11 May 2022

Academic Editor: Maciej Lawrynczuk

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

Grey wolf optimizer (GWO) is a global search algorithm based on grey wolf hunting activity. However, the traditional GWO is prone to fall into local optimum, affecting the performance of the algorithm. Therefore, to solve this problem, an equalized grey wolf optimizer with refraction opposite learning (REGWO) is proposed in this study. In REGWO, the issue about the low swarm population variety of GWO in the late iteration is well overcome by the opposing learning of refraction. In addition, the equilibrium pool strategy reduces the likelihood of wolves going to the local extremum. To investigate the effectiveness of REGWO, it is evaluated on 21 widely used benchmark functions and IEEE CEC 2019 test functions. Experimental results show that REGWO performs better than the other competitors on most benchmarks.

1. Introduction

Complex optimization problems in practice are often discontinuous, nondifferentiable, and nonconvex. Prior to the advent of metaheuristic optimization technology [1], the most widely used optimization methods were the gradient descent algorithm and the Gauss-Newton method [2]. However, the gradient-based optimization approach is prone to obtaining the local extremum interference, resulting in a reduction in optimization precision. On the contrary, metaheuristic optimization algorithms are able to identify optimum or near-optimum solutions within an acceptable time. Therefore, many scholars have researched metaheuristic optimization algorithms in order to address challenging optimization problems, such as particle swarm algorithm (PSO) [3], artificial bee colony algorithm (ABC) [4], ant colony optimization algorithm (ACO) [5], slime mould algorithm (SMA) [6], hunger games search (HGS) [7], Runge-Kutta method (RUN) [8], grey wolf optimizer (GWO) [9], weighted mean of vectors (INFO) [10], and so on. Among these algorithms, GWO has become the focus of research in

recent years due to its advantages, such as few parameters and straightforward principle.

GWO is a swarm intelligence stochastic optimization algorithm introduced in 2014, as a result of information interaction between the social level and hunting behavior of grey wolves in nature [9]. GWO is an effective metaheuristic, and it attracted the interest of academics when it was initially introduced. It has been widely used in many fields such as feature selection [11, 12], image processing [13, 14], path planning [15], weld shop inverse scheduling [16], and so on. In GWO, the search process is guided by the leading wolves in each iteration, which shows great convergence toward leading wolves. The leading wolves sometimes fall into the local extremum, especially in multimodal problems. However, when leading wolves get trapped at local optima, other individuals in the population are also vulnerable to local extremes. This is the cause of the decrease in population diversity. Therefore, the standard GWO suffers from the same issues as most swarm intelligence algorithms, such as lack of the population diversity and ease of falling into local optimum [17].

The motivation of this paper is to solve the above problem; in order to overcome these weaknesses, an equalized grey wolf optimizer with refraction opposite learning (REGWO) is proposed in this paper. In REGWO, two search strategies with different features are introduced to generate candidate solutions. Among them, the opposite learning of refraction strategy is inspired by the principle of light refraction in nature. This strategy is introduced to improve population diversity during the search and expand the scope of solution space. At the same time, the fuzzy theory is used to adjust the parameter so that the refraction solution is more random and the algorithm can find more potential solutions. Moreover, the equilibrium pool strategy is designed to impair the leadership of the leading wolves. This method can make wolves update their position following nonoptimal solution with a certain probability, so wolves have the ability to jump out of the local extremum even when the optimal solution falls into the local optimum. REGWO can achieve better performance by combining the aforementioned tactics.

The remainder of this paper is organized as follows. The related work is discussed in Section 2. Section 3 presents the original GWO algorithm. The proposed REGWO algorithm is introduced in Section 4. In Section 5, the performance of the proposed REGWO is evaluated on different benchmark functions; furthermore, the significance of the results is proved by statistical analysis. Finally, we end the paper with conclusions and future work in Section 6.

2. Related Work

The metaheuristic optimization algorithms have been widely used to solve optimization problems. These algorithms are divided into three categories: physics-based algorithms, evolutionary algorithms, and swarm intelligence algorithms. Physics-based algorithms mimic the physical rules in nature in which the individuals communicate around the search space by the physical concepts, such as inertia force, light refraction law, gravitational force, and so on. There are some popular algorithms in this category, such as atom search optimization (ASO) [18] and Henry gas solubility optimization [19]. Evolutionary algorithms are a kind of iterative optimization algorithms simulating natural evolutionary processes. The best individuals are combined to form a new generation, which is the main advantage of EAs because it promotes population improvement during iteration, such as genetic algorithm (GA) [20] and differential evolution (DE) [21]. Swarm intelligence algorithms are inspired by the collective behavior of swarm organisms, such as bird flocking, animal grazing, and so on. Individuals in a population with cooperation and interaction move collectively to the promising areas in the search space. Some recently proposed swarm-based intelligence algorithms are grey wolf optimizer (GWO) [9], monarch butterfly optimization (MBO) [22], moth search algorithm (MSA) [23], Harris hawks optimization (HHO) [24], colony predation algorithm (CPA) [25], and so on.

Swarm intelligence algorithms have been shown to be effective at solving optimization problems, but they may fall into local optimum and loss of diversity. As a result, some scholars have proposed modified variations to tackle the

flaws. DEWCO algorithm has improved the initial population through a hyperheuristic to increase its convergence speed [26]. EFSABC algorithm is proposed by a search strategy for group escape and foraging based on Levy flight to exit from local optima [27].

GWO is a kind of swarm intelligence algorithm, which imitates the social level of wolves and the group hunting behavior. It has fewer parameters and is easy to implement. Therefore, this algorithm has been widely used to solve different optimization problems, such as multidimensional knapsack problem [28], path planning [29], parameter estimation [30], economic dispatch [31], feature selection [32], large scale unit commitment problem [33], wind speed forecasting [34], and so on.

In recent years, numerous scholars have developed variants of the basic GWO to address the weaknesses of GWO and provide better performance. Three different position update methods are proposed [35], which are weighted average, fitness-based, and fuzzy logic. Further experimental analysis reveals that the GWO improved using the fuzzy logic method has better performance. In order to improve the search ability of grey wolf, a modified algorithm RW-GWO based on a random walk has been imported [36]. A cellular grey wolf optimizer with a topological structure (CGWO) is introduced. In CGWO, each wolf has its own topological neighbors, and interactions among wolves are restricted to their neighbors, which favors exploitation. Furthermore, the information diffusion mechanism by overlap among neighbors can allow maintaining the population diversity for longer, usually contributing to exploration [37]. Grey wolf optimizer with crossover and opposition-based learning (GWO-XOBL) is presented to the jump out local optima [38]. An improved grey wolf optimizer is proposed using the explorative equation and opposition-based learning (OBL) [39]. To get a more stable sense of balance between exploitation and exploration, a new modified GWO called memory-based grey wolf optimizer (mGWO) is introduced [40]. Randomized balanced grey wolf optimizer (RBGWO), which improves the overall efficiency of the search process by establishing a balance between its exploitation and exploration capability incorporating three successive enhancement strategies equipped with a social hierarchy mechanism and random walk with student's t-distributed random numbers [41]. By dividing the search process into three stages and using different population updating strategies at each stage, an improved GWO called multistage grey wolf optimizer (MGWO) is proposed; the MGWO is improved while maintaining a certain convergence speed [42].

Another area of interest for researchers is to combine other evolutionary algorithms or operators to improve the performance of GWO. PSO-GWO algorithm merged with PSO [43], the idea of PSO was introduced into the GWO to update the position information of each individual using the optimal value of the individual and the optimal value of the group, which enhanced the diversity of the population and improved the global search ability. The crossover operator is introduced into GWO to promote population diversity [44]. The purpose of the crossover operator is to enhance information sharing among individuals in the population. At

the same time, the search accuracy and convergence speed of the algorithm are improved. Grey wolf optimizer has been hybridized with differential evolution (DE) mutation, and two versions, namely DE-GWO and gDE-GWO, have been proposed to avoid the stagnation of the solution [45]. To improve the performance of the GWO, a new variant of the GWO called a mutation-driven modified grey wolf optimizer and denoted by MDM-GWO is proposed. The MDM-GWO combines a new update search mechanism, modified control parameter, mutation-driven scheme, and greedy approach of selection in the search procedure of the GWO [46]. SCGWO algorithm combines GWO with an improved spread strategy and a chaotic local search mechanism to accelerate the convergence rate of the evolving agents [47]. GWO variant enhanced with a covariance matrix adaptation evolution strategy, Levy flight mechanism, and orthogonal learning strategy named GWOCMALOL is proposed. GWOCMALOL algorithm uses these strategies to bring more effective exploratory inclinations [48].

According to the various improvement strategies mentioned above, the main aim of the GWO variants is to improve search accuracy and convergence speed. Although the above GWOs can overcome some drawbacks of the original GWO, GWO still faces the problem of poor global exploration ability in the late iteration. Therefore, an equalized grey wolf optimizer with refraction opposite learning (REGWO) is presented in this paper.

3. Grey Wolf Optimizer

GWO is a typical swarm intelligence optimization algorithm. The model of GWO originates from the leading class and hunting behavior of grey wolves. There is a clear division of labor and cooperation among grey wolf individuals. As shown in Figure 1, the grey wolf population is separated into four levels, namely α , β , δ , and ω wolves. The first layer is the α wolf, and the next layer is called the β wolf. The δ wolf is located in the third layer. The grey wolf in the population is called the ω wolf (search wolf), which is located in the bottom layer. The α , β , and δ wolves are called leading wolves, and their number is set to 1. In GWO, the ω wolf must update its position to obtain the optimal solution, and α , β , and δ wolves represent the optimal value, suboptimal value, and third optimal value, respectively. The hunting process of grey wolves is mainly guided by α , β , and δ wolves, and ω wolves update iteratively according to the position of leading wolves.

The formula of the grey wolf around prey can be expressed as follows [41]:

$$\begin{aligned} \mathbf{D} &= \mathbf{C} \circ \mathbf{X}_p(t) - \mathbf{X}(t), \\ \mathbf{X}(t+1) &= \mathbf{X}_p(t) - \mathbf{A} \circ \mathbf{D}. \end{aligned} \quad (1)$$

Here, t is the current number of iterations; \circ is the Hadamard product operation; \mathbf{X}_p and \mathbf{X} denote the position vector of the prey and a grey wolf, respectively; and the calculation formulas of random vectors \mathbf{A} and \mathbf{C} are as follows:

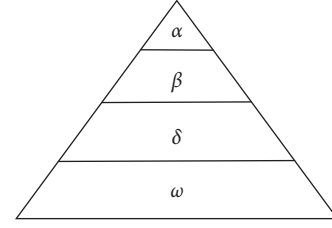


FIGURE 1: Four-layer pyramid model in GWO.

$$\begin{aligned} \mathbf{A} &= 2\mathbf{a} \circ \mathbf{r}_1 - \mathbf{a}, \\ \mathbf{C} &= 2\mathbf{r}_2, \end{aligned} \quad (2)$$

where \mathbf{r}_1 and \mathbf{r}_2 are random vectors between $[0, 1]$ and the vector \mathbf{a} is linearly decreased from 2 to 0 over the course of iterations.

To better understand the optimization rules of GWO, the possible areas are shown in Figure 2 when the position of the grey wolf is updated. It can be seen from Figure 2 that ω wolf can reach different positions around the prey by adjusting the values of parameters \mathbf{A} and \mathbf{C} . Furthermore, random variables \mathbf{r}_1 and \mathbf{r}_2 can assist search wolves to reach any of the points depicted in Figure 2. Parameters \mathbf{A} and \mathbf{C} are responsible for exploration and exploitation behavior in GWO. \mathbf{A} is a random value between $[-\mathbf{a}, \mathbf{a}]$. When $\mathbf{A} > 1$ and $\mathbf{C} > 1$, the population is inclined to exploration. In addition, when $\mathbf{A} < 1$ and $\mathbf{C} < 1$, the population is prone to exploitation.

The formulas of grey wolf tracking target prey are as follows [41]:

$$\begin{cases} \mathbf{D}_\alpha = \mathbf{C}_1 \circ \mathbf{X}_\alpha - \mathbf{X}, \\ \mathbf{D}_\beta = \mathbf{C}_2 \circ \mathbf{X}_\beta - \mathbf{X}, \\ \mathbf{D}_\delta = \mathbf{C}_3 \circ \mathbf{X}_\delta - \mathbf{X}, \\ \mathbf{X}_1 = \mathbf{X}_\alpha - \mathbf{A}_1 \circ \mathbf{D}_\alpha, \\ \mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \circ \mathbf{D}_\beta, \\ \mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3 \circ \mathbf{D}_\delta, \end{cases} \quad (3)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3}.$$

Here, \mathbf{D}_α , \mathbf{D}_β , and \mathbf{D}_δ denote the distance between α , β , and δ wolves and other individuals, respectively; \mathbf{X}_α , \mathbf{X}_β , and \mathbf{X}_δ represent the positions of α , β , and δ , severally, respectively; \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are random vectors; \mathbf{X}_1 , \mathbf{X}_2 , and \mathbf{X}_3 represent the step length and direction of ω wolf toward α , β , and δ , respectively; and ω wolf determines the final position according to equation (3).

4. Proposed Algorithm REGWO

To improve the global search ability of GWO, the opposite learning of refraction and equilibrium pool technique are introduced in this work. Opposite learning of refraction is chosen in the proposed algorithm to generate more potential solutions. Besides, the equilibrium pool strategy can achieve a better exploration by weakening the leadership of the leading wolf. The two strategies are introduced in Sections 4.1 and 4.2,

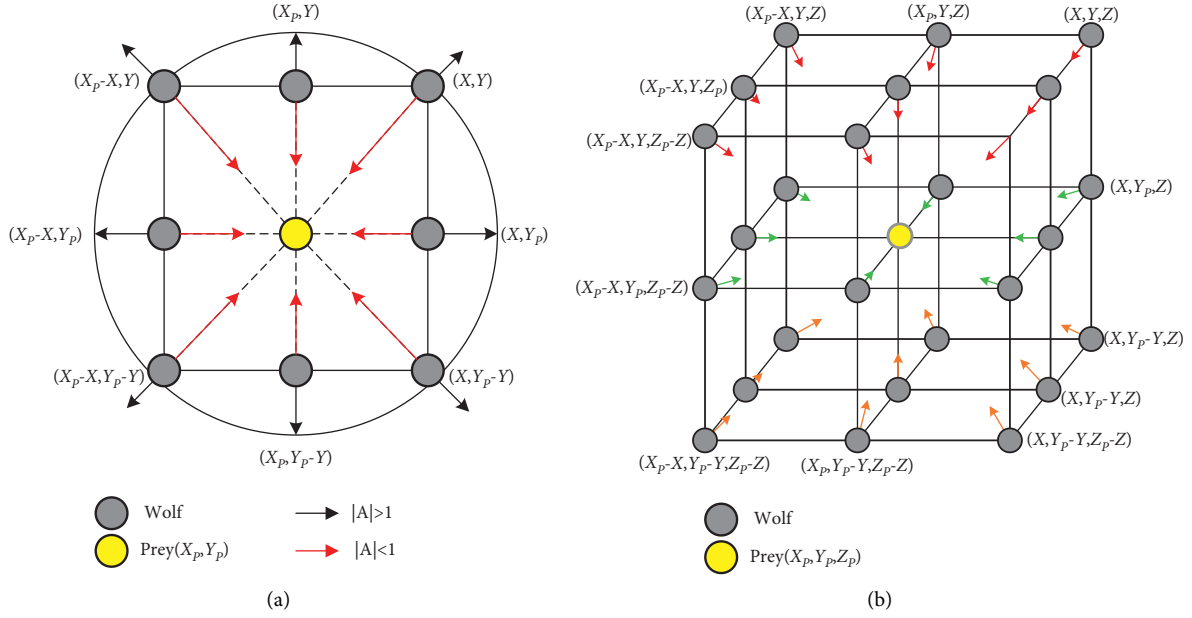


FIGURE 2: 2D and 3D position vectors and the possible next position of grey wolf: (a) 2D view of grey wolf in GWO and (b) 3D view of grey wolf in GWO.

respectively. The proposed algorithm is named REGWO and described in Section 4.3.

4.1. Opposite Learning of Refraction. The opposition-based learning (OBL) technique is proposed in 2005 [49]. The fundamental idea is to expand the search space of the population by calculating the opposite solution of the current solution, so as to select the candidate solution that is more suitable for the optimization problem. Applying this method to the optimization algorithm can effectively improve the search accuracy of the algorithm [50]. However, the standard OBL has certain shortcomings. OBL only speeds up the convergence of GWO and obtains only one opposite solution of fixed position. Therefore, the opposite solution may fall near the local optimal solution, causing the algorithm to fall into the local optimum with the iteration [51]. In order to tackle this problem, this paper introduces the refraction principle to improve the traditional opposite learning process. Refracted opposition-based learning (ROBL) strategy is based on the OBL, combined with the principle of light refraction to identify a better solution. ROBL strategy not only considers the opposite direction of individuals but also considers other directions of individuals. The schematic is shown in Figure 3.

In the one-dimensional space where the individual of the population is located, the x -axis is separated into the upper and lower parts. Above the x -axis is the natural vacuum part of the refraction model, and below the x -axis is the other propagation medium of the refraction model. In Figure 3, the search range of individuals on the x -axis is $[a, b]$, that is, $x \in [a, b]$, and the y -axis is normal. x' is the incident point of the light source, and the length of the incident light segment $x'o$ is denoted by h . The incident light refracts at the intersection o , and x'^* is the refraction point; the length of

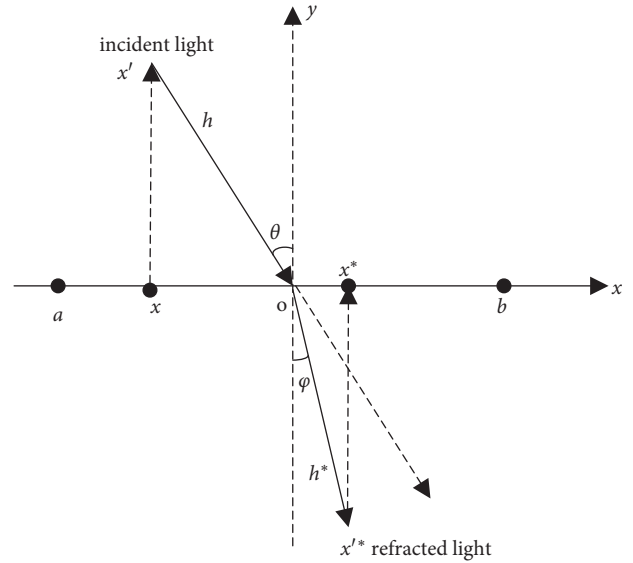


FIGURE 3: Refraction opposite learning model.

refraction light segment ox'^* is denoted by h^* ; and θ and φ are the incidence angle and the refraction angle, respectively. The x coordinate (position) of intersection o is $(a+b)/2$, which is the midpoint of the individual search range $[a, b]$. From the geometric relationship in Figure 3, we can obtain

$$\sin \theta = \frac{((a+b)/2 - x)}{h}. \quad (4)$$

$$\sin \varphi = \frac{(x^* - (a+b)/2)}{h^*}. \quad (5)$$

The refractive index ($n = \sin \theta / \sin \varphi$) of light obtained by equations (4) and (5) is

$$n = \frac{((a+b)/2 - x)/h}{((x^* - (a+b)/2)/h^*)}. \quad (6)$$

Let $k = h/h^*$; equation (6) can be transformed into

$$kn = \frac{((a+b)/2 - x)}{(x^* - (a+b)/2)}. \quad (7)$$

According to equation (7), we can obtain

$$x^* = \frac{(a+b)/2 - x}{kn} + \frac{a+b}{2}. \quad (8)$$

When n and k are both 1, equation (8) is the standard opposite learning formula:

$$x^* = a + b - x. \quad (9)$$

Obviously, the OBL strategy is a special case of the ROBL strategy. In order to improve the ability of GWO to jump out of the local extremum, the above ROBL model is applied to GWO. Since the individuals in GWO are multidimensional, equation (8) can be extended to the D -dimensional space as follows:

$$x_{i,j}^* = \frac{(a_j + b_j)/2 - x_{i,j}}{kn} + \frac{a_j + b_j}{2}. \quad (10)$$

Here, $x_{i,j}$ represents the value of the j -th dimension of the i -th individual; $x_{i,j}^*$ is the opposite solution obtained by the ROBL model; and a_j and b_j are the j -th dimension upper bound and lower bound, respectively.

As shown in Figure 3 and above, as the k value changes, the position of the opposite solution generated by (10) will be changed. That is to say, the adjustment of k improves the randomness of the solution. The k is calculated by

$$k(t) = k_{\max} - \frac{(k_{\max} - k_{\min})t}{T}, \quad (11)$$

where $k_{\max} = 1$, $k_{\min} = 0$, t denotes the current iteration number, and T is the total number of iterations.

Meanwhile, in order to make the decline rate of k well match the convergence rate of the fitness value, this paper proposes a method to adjust the k value as shown in Figure 4. The fuzzy membership degree $\mu_k(t)$ of k is

$$\mu_k(t) = \frac{k(t) - k_{\min}}{k_{\max} - k_{\min}}, \quad (12)$$

where $\mu_k(t) \in [0,1]$, $\mu_k(t)$ increases with the increase of $k(t)$, $\mu_k(t)$ decrease with the decrease of $k(t)$, and the optimal fitness relative change rate η is

$$\eta = \frac{f(t) - f(t-10)}{f(t-10)}. \quad (13)$$

Here, $f(t)$ is the objective function, namely the optimal fitness value of the t -th iteration of the population; $f(t-10)$ represents the optimal fitness value of the $(t-10)$ -th iteration of the population; then η denotes the relative change rate of the optimal fitness value in 10 iterations of evolution; and when η value is large, it indicates that the change rate of

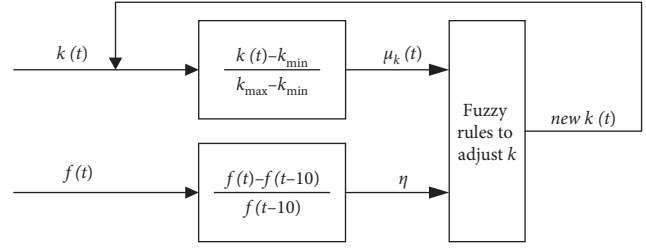


FIGURE 4: Fuzzy control structure.

the optimal fitness value is large, and at this time, the k value should be larger to improve global search capability; otherwise, when η value is small, it indicates that the change rate of the optimal fitness value is small, and at this time, the k value should be smaller. In the ROBL model, firstly, the current η and $\mu_k(t)$ are judged. Then, the k of the next iteration is adaptively adjusted by fuzzy rules, which can accelerate the convergence speed of the algorithm.

Fuzzy rules to adjust (k):

Rule 1. If $(\mu_k(t) > 0.5$ and $\eta > \gamma)$ or $(\mu_k(t) \leq 0.5$ and $\eta \leq \gamma)$, then $k(t) = \sqrt{(t_{\max} - t)/t_{\max}} \times (k_{\max} - k_{\min}) + k_{\min}$

Rule 2. If $(\mu_k(t) \leq 0.5$ and $\eta > \gamma)$, then $k(t) = \tau_1/4 + (k_{\max} + k_{\min})/2$

Rule 3. If $(\mu_k(t) > 0.5$ and $\eta \leq \gamma)$, then $k(t) = \tau_2/4 + k_{\min}$

Here, γ is the threshold, and its value is 0.05; τ_1 and τ_2 are the parameters between $[0, 1]$ in the rules.

On further observation of Figure 3, the purpose of k is to adjust the population, improve population diversity, expand the search space, and improve the global search ability of the algorithm. Therefore, the adjustment of parameter k based on fuzzy rules in the ROBL model can effectively improve the diversity of individual distribution in the search space. It makes up for the weak exploration ability of GWO in the late iteration. The advantage of the ROBL model over the OBL model is that the candidate solution can be obtained dynamically by parameter adjustment, which enhances the chance of the algorithm jumping out of the local optimum to a great extent. However, OBL can only obtain a fixed candidate solution. That is, parameter k has the ability to extend search space.

4.2. Equilibrium Pool Strategy. In GWO, the search is primarily guided by α , β , and δ wolves. If the leading wolves fall into the local optimum, the entire population will update their position in the direction of the local optimum. To address this issue, this paper introduces the equilibrium pool strategy to enhance population diversity [52]. The fundamental idea of the strategy is to calculate the fitness value of each individual after population initialization and choose three candidate solutions (X_1 , X_2 , and X_3) based on the fitness value. Among them, X_1 , X_2 , and X_3 represent α , β , and δ wolves, respectively. In addition, the average value of three candidate solutions is calculated as the average candidate solution X_{avg} , and then the equalization pool X_{pool} is constructed. The mathematical model is as follows:

```

Initialization parameters  $a$ ,  $A$ , and  $C$ , iteration number  $t$ , maximum iteration number  $T$ 
Initialize the location of the wolves  $x_i$  ( $i = 1, 2, \dots, N$ )
Calculate xi fitness values  $f(x_i)$ 
Select the former three-excellent solutions as  $\alpha$ ,  $\beta$ , and  $\delta$  wolves
while  $t < T$ 
  if rand  $> 0.5$ 
    for  $i = 1:N$ 
      Position update according to equation (3)
    end for
  else
    for  $i = 1:N$ 
      Position update according to equation (17)
    end for
  end if
  Calculate refraction solution  $x_i^*$  of individual  $x_i$  by equation (10)
  Calculate fitness values  $f(x_i)$  and  $f(x_i^*)$  of  $x_i$  and  $x_i^*$ , respectively
  if  $f(x_i) > f(x_i^*)$ 
     $x_i = x_i^*$ 
  end if
  Select the former three-excellent solution as  $\alpha$ ,  $\beta$ , and  $\delta$  wolves
   $t = t + 1$ 
end while
return  $\alpha$ 

```

ALGORITHM 1: Equalized grey wolf optimizer with refraction opposite learning (REGWO).

$$\begin{aligned} \mathbf{X}_{\text{pool}} &= (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_{\text{avg}}), \\ \mathbf{X}_{\text{eq}} &= \text{rand}(\mathbf{X}_{\text{pool}}). \end{aligned} \quad (14)$$

Here, three candidate solutions ($\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$) are contributed to exploitation and the average candidate solution (\mathbf{X}_{avg}) is contributed to exploration. ω wolf randomly selects candidate individuals from the candidate pool with equal probability for location updating. Besides, parameter F is used to balance exploration and exploitation, and the mathematical description is as follows:

$$\begin{aligned} F &= \text{sign}(r - 0.5)(e^{-\lambda m} - 1), \\ m &= (1 - t/T)^{(t/T)}, \end{aligned} \quad (15)$$

where λ is the random vector between $[0,1]$, r denotes a random number between $[0,1]$, $\text{sign}(r-0.5)$ is used to control the direction of exploration and exploitation, t represents the current number of iterations, and T notes the maximum number of iterations. In addition, the generation rate G is used to improve exploitation capability. The mathematical expression is as follows:

$$\begin{aligned} \mathbf{G} &= \mathbf{G}_0 \mathbf{F}, \\ \mathbf{G}_0 &= \begin{cases} \text{rand} \times (\mathbf{X}_{\text{eq}} - \lambda \mathbf{X}), & \text{rand} \geq 0.5 \\ 0, & \text{rand} < 0.5 \end{cases}, \end{aligned} \quad (16)$$

where \mathbf{X}_{eq} is a candidate solution randomly chosen from the equilibrium pool with equal probability.

In summary, when GWO applies equilibrium pool strategy, the position update formula of the individual is as follows:

$$\mathbf{X}(t+1) = \mathbf{X}_{\text{eq}} + (\mathbf{X}(t) - \mathbf{X}_{\text{eq}}) \mathbf{F} + \frac{\mathbf{G}}{\lambda} (1 - \mathbf{F}). \quad (17)$$

4.3. REGWO Algorithm. It is well known that exploration and exploitation are necessary for population-based optimization algorithms, such as PSO, ABC, ACO, and so on. In standard GWO, the issue is that since all of the other individuals are attracted toward leader wolves, they may converge prematurely without enough exploration of search space, that is, standard GWO is prone to premature convergence.

To improve the performance of GWO, each grey wolf obtains the opposite solution via the ROBL strategy, which enhances individual randomness. Refraction opposite learning strategy makes up for the shortcomings of traditional opposite learning, expands the search space, and effectively enhances population diversity. In addition, an equilibrium pool strategy is introduced to reduce the likelihood of the algorithm falling into the local extremum. The equilibrium pool retains four individuals, namely α, β , and δ wolves as well as their mean values. The ability of exploration is properly improved by randomly selecting an individual from the equalization pool to lead the position update of ω wolf. The process of REGWO is described in Algorithm 1, and the flowchart of the proposed REGWO algorithm is shown in Figure 5. In REGWO, the α, β , and δ wolves are chosen by population initialization and fitness calculations. Then, the position is updated with equal probability by equation (3) or (17). Finally, the refraction opposite solution and its fitness value are calculated by equation (10); furthermore, the refraction solution is retained if the refraction

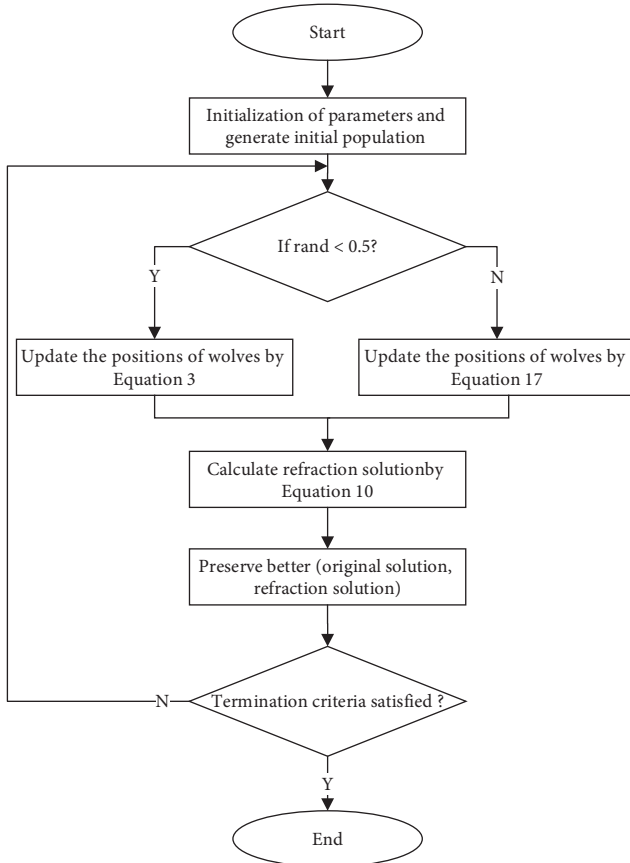


FIGURE 5: The flowchart of the REGWO algorithm.

solution is better than the original solution; otherwise, the original solution is retained. Until the end of the iteration, α wolf is the ultimate optimization result.

5. Experiments and Analysis

5.1. Experimental Settings. To fairly compare the performance of different algorithms, the function test set is needed. The numerical efficiency of REGWO developed in this paper was tested by solving 31 mathematical optimization problems. The first 21 benchmark functions are the classical functions utilized in literature [9, 53]; they ($f_1 \sim f_{21}$) are composed of unimodal, multimodal, fixed-dimensional multipeak, and shifted functions. The specific expressions and search intervals of these functions are shown in Table 1. The unimodal function ($f_1 \sim f_5$) with just one local and also global optimal solution is commonly used to evaluate the local exploitation ability of the algorithm; $f_{10} \sim f_{14}$ are multimodal function and often used to test the ability of the algorithm to explore. The f_{15} and f_{16} are fixed-dimensional multipeak functions with many extreme points but low dimensions, so it is easy to optimize and can be used to assess the stability of the algorithm. The last 5 functions in Table 1 are shifted functions, which are mainly to avoid the situation that some algorithms copy one parameter to another to generate a neighbor solution [53]. The other 10 test problems ($f_{22} \sim f_{31}$) considered in this paper (see Table 2) regard

composite benchmark functions considered in the IEEE CEC 2019 special session [54]. These benchmark functions are more complex than the first 21 benchmark functions, and $f_{22} \sim f_{31}$ are designed to have a minimum value of 1. The optimization performance of each algorithm can be further verified by solving complex problems.

Two sets of experiments are conducted in this paper. In the first experiment, REGWO is compared with some popular algorithms and novel algorithms to evaluate convergence speed and optimization accuracy. Furthermore, the comparisons of GWO, RGWOL (GWO improved only by refraction principle with linear control parameter k), RGWOF (GWO improved only by refraction principle with fuzzy control parameter k), EGWO (GWO improved only by equilibrium pool strategy), and REGWO are executed. The influences of two strategies and dynamic changes of parameter k on the optimization results can be observed through the experiment. In the second experiment, the comparisons of mean and standard deviation are performed between REGWO and the other six different GWO variants. Meanwhile, the population size is set to 30; the maximum number of iterations is set to 5000, and experimental results are based on 30 independent experiments.

5.2. Comparison with Swarm Intelligence Algorithms and Strategies Analysis. To validate the performance of the proposed REGWO algorithm in this paper, it is used to solve the functions $f_1 \sim f_{31}$, and the performance is compared with RGWOL, RGWOF, EGWO, and other swarm intelligence algorithms, including standard GWO [9], sparrow search algorithm (SSA) [55], Archimedes optimization algorithm (AOA) [56], particle swarm optimization (PSO) [57], firefly algorithm (FA) [58], and artificial bee colony (ABC) [4]. Among them, SSA and AOA are novel intelligent algorithms, and PSO, FA, and ABC are popular intelligent algorithms. For better comparison, the other parameters of algorithms are shown in Table 3.

The results on each benchmark function of the algorithms are shown in Table 4. It can be seen from Table 4 that the effectiveness of each strategy has been verified. RGWOL outperforms six test functions. RGWOF outperforms nine test functions. EGWO outperforms six test functions. Among them, the results of the RGWOF algorithm on most functions are better than RGWOL. The experimental results in Table 4 show that the effect of fuzzy control parameter k is better than the linear decrease of parameter k . Therefore, the REGWO algorithm proposed in this paper combines the equilibrium pool strategy (EGWO) with the refraction opposite learning strategy (RGWOF). At the same time, the fuzzy theory control parameter k is used in the refraction opposite learning. Although the RGWOF and EGWO can find the solutions, combining the two strategies has more benefits since the solutions obtained by REGWO are always better than theirs.

On the other hand, it can be seen from Table 4 that the average fitness of REGWO outperforms on 26 test functions compared with other swarm intelligence optimization algorithms. Astonishingly, REGWO can converge into the

TABLE 1: Benchmark functions ($f_1 \sim f_{21}$).

No.	Functions
f_1	$f_1(x) = \sum_{i=1}^n x_i^2$
f_2	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
f_3	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^j x_j)^2$
f_4	$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$
f_5	$f_5(x) = \sum_{i=1}^n ([x_i + 0.5])^2$
f_6	$f_6(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$
f_7	$f_7(x) = \sum_{i=1}^n (10^6)^{i-1/n-1} x_i^2$
f_8	$f_8(x) = \sum_{i=1}^n ix_i^2$
f_9	$f_9(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
f_{10}	$f_{10}(x) = -20 \exp(-0.2\sqrt{1/n \sum_{i=1}^n x_i^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + e$
f_{11}	$f_{11}(x) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10], y_i = \begin{cases} x_i, & x_i < 0.5 \\ \text{round}(2x_i)/2, & x_i \geq 0.5 \end{cases}$
f_{12}	$f_{12}(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$
f_{13}	$f_{13}(x) = 1/n \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$
f_{14}	$f_{14}(x) = \sum_{i=1}^n x_i \cdot \sin(x_i) + 0.1 \cdot x_i $
f_{15}	$f_{15}(x) = (1/500 + \sum_{j=1}^{25} 1/\sum_{i=1}^2 (x_i - a_{ij})^6)^{-1}$
f_{16}	$f_{16}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)].$
f_{17}	$f_{17}(x) = \sum_{i=1}^n z_i^2, z = x - o$
f_{18}	$f_{18}(x) = \sum_{i=1}^n (z_i^2 - 10 \cos(2\pi z_i) + 10), z = x - o$
f_{19}	$f_{19}(x) = 1/4000 \sum_{i=1}^n z_i^2 - \prod_{i=1}^n \cos(z_i/\sqrt{i}) + 1, z = x - o$
f_{20}	$f_{20}(x) = -20 \exp(-0.2\sqrt{1/n \sum_{i=1}^n z_i^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi z_i)) + e, z = x - o$
f_{21}	$f_{21}(x) = \sum z_i \cdot \sin(z_i) + 0.1 \cdot z_i , z = x - o$

TABLE 2: IEEE CEC 2019 test suite ($f_{22} \sim f_{31}$).

No.	Function names	$F_i^* = F_i(x^*)$	Dim	Range
f_{22} (F1)	Storn's Chebyshev polynomial fitting problem	1	9	[-8,192, 8,192]
f_{23} (F2)	Inverse Hilbert matrix problem	1	16	[-16,384, 16,384]
f_{24} (F3)	Lennard-Jones minimum energy cluster	1	18	[-4, 4]
f_{25} (F4)	Rastrigin's function	1	10	[-100, 100]
f_{26} (F5)	Griewank's function	1	10	[-100, 100]
f_{27} (F6)	Weierstrass function	1	10	[-100, 100]
f_{28} (F7)	Modified Schwefel's function	1	10	[-100, 100]
f_{29} (F8)	Expanded Schaffer's F6 function	1	10	[-100, 100]
f_{30} (F9)	Happy Cat function	1	10	[-100, 100]
f_{31} (F10)	Ackley function	1	10	[-100, 100]

TABLE 3: Parameter setting (PSO, FA, ABC, SSA, AOA, and GWO).

Algorithms	Parameter setting
PSO [57]	Inertia weight $w = 0.75$, $b_1 = b_2 = 2$, $v_{\max} = 0.1 * (x_{\max} - x_{\min})$
FA [58]	Light absorption coefficient $\zeta = 1$, step size $s = 0.2$
ABC [4]	Control parameter $limit = 0.6 * \text{population size} * \text{dim}$
SSA [55]	Discoverers $n = 0.2 * \text{population size}$
AOA [56]	Constants $d_1 = 2$, $d_2 = 6$, $d_3 = 1$, $d_4 = 2$
GWO [9]	Control parameter a decrease linearly from 2 to 0

global optimal solution in solving functions $f_1 \sim f_3, f_7, f_8$, and $f_{11} \sim f_{19}$, indicating that REGWO has the potential to converge to the global optimal value. On further observation in Table 4, the standard GWO is still competitive compared

with SSA and AOA for unimodal functions. It is proved that the standard GWO has good exploitation ability when solving unimodal functions. However, the optimization performance of standard GWO is relatively weak for multimodal functions, while REGWO shows better performance both in unimodal functions and multimodal functions. Moreover, it can be seen from the Friedman test average rank in Table 4 that the order from low to high is REGWO, RGWOF, EGWO, RGWOL, SSA, FA, GWO, AOA, PSO, and ABC. Obviously, REGWO has preferable competitiveness compared with novel algorithms (SSA, AOA) and the classical algorithms (PSO, FA, ABC). The superior performances of REGWO should be attributed to the improved strategies. Individuals maintain high diversity during optimization due to refraction opposite solution strategy, and the equalization pool strategy weakens the leadership of

TABLE 4: Search result (comparisons of PSO, FA, ABC, SSA, AOA, GWO, RGWOL, RGWOF, EGWO, and REGWO).

No.	Performance index	PSO	FA	ABC	SSA	AOA	GWO	RGWOL	RGWOF	EGWO	REGWO
f_1	Mean	1.96e-09	1.18e-86	2.91e-11	5.22e-191	3.09e-263	1.34e-311	0	0	0	0
	Std	4.81e-09	1.90e-87	3.05e-11	0	0	0	0	0	0	0
f_2	Mean	9.12e-02	4.56e-44	5.29e-12	1.41e-101	8.14e-165	4.26e-179	1.00e-323	0	3.60e-214	0
	Std	1.60e-01	2.49e-45	2.76e-11	7.73e-101	0	0	0	0	0	0
f_3	Mean	6.57e-10	1.19e-44	4.88e+04	1.05e-66	7.19e-60	1.28e-75	1.13e-92	0	1.28e-78	0
	Std	1.22e-09	3.76e-44	6.73e+03	3.33e-66	2.27e-59	4.06e-75	2.11e-92	0	4.06e-78	0
f_4	Mean	3.78e-06	1.99e+00	5.39e+01	2.21e-45	1.09e-35	1.75e-75	1.91e-241	3.32e-245	1.04e-84	5.06e-251
	Std	2.37e-06	2.16e+00	5.19e+00	6.99e-45	1.74e-35	4.81e-75	0	0	1.47e-84	0
f_5	Mean	5.31e-07	0	3.04e-11	2.61e-32	2.99e-01	4.75e-01	6.25e-01	5.24e-01	9.05e-07	5.01e-02
	Std	1.66e-06	0	2.54e-11	3.10e-32	3.06e-01	3.43e-01	3.17e-01	2.18e-01	3.44e-07	1.05e-01
f_6	Mean	5.32e-03	6.36e-04	2.11e-01	3.03e-04	1.02e-03	1.20e-04	7.23e-04	8.63e-05	2.74e-04	2.60e-05
	Std	2.93e-03	3.40e-04	4.83e-02	2.79e-04	5.02e-04	6.23e-05	5.53e-04	7.74e-05	1.37e-04	1.48e-05
f_7	Mean	5.13e+00	5.73e-82	1.82e-09	1.76e-90	5.07e-264	6.67e-306	0	0	0	0
	Std	2.80e+01	1.30e-82	2.59e-09	9.38e-90	0	0	0	0	0	0
f_8	Mean	1.36e-07	1.81e-87	1.17e-12	7.25e-168	1.37e-270	4.75e-312	0	0	0	0
	Std	4.32e-07	1.44e-88	9.66e-13	0	0	0	0	0	0	0
f_9	Mean	2.87e+01	7.71e-01	3.19e+03	3.12e-07	2.59e+01	2.66e+01	2.71e+01	2.70e+01	2.38e+01	2.88e+00
	Std	2.09e+01	1.74e+00	1.75e+03	6.46e-07	9.00e-01	8.27e-01	1.48e+00	3.59e-02	1.30e+00	5.51e-01
f_{10}	Mean	1.30e+00	2.18e-14	4.57e-05	8.88e-16	1.99e+01	7.99e-15	4.44e-15	4.44e-15	7.28e-15	4.44e-15
	Std	7.84e-01	5.41e-15	3.93e-05	0	1.46e-03	0	0	0	1.49e-15	0
f_{11}	Mean	5.92e+01	7.33e+01	1.96e+02	0	1.54e+01	4.66e-01	2.53e-02	7.21e-04	2.12e-03	0
	Std	1.76e+01	2.21e+01	2.18e+01	0	4.19e+00	1.77e+00	3.55e-02	4.65e-03	1.42e-03	0
f_{12}	Mean	2.15e-02	4.18e-03	8.41e-03	1.56e-05	1.09e-02	4.19e-04	0	0	7.39e-06	0
	Std	3.06e-02	7.94e-03	3.58e-02	5.77e-04	2.28e-02	2.29e-03	0	0	2.32e-05	0
f_{13}	Mean	-6.65e+01	-7.08e+01	-5.13e+01	-7.20e+01	-6.93e+01	-6.31e+01	-7.58e+01	-7.62e+01	-6.80e+01	-7.83e+01
	Std	2.76e+00	2.28e+00	1.79e+00	2.36e+00	2.83e+00	3.36e+00	3.82e-13	1.62e-13	2.23e+00	2.34e-14
f_{14}	Mean	4.84e-03	6.31e-15	2.26e+01	1.54e-17	2.89e-04	2.81e-170	1.37e-319	1.33e-322	8.72e-13	0
	Std	4.02e-03	3.27e-15	1.89e+00	2.53e-17	3.79e-04	0	0	0	2.75e-12	0
f_{15}	Mean	3.55e+00	1.06e+00	1.39e+00	6.51e+00	9.97e-01	4.26e+00	1.65e+00	6.21e+00	9.98e-01	9.98e-01
	Std	3.16e+00	3.62e-01	1.81e+00	5.76e+00	1.63e-06	4.23e+00	1.14e+00	5.59e+00	6.25e-13	1.82e-12
f_{16}	Mean	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
	Std	1.09e-15	1.22e-15	2.71e-07	9.70e-16	1.50e-07	2.66e-07	3.15e-07	3.73e-07	7.40e-16	4.44e-16
f_{17}	Mean	4.54e-10	1.20e-86	1.42e-10	4.30e-192	9.24e-269	1.15e-310	0	0	0	0
	Std	1.40e-09	2.11e-87	3.86e-10	0	0	0	0	0	0	0
f_{18}	Mean	5.23e+01	6.31e+01	2.12e+02	6.61e-04	1.42e+01	7.61e-02	3.51e-08	2.89e-10	3.51e-10	0
	Std	1.49e+01	2.01e+01	1.03e+01	4.97e-03	7.59e+00	4.97e-01	2.28e-07	1.53e-09	9.52e-09	0
f_{19}	Mean	3.61e-02	1.72e-03	5.27e-04	3.17e-06	2.10e-03	2.17e-04	8.27e-108	0	1.72e-03	0
	Std	2.93e-02	4.21e-03	4.80e-04	1.66e-06	6.67e-03	1.66e-04	0	0	5.44e-03	0
f_{20}	Mean	2.04e+00	2.57e-14	5.54e-05	8.88e-16	1.99e+01	7.99e-15	4.44e-12	4.44e-15	7.63e-15	4.44e-15
	Std	7.38e-01	8.53e-15	4.13e-05	0	1.93e-03	0	0	0	1.12e-15	0

TABLE 4: Continued.

No.	Performance index	PSO	FA	ABC	SSA	AOA	GWO	RGWOL	RGWOF	EGWO	REGWO
f_{21}	Mean	8.50e-11	5.26e-16	2.22e-02	9.54e-18	4.86e-03	5.87e-05	5.56e-11	2.62e-13	3.12e-08	1.54e-46
	Std	1.43e-10	4.41e-15	2.39e-01	1.79e-17	2.21e-02	2.28e-04	9.24e-10	7.55e-12	1.56e-07	8.43e-46
f_{22}	Mean	2.01e+04	2.55e+03	7.02e+06	1.00e+00	3.63e+02	4.77e+03	1.50e+00	1.21e+00	1.18e+03	1.00e+00
	Std	2.98e+04	4.41e+03	2.62e+06	0	9.35e+02	1.39e+04	7.65e-15	0	2.72e+03	0
f_{23}	Mean	2.48e+02	3.75e+02	7.63e+03	4.25e+01	1.99e+02	1.57e+02	6.27e+00	5.23e+00	7.59e+01	4.26e+00
	Std	1.22e+02	1.40e+02	1.13e+03	2.60e-01	1.29e+02	1.08e+02	4.79e-02	2.73e-02	5.39e+01	3.85e-02
f_{24}	Mean	4.36e+00	5.52e+00	1.20e+01	7.43e+00	2.95e+00	5.21e+00	4.59e+00	5.02e+00	4.42e+00	2.04e+00
	Std	3.13e+00	2.83e+00	4.12e-01	2.65e+00	1.86e+00	3.20e+00	1.08e+00	6.63e-01	2.67e+00	1.11e-00
f_{25}	Mean	2.52e+01	1.43e+01	2.72e+01	3.80e+01	3.02e+01	1.37e+01	1.45e+01	1.01e+01	7.76e+00	5.87e+00
	Std	1.09e+01	5.83e+00	3.62e+00	1.18e+01	7.47e+00	6.43e+00	8.07e+00	7.02e+00	2.60e+00	3.49e+00
f_{26}	Mean	1.14e+00	1.06e+00	1.29e+00	1.23e+00	1.83e+00	1.94e+00	1.62e+00	1.48e+00	1.19e+00	1.16e+00
	Std	4.71e-02	2.95e-02	3.47e-02	1.63e-01	2.20e-01	1.12e+00	1.98e-01	2.39e-01	6.74e-02	6.89e-02
f_{27}	Mean	2.83e+00	1.63e+00	9.18e+00	5.46e+00	3.53e+00	2.69e+00	2.61e+00	2.28e+00	2.26e+00	1.56e+00
	Std	1.38e+00	8.87e-01	7.59e-01	1.43e+00	1.85e+00	1.03e+00	6.85e-01	1.06e+00	1.33e+00	1.08e+00
f_{28}	Mean	8.45e+02	4.63e+02	1.48e+03	8.96e+02	5.42e+02	7.62e+02	5.31e+02	7.39e+02	2.75e+02	2.16e+02
	Std	2.72e+02	2.75e+02	1.70e+02	2.50e+02	2.09e+02	3.17e+02	3.59e+02	3.22e+02	1.72e+02	1.35e+02
f_{29}	Mean	4.06e+00	3.84e+00	5.07e+00	4.28e+00	3.47e+00	3.78e+00	3.77e+00	3.22e+00	3.25e+00	2.96e+00
	Std	4.90e-01	2.96e-01	7.67e-02	4.13e-01	6.17e-01	5.17e-01	3.27e-01	4.09e-01	5.21e-01	6.79e-01
f_{30}	Mean	1.14e+00	1.08e+00	1.18e+00	1.38e+00	1.19e+00	1.09e+00	1.11e+00	1.19e+00	1.10e+00	1.04e+00
	Std	3.76e+00	2.11e-02	2.31e-02	2.06e-01	1.09e-01	6.76e-02	6.13e-02	6.87e-02	2.63e-02	2.96e-02
f_{31}	Mean	2.09e+01	1.31e+01	2.12e+01	2.10e+01	2.11e+01	2.05e+01	1.88e+01	1.87e+01	1.94e+01	1.12e+01
	Std	2.88e-03	1.04e+01	7.46e-02	8.72e-02	1.37e-01	2.40e+00	5.28e+00	1.17e-01	6.06e+00	1.05e+01
Friedman test average rank		7.85	5.83	8.69	5.59	6.83	5.91	4.50	3.67	4.17	1.90

TABLE 5: Results of Wilcoxon signed-rank test (comparison with PSO, FA, ABC, SSA, AOA, GWO, RGWOL, RGWOF, and EGWO).

Case	+/-/-	R-	R+	p value
REGWO vs. PSO	30/1/0	21	444	1.36e-05
REGWO vs. FA	28/1/2	55	410	2.61e-04
REGWO vs. ABC	30/1/0	11	454	5.21e-06
REGWO vs. SSA	25/3/3	57	349	8.85e-04
REGWO vs. AOA	30/1/0	10	455	4.72e-06
REGWO vs. GWO	30/1/0	0	465	1.73e-06
REGWO vs. RGWOL	24/7/0	123	276	2.70e-05
REGWO vs. RGWOF	20/11/0	178	210	8.85e-05
REGWO vs. EGWO	25/6/0	96	311	6.45e-05

the optimal solution. Therefore, the combination of the two strategies can effectively improve the performance of standard GWO in solving multimodal functions. That is to say, the ability of the algorithm to jump out of the local extremum is enhanced.

Due to the stochastic nature of these algorithms, the statistical test is necessary for providing confidential comparisons [44]. Therefore, the Wilcoxon sign rank test is conducted in this paper. The test results of the REGWO algorithm and the other 9 selected algorithms on 31 test functions are shown in Table 5. The sign + (-) denotes that the REGWO algorithm is better (worse) than its compared algorithms. The symbol = indicates that the REGWO algorithm gets the same results as its competitors. It can be seen from Table 5 that REGWO provides higher R^+ values than R^- values in all cases. Moreover, the p values of 9 algorithms are less than 0.05, indicating that they are significantly different from REGWO, and REGWO is superior to other algorithms.

There are some reasons for REGWO has such a good performance. Firstly, the refraction principle and fuzzy control parameter are introduced in GWO; individual diversity is enhanced. RGWO improves the optimization precision and convergence speed of the standard GWO algorithm by retaining better (original solution and refraction inverse solution). Secondly, the equilibrium pool strategy enhances the global search ability of the original GWO algorithm by reducing the leadership of the leading wolves. The advantage of this strategy is particularly obvious when solving the multimodal function. Then, REGWO, as an improved GWO algorithm, combines the advantages of the two strategies. It not only improves the convergence speed of the GWO but also improves the optimization precision of the original GWO algorithm.

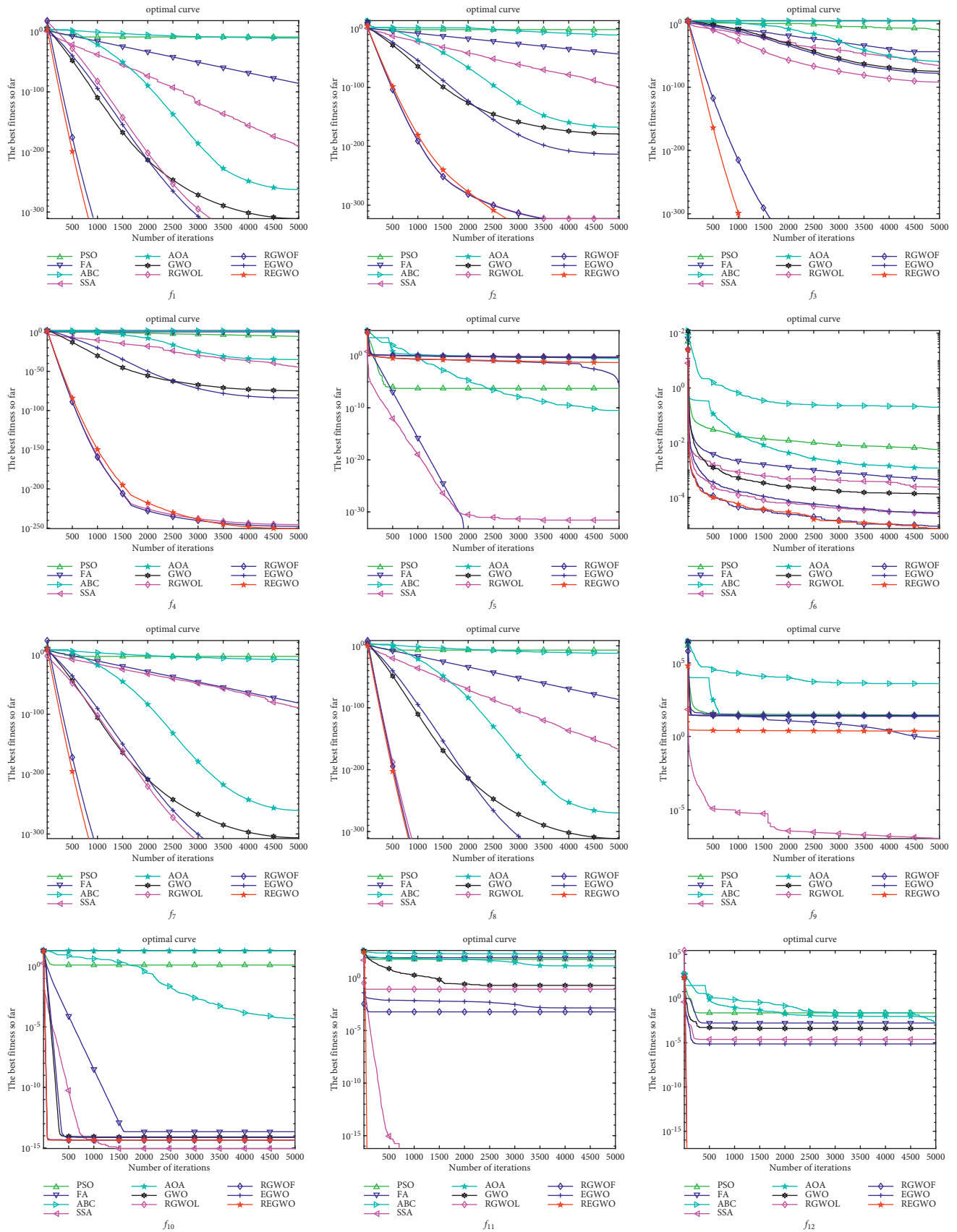
The convergence histories of the compared algorithms are shown in Figure 6. Through the convergence histories in Figure 6, we can find that the convergence speed of REGWO is faster than other swarm intelligence optimization algorithms on unimodal functions except f_5 . Although the convergence speed of the REGWO algorithm on multimodal function is not as fast as that on unimodal function, REGWO has high search precision than other algorithms. Especially, the optimization performance of REGWO is remarkable when solving more complex functions (IEEE CEC 2019 test

suite). It demonstrates that REGWO algorithm not only improves the convergence speed of the standard GWO algorithm on unimodal function but also enhances the optimization precision on complex functions.

5.3. *Comparison with GWOs.* To further validate the effectiveness of the REGWO. The performances of different GWO variants are compared, and the benchmark functions ($f_1 \sim f_{31}$) are solved by REGWO, WGWO [35], DGWO [59], AGWO [60], IGWO [61], RLGWO [62], and GNHGWO [63]. To make a fair comparison, the 6 algorithms use the same parameter settings as their original literature. Then, the results are analyzed by Friedman test average rank and Wilcoxon signed-rank test, and statistical results (mean cost and standard deviation) for 30 independent experiments are reported in Table 6.

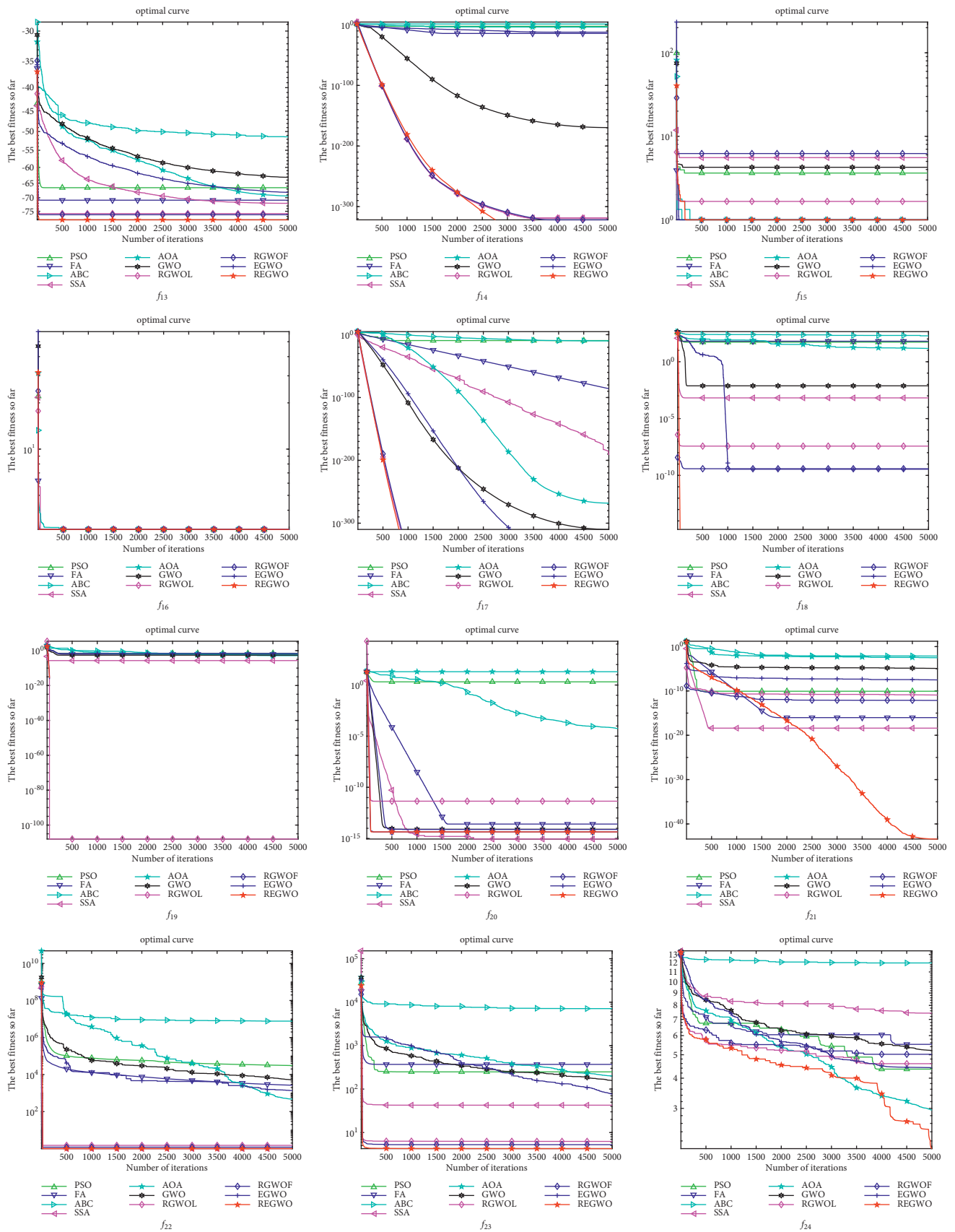
It can be seen from Table 6 that the average fitness of the REGWO is superior except for the functions f_5 , f_{10} , and f_{20} . In addition, the corresponding standard deviation is much smaller than other algorithms for most functions. The average fitness of the REGWO outperforms the other 6 enhanced GWOs on 14 benchmark functions. It can be seen that the combination of the refraction opposite learning approach and equilibrium pool strategy effectively improves the optimization accuracy of the standard GWO. The above algorithms can achieve the theoretical optimal value for the function f_{16} . Because f_{16} is a fixed-dimensional multimodal function with low dimension, it is simple to solve. However, REGWO shows better stability in terms of standard deviation. The $f_{22} \sim f_{31}$ are more complicated than the test functions listed in Table 1. It can better test the algorithm exploration and exploitation ability. Especially, REGWO converges to the theoretical optimal value 1 on function f_{22} . For functions f_{26} and f_{30} , the iterative optimization results of REGWO are also close to the theoretical optimal value 1. It demonstrates that REGWO still has the ability to converge to the global optimum for more complex mathematical optimization problems. Moreover, REGWO also achieves better performance in most functions in terms of standard deviation, indicating that REGWO has a better stability. From the Friedman test average rank in Table 6, the order from low to high is REGWO, IGWO, DGWO, RLGWO, AGWO, WGWO, and GNHGWO (IGWO is equal to DGWO). It shows that the performance of REGWO is much superior to other GWOs in accuracy. From the results of the Wilcoxon sign rank test in Table 7, REGWO provides higher R^+ values than R^- values in all cases. Moreover, it can be seen that the p values of WGWO, DGWO, AGWO, IGWO, RLGWO, and GNHGWO are less than 0.05, indicating that they are significantly different from REGWO, and REGWO is far superior to the other six algorithms.

In summary, REGWO has such good performance because of the contribution of the two strategies. Firstly, the diversity of solutions is increased by refractive opposite learning. Secondly, the equilibrium pool strategy weakens the leadership of the leading wolves to increase the probability of individuals jumping out of the local optimum. Therefore, the REGWO algorithm combining two strategies



(a)

FIGURE 6: Continued.



(b)

FIGURE 6: Continued.

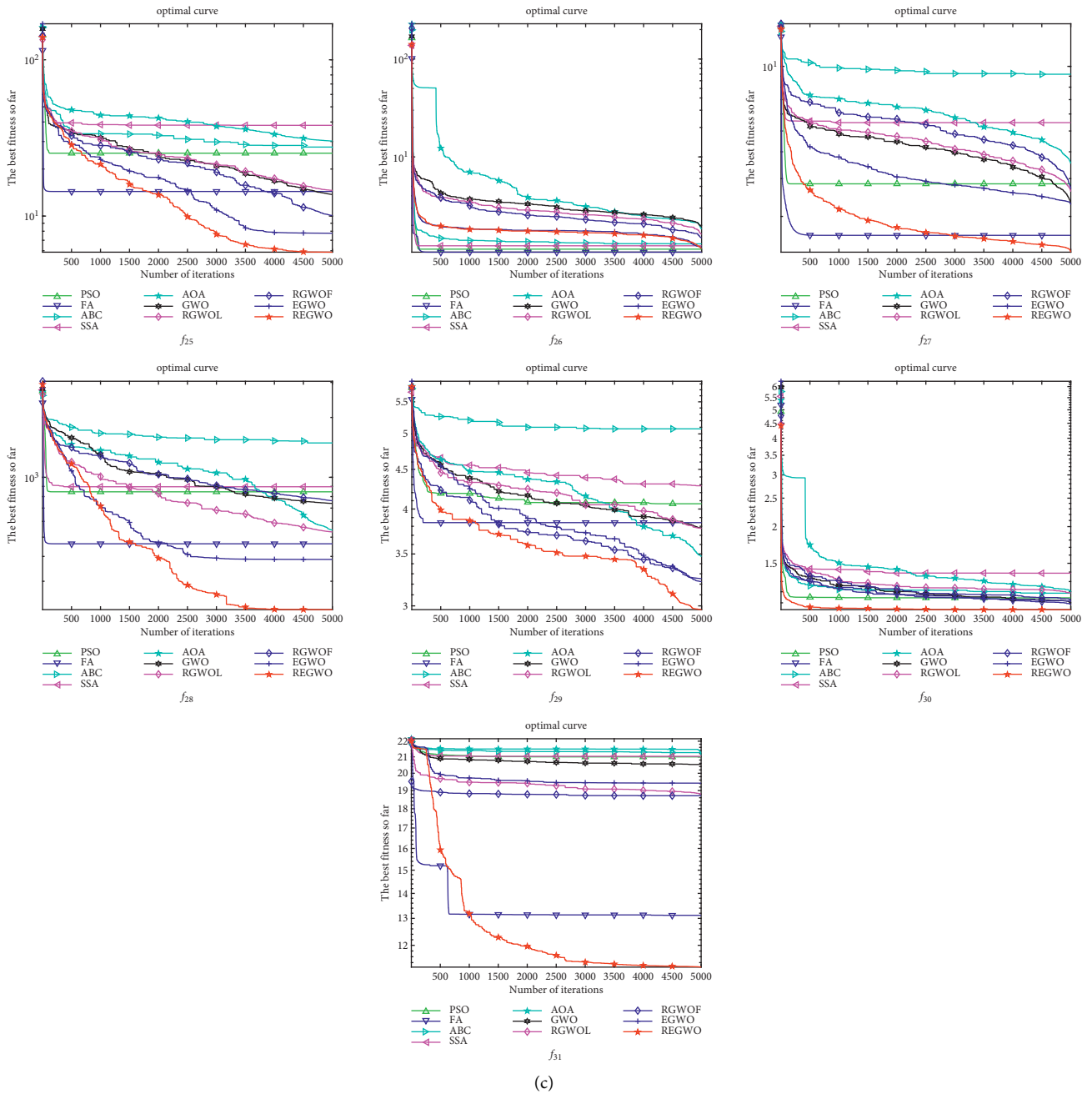


FIGURE 6: Convergence diagrams.

TABLE 6: Search result comparisons of GWOs.

No.	Performance index	WGWO	DGWO	AGWO	IGWO	RLGWO	GNHGWO	REGWO
f_1	Mean	0	0	0	$3.72e-317$	0	$1.02e-04$	0
	Std	0	0	0	0	0	$2.18e-04$	0
f_2	Mean	$2.41e-200$	0	$1.82e-281$	$1.69e-187$	0	$1.64e-04$	0
	Std	0	0	0	0	0	$2.93e-04$	0
f_3	Mean	$8.92e-98$	$1.80e-172$	$1.46e-144$	$1.53e-62$	0	$2.09e-03$	0
	Std	$2.82e-97$	0	$4.64e-144$	$4.53e-62$	0	$1.14e-03$	0
f_4	Mean	$5.31e-82$	$3.54e-151$	$3.00e-128$	$7.94e-60$	$3.54e-46$	$5.59e-03$	$3.06e-252$
	Std	$2.28e-81$	$1.11e-150$	$1.64e-128$	$3.25e-59$	$1.84e-45$	$1.83e-02$	0

TABLE 6: Continued.

No.	Performance index	WGWO	DGWO	AGWO	IGWO	RLGWO	GNHGWO	REGWO
f_5	Mean	5.90e-01	5.56e-01	1.03e+00	1.09e-27	4.75e-01	6.85e+00	1.66e-02
	Std	2.94e-01	3.55e-01	4.55e-01	9.26e-27	2.48e-01	3.08e-01	6.35e-02
f_6	Mean	1.53e-04	2.14e-04	3.59e-05	6.72e-04	8.89e-04	1.28e-05	7.43e-06
	Std	6.91e-04	1.14e-04	5.84e-05	9.98e-04	9.56e-04	1.61e-05	1.38e-06
f_7	Mean	0	0	0	0	0	3.52e-03	0
	Std	0	0	0	0	0	1.72e-02	0
f_8	Mean	0	0	0	0	0	3.15e-08	0
	Std	0	0	0	0	0	1.02e-07	0
f_9	Mean	5.68e+00	6.51e+00	6.00e+00	5.75e+00	6.75e+00	8.83e+00	2.99e+00
	Std	7.70E-01	6.68e-01	3.93e-01	4.23e+00	1.02e+00	3.51e-01	5.53e-01
f_{10}	Mean	7.51e-15	7.99e-15	6.21e-15	7.63e-15	1.24e-15	1.73e-03	4.44e-15
	Std	1.22e-15	0	1.77e-15	1.08e-15	1.12e-15	3.36e-03	0
f_{11}	Mean	0	0	0	3.70e+00	0	8.65e-06	0
	Std	0	0	0	1.82e+00	0	2.73e-05	0
f_{12}	Mean	0	8.20e-03	0	1.45e-02	0	6.51e-07	0
	Std	0	1.17e-02	0	1.39e-02	0	1.25e-06	0
f_{13}	Mean	-7.22e+01	-7.25e+01	-7.46e+01	-7.74e+01	-7.23e+01	-5.82e+01	-7.83e+01
	Std	3.43e+00	4.01e+00	2.80e+00	1.31e+00	3.80e+00	3.74e+00	1.25e+00
f_{14}	Mean	0	0	0	1.42e-07	4.63e-06	2.60e-07	0
	Std	0	0	0	4.51e-07	1.15e-05	5.86e-07	0
f_{15}	Mean	1.39e+00	4.53e+00	3.54e+00	9.98e-01	4.90e+00	1.06e+00	9.98e-01
	Std	8.36e-01	4.39e+00	3.91e+00	5.03e-01	5.02e+00	1.44e-01	1.81e-12
f_{16}	Mean	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
	Std	3.63e-07	5.57e-07	3.33e-07	2.17e-15	3.89e-07	1.94e-02	1.18e-15
f_{17}	Mean	0	0	0	0	0	4.66e-06	0
	Std	0	0	0	0	0	1.43e-05	0
f_{18}	Mean	0	0	0	3.97e-01	0	3.83e-10	0
	Std	0	0	0	1.20e-01	0	8.87e-10	0
f_{19}	Mean	7.44e-04	2.35e-03	0	2.02e-02	6.26e-06	5.09e-06	0
	Std	2.46e-03	3.61e-03	0	2.49e-02	4.12e-05	1.59e-05	0
f_{20}	Mean	4.44e-15	4.41e-15	4.44e-15	4.43e-15	8.88e-16	7.16e-07	4.44e-15
	Std	0	0	9.01e-16	0	1.70e-15	1.16e-04	0
f_{21}	Mean	1.77e-07	2.44e-06	9.87e-09	5.01e-14	6.54e-10	4.14e-06	1.54e-46
	Std	6.75e-07	1.26e-05	5.40e-08	2.58e-13	5.88e-09	6.09e-06	8.43e-46
f_{22}	Mean	2.85e+01	1.98e+04	8.64e+01	1.29e+03	1.00e+00	1.00e+00	1.00e+00
	Std	5.43e+01	6.25e+04	2.48e+02	1.90e+03	0	6.83e-06	6.37e-15
f_{23}	Mean	4.68e+00	4.62e+00	4.64e+00	4.31e+00	4.89e+00	4.98e+00	4.24e+00
	Std	3.53e-01	3.73e-01	3.98e-01	2.77e-01	2.61e-01	8.71e-02	4.77e-02
f_{24}	Mean	4.75e+00	3.27e+00	5.39e+00	7.19e+00	4.83e+00	9.07e+00	1.46e+00
	Std	3.01e+00	3.06e+00	3.54e+00	1.69e+00	3.28e+00	8.73e-01	4.87e-00
f_{25}	Mean	1.33e+01	1.64e+01	1.65e+01	8.19e+00	2.01e+01	7.50e+01	7.64e+00
	Std	5.30e+00	8.47e+00	8.05e+00	2.09e+00	6.01e+00	9.40e+00	1.14e+00
f_{26}	Mean	1.39e+00	1.42e+00	1.63e+00	1.42e+00	1.42e+01	4.74e+01	1.16e+00
	Std	2.30e-01	2.41e-01	7.57e-01	6.68e-02	3.91e+01	8.78e+00	8.47e-02
f_{27}	Mean	2.46e+00	2.17e+00	3.88e+00	1.51e+00	2.77e+00	1.00e+01	1.50e+00
	Std	8.17e-01	9.82e-01	3.57e-01	1.24e+00	1.65e+00	7.06e-01	6.23e-01
f_{28}	Mean	5.54e+02	6.64e+02	6.04e+02	2.82e+02	7.79e+02	1.60e+03	2.17e+02
	Std	2.41e+02	2.91e+02	2.37e+02	2.70e+02	3.37e+02	1.49e+02	2.19e+02
f_{29}	Mean	3.72e+00	3.73e+00	4.02e+00	3.18e+00	3.69e+00	4.96e+00	2.69e+00
	Std	6.63e-01	6.74e-01	4.03e-01	6.61e-01	4.71e-01	1.47e-01	7.66e-01
f_{30}	Mean	1.11e+00	1.10e+00	1.14e+00	1.09e+00	1.13e+00	2.80e+00	1.06e+00
	Std	5.04e-02	5.58e-02	3.89e-02	3.48e-02	6.24e-02	5.32e-01	2.88e-02
f_{31}	Mean	2.13e+01	2.13e+01	2.07e+01	1.99e+01	2.10e+01	2.13e+01	1.11e+01
	Std	9.81e-02	1.98e+00	9.55e-02	6.42e+00	1.38e-01	9.34e-02	9.80e+00
Friedman test average rank		3.85	3.98	4.09	3.98	4.04	6.09	1.93

TABLE 7: Results of Wilcoxon signed-rank test (comparison with GWOs).

Case	+/-/0	R-	R+	p value
REGWO vs. WGWO	21/10/0	127	231	5.95e-05
REGWO vs. DGWO	22/9/0	117	250	6.08e-05
REGWO vs. AGWO	20/11/0	146	210	8.85e-05
REGWO vs. IGWO	25/5/1	72	334	5.68e-05
REGWO vs. RLGWO	20/9/2	161	205	1.89e-04
REGWO vs. GNHGWO	29/2/0	21	435	2.56e-06

is able to achieve competitive optimization results on the test problems compared with other GWOs.

6. Conclusion and Future Work

In order to further improve the optimization performance of GWO, this paper proposes an equalized grey wolf optimizer with refraction opposite learning (REGWO). The main idea of the algorithm is to improve the opposite learning process of OBL based on the refraction principle of light. This strategy further expands the search space of the population, increases population variety, and enhances the ability of individuals to jump out of the local extremum. At the same time, the equilibrium pool strategy is combined to weaken the leadership of the leading wolves, which effectively avoids the situation that the rest of the individuals move to the leading wolves when the leading wolves fall into the local optimum. Therefore, the combination of the two strategies effectively enhances the exploration of GWO in the late iteration. In addition, REGWO is tested on 31 benchmark functions. The experimental results show that REGWO has higher convergence speed, search accuracy, and stability compared with standard GWO, other state-of-art GWOs, and other swarm intelligence algorithms. On the whole, REGWO is more effective in solving complex optimization problem.

In our future work, the selection of search strategies still needs to be further investigated. Furthermore, the REGWO algorithm can be extended to solve multiobjective optimization, binary optimization, and application-designed problems in the future.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 62173127, 61803146, and 61973104); the Scientific and Technological Innovation Leaders in Central Plains (No. 224200510008); the Henan

Excellent Young Scientists Fund (No. 212300410036); the Program for Science and Technology Innovation Talents in Universities of Henan Province (Grant no. 21HASTIT029); the Training Program for Young Backbone Teachers in Universities of Henan Province (Grant no. 2019GGJS089); the Key Science and Technology Projects in Henan Province (Grant nos. 212102210169 and 212102210086); the Innovative Funds Plans of Henan University of Technology (Grant no. 2020ZKCJ06); the Zhengzhou Science and Technology Collaborative Innovation Project (No. 21ZZXTCX06); the Cultivation Program of Young Backbone Teachers in Henan University of Technology (Grant chentianfei); and the Open Fund from Research Platform of Grain Information Processing Center in Henan University of Technology (Grant nos. KFJJ-2020-107, KFJJ-2020-111, and KFJJ-2020-114).

References

- [1] J. Del Ser, E. Osaba, D. Molina et al., "Bio-inspired computation: where we stand and what's next," *Swarm and Evolutionary Computation*, vol. 48, pp. 220–250, 2019.
- [2] H. k. Hjalmarsson, "Iterative feedback tuning: an overview," *International Journal of Adaptive Control and Signal Processing*, vol. 16, no. 5, pp. 373–395, 2002.
- [3] L. Li, Y. Liang, T. Li, C. Wu, G. Zhao, and X. Han, "Boost particle swarm optimization with fitness estimation," *Natural Computing*, vol. 18, no. 2, pp. 229–247, 2019.
- [4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [5] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [6] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [7] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.
- [8] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.
- [9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [10] I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, and A. H. Gandomi, "INFO: an efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, Article ID 116516, 2022.
- [11] E. M. R. Devi and R. C. Suganthe, "Feature selection in intrusion detection grey wolf optimizer," *Asian Journal of Research in Social Sciences and Humanities*, vol. 7, no. 3, pp. 671–682, 2017.
- [12] Q. Li, H. Chen, H. Huang et al., "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Computational and Mathematical Methods in Medicine*, vol. 15, Article ID 9512741, 2017.

- [13] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," *Expert Systems with Applications*, vol. 86, pp. 64–76, 2017.
- [14] L. Li, L. Sun, B. Xu, and S. Li, "Modified discrete grey wolf optimizer algorithm for multilevel image thresholding," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 3295769, 2017.
- [15] 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.
- [16] C. Wang, L. Zhao, X. Li, and Y. Li, "An improved grey wolf optimizer for welding shop inverse scheduling," *Computers & Industrial Engineering*, vol. 163, Article ID 107809, 2022.
- [17] Q. Tu, X. Chen, and X. Liu, "Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection," *IEEE Access*, vol. 7, pp. 78012–78028, 2019.
- [18] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems*, vol. 163, pp. 283–304, 2019.
- [19] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: a novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.
- [20] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [21] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [22] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2015.
- [23] G.-G. Wang, "Moth search algorithm: a bio-inspired meta-heuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2016.
- [24] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [25] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.
- [26] M. A. Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowledge-Based Systems*, vol. 172, pp. 42–63, 2019.
- [27] M. Chen, "Improved artificial bee colony algorithm based on escaped foraging strategy," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 6, pp. 516–524, 2019.
- [28] K. Luo and Q. Zhao, "A binary grey wolf optimizer for the multidimensional knapsack problem," *Applied Soft Computing*, vol. 83, Article ID 105645, 2019.
- [29] J. Liu, X. Wei, and H. Huang, "An improved grey wolf optimization algorithm and its application in path planning," *IEEE Access*, vol. 9, pp. 121944–121956, 2021.
- [30] D. Miao, W. Chen, and W. Zhao, "Parameter estimation of PEM fuel cells employing the hybrid grey wolf optimization method," *Energy*, vol. 193, Article ID 116616, 2020.
- [31] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, 2016.
- [32] P. M. Kitonyi, "Hybrid gradient descent grey wolf optimizer for optimal feature selection," *BioMed Research International*, vol. 2021, 2021.
- [33] L. K. Panwar and S. Reddy K, "Binary grey wolf optimizer for large scale unit commitment problem," *Swarm and Evolutionary Computation*, vol. 38, pp. 251–266, 2018.
- [34] J. Song, J. Wang, and H. Lu, "A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting," *Applied Energy*, vol. 215, pp. 643–658, 2018.
- [35] L. Rodríguez, O. Castillo, and J. Soria, "A fuzzy hierarchical operator in the grey wolf optimizer algorithm," *Applied Soft Computing*, vol. 57, pp. 315–328, 2017.
- [36] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.
- [37] C. Lu, L. Gao, and J. Yi, "Grey wolf optimizer with cellular topological structure," *Expert Systems with Applications*, vol. 107, no. 10, pp. 89–114, 2018.
- [38] S. Singh and J. Chand Bansal, "Grey wolf optimizer with crossover and opposition-based learning," in *Proceedings of the International Conference on Harmony Search Algorithm (ICHSA)*, pp. 401–410, Springer, Singapore, 2020.
- [39] J. C. Bansal and S. Singh, "A better exploration strategy in Grey Wolf Optimizer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1099–1118, 2021.
- [40] S. Gupta and K. Deep, "A memory-based Grey Wolf Optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, Article ID 106367, 2020.
- [41] J. Adhikary and S. Acharyya, "Randomized Balanced Grey Wolf Optimizer (RBGWO) for solving real life optimization problems," *Applied Soft Computing*, vol. 117, Article ID 108429, 2022.
- [42] H. Yu, J. Song, C. Chen, A. Zaguia, and M. Mafarja, "Image segmentation of Leaf Spot Diseases on Maize using multi-stage Cauchy-enabled grey wolf algorithm," *Engineering Applications of Artificial Intelligence*, vol. 109, Article ID 104653, 2022.
- [43] Z.-j. Teng, J.-l. Lv, and L.-w. Guo, "An improved hybrid grey wolf optimization algorithm," *Soft Computing*, vol. 23, no. 15, pp. 6617–6631, 2019.
- [44] A. Avadh Kishor and P. K. Singh, "Empirical study of grey wolf optimizer," in *Proceedings of the fifth international conference on soft computing for problem solving (SocProS)*, pp. 1037–1049, March 2016.
- [45] S. Gupta and K. Deep, "Hybrid grey wolf optimizer with mutation operator," *Advances in Intelligent Systems and Computing*, vol. 817, pp. 961–968, 2019.
- [46] S. Singh and J. C. Bansal, "Mutation-driven grey wolf optimizer with modified search mechanism," *Expert Systems with Applications*, vol. 109, Article ID 116450, 2022.
- [47] J. Hu, A. A. Heidari, and L. Zhang, "Chaotic diffusion-limited aggregation enhanced grey wolf optimizer: insights, analysis, binarization, and feature selection," *International Journal of Intelligent Systems*, 2021.
- [48] J. Hu, H. Chen, and A. A. Heidari, "Orthogonal learning covariance matrix for defects of grey wolf optimizer: insights, balance, diversity, and feature selection," *Knowledge-Based Systems*, vol. 213, Article ID 106684, 2021.
- [49] H. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Intelligent Agent, Web Technologies and Internet Commerce (IAWTIC'04)*, pp. 695–701, Vienna, Austria, November 2005.

- [50] M. Ghaemidizaji, C. Dadkhah, and H. Leung, "OHDA: an opposition based high dimensional optimization algorithm," *Applied Soft Computing*, vol. 91, no. 6, Article ID 106185, 2020.
- [51] P. Shao, Z. J. Wu, and X. Y. Zhou, "Improved particle swarm optimization algorithm based on opposite learning of refraction," *Acta Electronica Sinica*, vol. 43, no. 11, pp. 2137–2144, 2015.
- [52] A. Faramarzi, M. Heidarinejad, and B. Stephens, "Equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2020.
- [53] X. Y. Song, M. Zhao, and Q. F. Yan, "A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100549, 2019.
- [54] P. Bujok and A. Zamuda, "Cooperative model of evolutionary algorithms applied to CEC 2019 single objective numerical optimization," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, June 2019.
- [55] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [56] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.
- [57] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, 1995.
- [58] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [59] X. Zhang, Y. Zhang, and Z. Ming, "Improved dynamic grey wolf optimizer," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 6, pp. 877–890, 2021.
- [60] H. Mohammed, Qais, M. Hany, and A. Hasanien, "Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems," *Applied Soft Computing*, vol. 69, pp. 504–515, 2018.
- [61] M. H. Mohammad, N. Shahraki, and T. H. Shokooh, "An improved grey wolf optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, Article ID 113917, 2020.
- [62] W. Long, T. Wu, S. Cai, X. Liang, J. Jiao, and M. Xu, "A novel grey wolf optimizer algorithm with refraction learning," *IEEE Access*, vol. 7, pp. 57805–57819, 2019.
- [63] E. Akbari, A. Rahimnejad, and S. A. Gadsden, "A greedy non-hierarchical grey wolf optimizer for real-world optimization," *Electronics Letters*, vol. 57, no. 13, pp. 499–501, 2021.