

Research Article

Improved Whale Optimization Algorithm for Solving Constrained Optimization Problems

Gui-Ying Ning ¹ and Dun-Qian Cao ²

¹Liuzhou Institute of Technology, Liuzhou, Guangxi 545616, China

²College of Science, Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Gui-Ying Ning; guiyingning@126.com

Received 21 September 2020; Revised 22 January 2021; Accepted 26 January 2021; Published 9 February 2021

Academic Editor: Manuel De la Sen

Copyright © 2021 Gui-Ying Ning and Dun-Qian Cao. 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.

In view of the shortcomings of the whale optimization algorithm (WOA), such as slow convergence speed, low accuracy, and easy to fall into local optimum, an improved whale optimization algorithm (IWOA) is proposed. First, the standard WOA is improved from the three aspects of initial population, convergence factor, and mutation operation. At the same time, Gaussian mutation is introduced. Then the nonfixed penalty function method is used to transform the constrained problem into an unconstrained problem. Finally, 13 benchmark problems were used to test the feasibility and effectiveness of the proposed method. Numerical results show that the proposed IWOA has obvious advantages such as stronger global search ability, better stability, faster convergence speed, and higher convergence accuracy; it can be used to effectively solve complex constrained optimization problems.

1. Introduction

The whale optimization algorithm (WOA) was proposed by Mirjalili and Lewis [1] in 2016. It is a new swarm intelligence optimization algorithm that simulates humpback whale hunting behavior. The main idea of the algorithm is to solve the target problem by imitating the whale's predatory behavior [2]. Since its introduction, the WOA has been favored by many scholars, and it has been widely used in optimal allocation of water resources [3], optimal control [4], and feature selection [5]. But as a swarm intelligence optimization algorithm, like DE, PSO, ACO, and other algorithms, they all have the shortcomings of slow convergence and easy to fall into local optimum. Therefore, in practical applications, various improvements have been made to the standard algorithms, such as [6–10]. Therefore, for the WOA algorithm, in recent years, many scholars have made a lot of improvements in improving algorithm convergence speed and optimization accuracy. For example, Abdel-Basset et al. [11] used Lévy flight and logical chaos mapping to replace and determine the coefficient vector C and switching probability P in the WOA, proposed an

improved whale optimization algorithm (IWOA), and verified the effectiveness of the proposed algorithm through experiments. Long Wen et al. [12] coordinated the exploration, developed the algorithm by updating the formula of the convergence factor of the nonlinear change, proposed an IWOA, and used experiments to verify the effectiveness of the improved algorithm. He et al. [13] introduced the adaptive strategy into the whale position update formula and proposed an IWOA for solving function optimization problems. This method balanced the global exploration and local development capabilities of the algorithm and accelerated the convergence speed and optimization accuracy of the algorithm. The experiments proved its superiority. Wu et al. [14] initialized the population through the quasireverse learning method to improve the population diversity. At the same time, the linear convergence factor was modified to a nonlinear convergence factor. In order to improve the solution of complex function optimization problems, Liu and He [15] proposed an IWOA based on adaptive parameters and niche technology, in which the adaptive probability threshold coordination algorithm was introduced for global search and local development

capabilities. Experiments show that the improved algorithm can effectively improve the ability to solve complex function optimization. WOA and its improved algorithms are all proposed for unconstrained optimization problems; there is no research on using this algorithm to solve constrained optimization problems. Therefore, this paper proposes an IWOA algorithm for solving constrained optimization problems.

The remainder of this paper has been organized as follows: The related research work has been described in Section 2, the introduction of the standard WOA and its improvement work has been described in Section 3, experimental results have been explained in Section 4, and the conclusion and future work have been explained in Section 5.

2. Related Research Work

Constrained optimization problems (Cops) are a type of nonlinear programming problems that often occur in the fields of daily life and engineering applications. There are usually two ways to solve this problem: deterministic algorithm and random algorithm [16]. Deterministic algorithms generally have high initial requirements, and they are generally unable to solve some problems that are not derivable, the feasible region is not connected, or there is no obvious mathematical expression. Even if some problems can be solved, the solutions obtained are mostly local optimal solutions [16]. The random algorithm is a swarm intelligence optimization algorithm that has emerged in recent years; it has obtained a lot of research in solving constrained optimization problems. Chen and Huo [17] proposed to use an improved GA to solve the Cops; this method used floating-point encoding; they also improved the genetic mutation operator and termination criterion. Long and Zhang [18] proposed an improved bat algorithm for solving Cops. This method used the good point set method to construct the initial population to maintain population diversity and also used inertial weights to improve the performance of the algorithm. An improved particle swarm optimization algorithm for solving Cops was proposed by Mi Yong and Gao [19]. This method used the penalty function method to treat constrained optimization problems as unconstrained optimization problems and used feasible basis rules to update individual and global extreme values. Lei et al. [20] proposed a new empire competition algorithm to solve the Cops and used the lexicographic method to simultaneously optimize the objective function of the problem and the degree of constraint violation. Long et al. [21] proposed the firefly algorithm to solve the constrained optimization problem. The algorithm used chaotic sequences to initialize the firefly position and introduced a dynamic random local search to speed up the convergence of the algorithm. Wang et al. [22] proposed an adaptive artificial bee colony algorithm to solve the Cops. The algorithm used an antilearning initialization method and an adaptive selection strategy. Mohamed et al. [23–25] proposed using an improved differential evolution algorithm to solve constrained optimization problems. This method is mainly to improve the mutation operator in standard differential

evolution, and the experimental results verify the effectiveness of the improved algorithm. There are many outstanding constraint-handling techniques in the literature. However, we do not review the literature of the WOA algorithm on constrained optimization problems.

Without loss of generality, the model of the Cops studied in this paper is as follows:

$$\begin{aligned} & \min f(x), \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, p, \\ \text{s.t. } & h_j(x) = 0, \quad j = p + 1, p + 2, \dots, m \\ & l_i \leq x_i \leq u_i, \quad i = 1, 2, \dots, d, \end{aligned} \quad (1)$$

where $f(x)$ is the objective function, $g_j(x)$ is the inequality constraint, $h_j(x)$ is the equality constraint, and l_i and u_i are the upper and lower bounds of the variable x_i , respectively.

3. The Introduction of the Standard WOA and Its Improvement

3.1. Standard WOA. WOA is a swarm intelligence optimization algorithm that simulates whale predation behavior. The algorithm simulates the unique bubble-net foraging method of whales [26] as shown in Figure 1.

The principle of the whale's bubble-net foraging is as follows: after the whale finds its prey, it creates a bubble net along the spiral path and moves upstream to prey. This predation behavior is divided into three stages: surrounding prey, bubble-net attack, and hunting prey.

3.1.1. Surrounding Prey Stage. In the WOA algorithm, the whale first recognizes the location of the prey and then surrounds it, but in fact, the whale cannot know the location of the prey in advance. Therefore, assuming that the current optimal position is the target prey, the other individuals in the group all move to the optimal position. The enclosing stage can be expressed by the following mathematical model:

$$\begin{aligned} X(t+1) &= X^*(t) - A \cdot D, \\ D &= |C \cdot X^*(t) - X(t)|, \end{aligned} \quad (2)$$

where t is the current number of iterations, $X^*(t)$ is the prey position vector (current optimal solution), $X(t)$ is the prey position vector, $A \cdot D$ is the surrounding step size, and

$$\begin{aligned} A &= 2a \cdot \text{rand} - a, \\ C &= 2 \cdot \text{rand}. \end{aligned} \quad (3)$$

In the above formula, rand is a random number between $[0, 1]$, a is a control parameter, and it decreases linearly from 2 to 0 with increasing of the iterations. The expression is as follows:

$$a = 2 - \frac{2t}{T_{\max}}, \quad (4)$$

where T_{\max} is the maximum number of iterations.

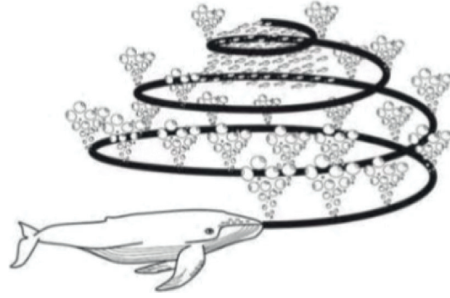


FIGURE 1: [26] Schematic diagram of whale bubble-net foraging.

3.1.2. *Bubble-Net Attack Stage.* The humpback whale's bubble-net foraging method is to move along the spiral path toward the prey in the constricted encirclement. Therefore, in WOA, two methods are designed to describe the predation behavior of whales: shrinking and surrounding mechanism and spiral update position, respectively.

Shrinking and surrounding mechanism: it is achieved by reducing the convergence factor a in equations (3) and (4).

Spiral update position: first calculate the distance between the individual whale and the current optimal position, and then simulate the whale to capture food in a spiral. The mathematical model can be expressed as follows:

$$\begin{aligned} X(t+1) &= D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), \\ D' &= |X^*(t) - X(t)|, \end{aligned} \quad (5)$$

where D' is the distance between the i th whale and the current optimal position, b is a constant coefficient used to define the logarithmic spiral form, and l takes the random number of $[-1, 1]$. In the predation process, the whale also needs to shrink the enclosure while spiraling to surround the prey. Therefore, in order to achieve this synchronous model, spiral envelopment and contraction envelopment are performed with the same probability.

3.1.3. *Hunting Prey Stage.* If $|A| \geq 1$, randomly select the whale to replace the current optimal solution; it can keep the whale away from the current reference target and enhance the algorithm's global exploration capabilities and also need to find a better prey to replace the current reference whale. The mathematical model is

$$\begin{aligned} X(t+1) &= X_{\text{rand}} - A \cdot D, \\ D &= |C \cdot X_{\text{rand}} - X(t)|, \end{aligned} \quad (6)$$

$$\begin{aligned} r_i &= e^i \quad (1 \leq i \leq M), \\ a_k^i &= \{(\{r_1 * i\}, \{r_2 * i\}, \dots, \{r_M * i\})\}, \quad (i = 1, 2, \dots, M, k = 1, 2, \dots, s), \end{aligned} \quad (7)$$

where $\{r_k * i\}$ means taking the decimal part of $r_k * i$.

In order to explain the difference between the initial population generated by the good point set and the general

where X_{rand} means randomly selecting the position vector of the whale.

Figure 2 shows the flowchart of the WOA.

3.2. *Improvement of WOA.* The WOA has the disadvantages of slow convergence and easy localization; this paper introduces three improved strategies and proposes an IWOA for solving constrained optimization problems.

3.2.1. *Generating Initial Population with Good Point Set Method.* For swarm intelligence optimization algorithms, the quality of the initial population directly affects the accuracy and speed of the algorithm [27]. The better the diversity of the initial population, the stronger the algorithm's global search ability. But for optimization problem, in the absence of any prior knowledge, it is necessary to use the information of the initial population. It is hoped that the diversity of the initial population will be used to fully reflect the basic information of the individual. The good point set method is an effective method that can reduce the number of trials. Experiments show that when the same number of points is taken, the good point set sequence is more uniformly distributed than the point sequence selected by the general random method, and it is closer to the integrate curve [28]. And the accuracy of the good point set method has nothing to do with the dimension; it can overcome the shortcomings of the random method. Take a good point set of M points in s -dimensional space:

random method, take the two dimensions as an example, where the value range of the variable is $[-80, 80]$ and the population size is 100. Figures 3 and 4 show the population

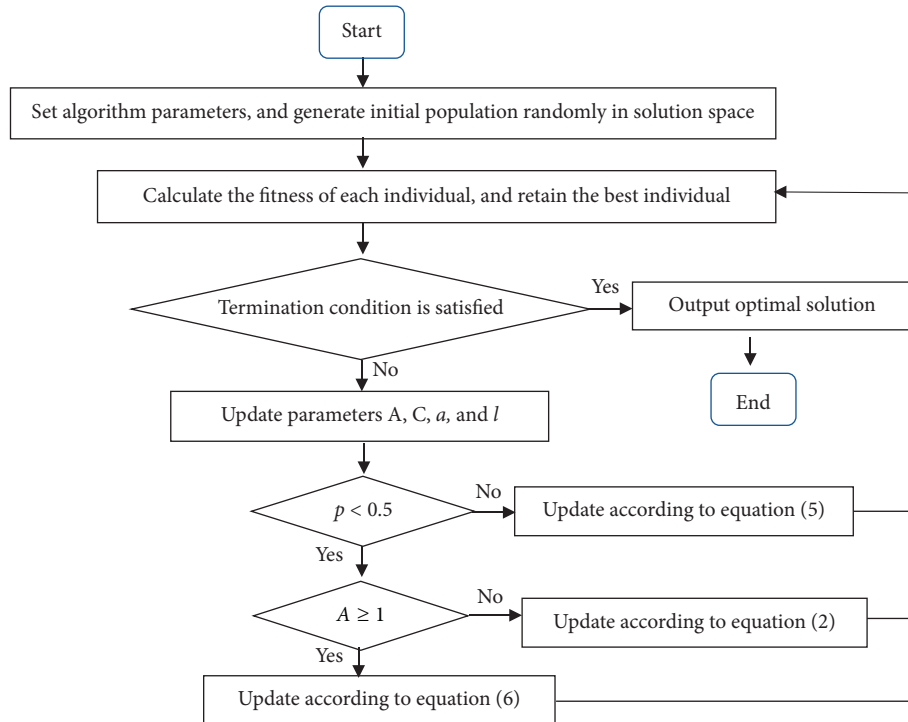


FIGURE 2: Flowchart of standard WOA.

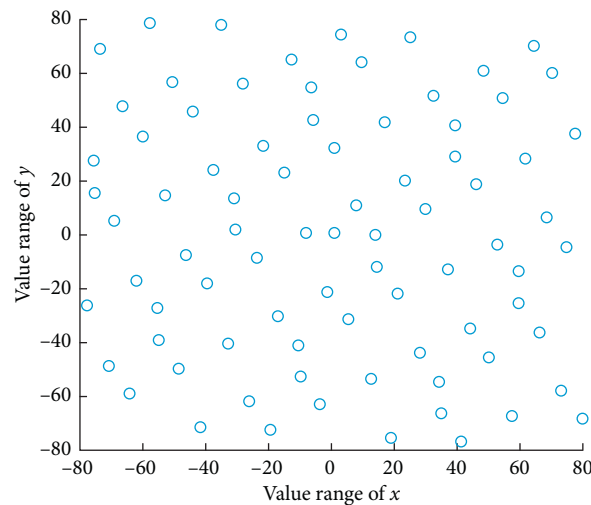


FIGURE 3: Initial solution generated by good point set.

distribution map generated by the random method and the good point set method, respectively.

As you can see from the abovementioned figures, the distribution of the initial population of individuals generated by the good point set method is more uniform and the diversity is better than that generated by the random method.

3.2.2. Nonlinear Convergence Factor. In the standard WOA, the global search ability and local development ability of the algorithm mainly depend on the parameter A , and the value

of A mainly depends on the convergence factor a ; therefore, the convergence factor is critical to the algorithm's ability to seek optimization. Literature [1] pointed out that a larger convergence factor a has better global search ability, to avoid the algorithm falling into local optimum; the smaller convergence factor a has stronger local search ability, to speed up the convergence of the algorithm. But in the standard WOA algorithm, the convergence factor a decreases linearly from 2 to 0 with the increasing of iteration. However, the linear reduction strategy of convergence factor a makes the algorithm have better global search ability in the early stage but the convergence speed is slow, and the convergence

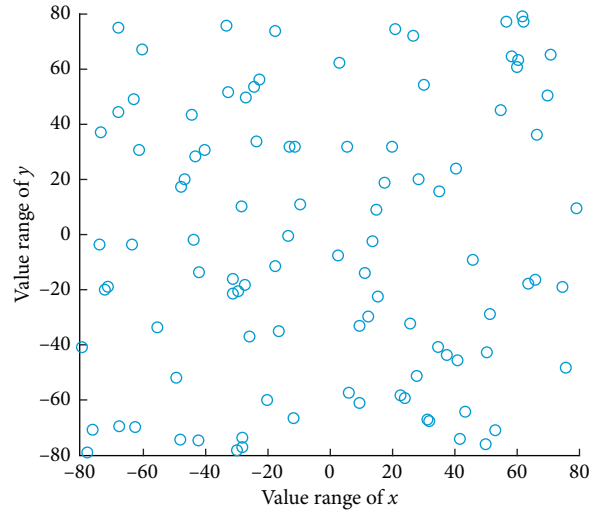


FIGURE 4: Initial solution generated by random.

speed in the later stage is fast, but it is easy to fall into the local optimal, and it is especially obvious when solving multipeak function problems. Therefore, in the evolutionary search process of WOA, the linear decreasing strategy of convergence factor a with the number of iterations cannot fully reflect the actual optimization search process [29]. Therefore, this paper proposes that, in the early stage of evolution, the convergence factor slowly decreases exponentially, to ensure a strong global search capability in the early stage of the search while maintaining a fast convergence speed; later in the search, the convergence factor decreases linearly, in order to ensure that the algorithm has a faster convergence speed and avoid the algorithm falling into the local optimal. The updated formula of the convergence factor is

$$\begin{aligned}
 a &= 2^{(1-t/T)}, \text{ if } t < 0.7 * T, \\
 a &= 2 - \frac{2t}{T}, \text{ if } t \geq 0.7 * T.
 \end{aligned}
 \tag{8}$$

Here, t is the current number of iterations and T is the maximum number of iterations.

3.2.3. Improvement of Mutation Operation. Similar to other swarm intelligence algorithms, in the late evolution of the WOA, the group gathers closer to the optimal solution, diversity decreases, and it is easy to fall into local optimum. Secondly, in the evolution update formula of the standard WOA, the evolution information of the current generation is less used. According to previous research on differential evolution algorithm [30], in the evolution process, making full use of current generation information can ensure the global search ability and local search ability of the algorithm; inspired by this, this paper also introduces the current generation information in the update formula of the WOA, set as follows:

Randomly generate probability p between $[0, 1]$; if $p < 0.5$ and $|A| \geq 1$, then update according to (9):

$$X(t+1) = \left(1 - \frac{t}{T}\right) \cdot X_{\text{rand}} - A \cdot D. \tag{9}$$

If $|A| < 1$, update according to (10):

$$X(t+1) = \frac{t}{T} X^*(t) - A \cdot D. \tag{10}$$

If the probability $p \geq 0.5$, update according to (11):

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + \frac{t}{T} X^*(t). \tag{11}$$

Here, $b = 1$, $l = (-2 - (t/T)) \cdot \text{rand} + 1$, and rand is a random number between $[0, 1]$; at the same time, in order to increase the diversity of the population and avoid premature convergence, Gaussian mutation is implemented on the individuals after the mutation, which takes the midpoint of the current individual and the optimal individual as the mean and the distance between the optimal individual and the current individual as the variance; the variation formula is as follows:

$$X(t+1) = N\left(0.5 * (X^*(t) + X(t)), |X^*(t) - X(t)|\right). \tag{12}$$

3.2.4. Constraint Processing Technology. In the constrained optimization problem, there are usually two methods for dealing with constraints. One is to transform into a multiobjective function, and the other is a penalty function method. This paper uses the penalty function method. The basic idea of the penalty function method is to take the violation of constraints as a kind of punishment for the minimum value and incorporate constraints into the objective function to obtain an unconstrained optimization problem and then use the optimization algorithm to optimize the unconstrained optimization problem and make the algorithm find the optimal solution of the problem under the action of the penalty [31]. Therefore, the objective function constructed in question (1) in this paper is as follows:

$$F(x) = f(x) + \delta(t)Q(x), \quad (13)$$

where $f(x)$ is the original objective function, $\delta(t)Q(x)$ is the penalty term, $\delta(t)$ is the penalty force, and $Q(x)$ is the penalty factor. In the above formula, if $\delta(t)$ is fixed, it is called the fixed penalty function method; otherwise, it is called the nonfixed penalty function method. In the nonfixed penalty function method, $\delta(t)$ changes with the number of iterations. This paper uses the nonfixed multisegment mapping penalty function method proposed by Parsopoulos and Vrahatis [32]; the expression is as follows:

$$Q(x) = \sum_{i=1}^m \theta(r_i(x))r_i(x)^\alpha. \quad (14)$$

$$\text{And } r_i(x) = \begin{cases} \max\{0, g_j(x)\}, & j = 1, 2, \dots, p, \\ \max\{|h_i(x)| - \varepsilon, 0\}, & i = p + 1, \dots, m \end{cases}$$

$$\theta(r_i(x)) = \begin{cases} 10, & r_i(x) < 0.01, \\ 50, & 0.01 \leq r_i(x) < 0.1, \\ 100, & 0.1 \leq r_i(x) < 1, \\ 300, & r_i(x) \geq 1, \end{cases} \quad \alpha = \begin{cases} 1, & r_i(x) < 1, \\ 2, & r_i(x) \geq 1, \end{cases} \quad \delta(t) = t\sqrt{t}. \quad (15)$$

3.3. Out-of-Bound Processing Method. During the evolution of swarm intelligence algorithm, it is a common problem that the new solutions generated exceed the prescribed boundaries. How to deal with individual solutions beyond boundaries is also critical. There are two common processing methods: one is to take the boundary value when beyond the boundary, and the other is to regenerate a new solution within the value range. This paper takes the first method.

The specific steps of the IWOA are as follows:

- (i) *Step 1.* Set the population size M and the maximum number of iterations T and generate the initial population with the point set method.
- (ii) *Step 2.* Calculate the fitness of the population and keep the current optimal solution.
- (iii) *Step 3.* Update parameters A, a, C, l .
- (iv) *Step 4.* Generate a random number p between $[0, 1]$; if $p < 0.5$ and $A \geq 1$, mutate according to equation (9); otherwise, update according to equation (10), if $p \geq 5$, and then according to (11).
- (v) *Step 5.* Perform Gaussian mutation to the individuals in step 4 according to formula (12).
- (vi) *Step 6.* Perform boundary treatment on the individual after step 5 mutation and calculate the fitness of the new individuals, keeping the current optimal solution.
- (vii) *Step 7.* Determine whether the termination condition is satisfied. If yes, output the current optimal solution; otherwise, $t = t + 1$ and return to step 2.

4. Simulation Experiment

In order to test the performance of the IWOA, firstly select 8 benchmark functions [33] for testing as shown in Table 1. The selected functions include single-modal functions and complex nonlinear multimodal functions. The single-modal function tests the local exploitation ability of the algorithm, and the multimodal function tests the global exploration ability of the algorithm. The dimension of all tested functions is set to 30. And compared with the standard WOA, Table 2 shows the mean values and the standard deviations of the two algorithms running 30 times.

It can be seen from Table 2 that, for multimodal functions F_6 and F_7 , both algorithms can find the optimal solution. But F_8 is better than the standard WOA. For single-mode functions F_1, F_2, F_3, F_4 , and F_5 , the mean and std. are significantly better than the standard WOA, especially F_3 ; the advantage of the IWOA is particularly obvious. This shows that the IWOA algorithm is superior to the standard WOA in both global search and local search capabilities.

Figure 5 shows the function evolution curves.

It can also be seen from the evolution curve that the IWOA algorithm converges fast; the tested functions can converge to the optimal solution or close to the optimal solution.

To further evaluate the performance of the improved algorithm to solve constrained optimization problems, in this paper, 13 benchmark constrained optimization problems in [34] are selected for testing. All functions are transformed into unconstrained optimization problems according to (13). In fact, it is similar to function optimization after conversion. The maximum evolutionary generation is 800. When the variable dimension $n < 10$, the population size is set to 80; $10 \leq n$, the population size is set to 100. Other parameter settings are the same as mentioned earlier. The WOA and IWOA algorithms are coded and realized in MATLAB. For each problem, 20 independent runs are performed and statistical results are provided including the maximum number of function evaluations (FEs), the best, mean values, and standard deviations. Table 3 shows the calculation results and comparison with other algorithms.

It can be seen from Table 3 that the IWOA algorithm can find the optimal solution or very close to the optimal solution. Among them, g01, g04, g05, g06, g08, g9, g10, g11, g12, and g13 have the best results, and all find the theoretical optimal solutions. And the solutions of g02, g03, and g07 are very close to the global optimal solutions. It can also be seen from the standard deviations that the stability of the IWOA is improved and has better robustness.

Further analysis found that, for g01, g04, g06, g08, g09, g10, and g12 with inequality constraints, g11 and g13 with equality constraints, and g05 with both equality and inequality constraints, the optimal solutions found by the IWOA algorithm are equivalent to ICA [20]. For g01, g08, g09, and g011, IWOA and ICA have almost the same stability, but for g02, g07, and

TABLE 1: [33] Main characteristics of 8 benchmark functions.

Function	Expression	Range	Theoretical optimal solution
Sphere	$F_1 = \sum_{i=0}^n x_i^2$	[-100, 100]	0
Schwefel2.21	$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10, 10]	0
Schwefel1.2	$F_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	[-100, 100]	0
Rosenbrock	$F_4 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]	0
Sumsquare	$F_5 = \sum_{i=1}^n ix_i^2$	[-10, 10]	0
Rastrigin	$F_6 = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12, 5.12]	0
Griewank	$F_7 = (1/4000)\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	[-600, 600]	0
Alpine	$F_8 = \sum_{i=1}^n x_i \cdot \sin(x_i) + 0.1 \cdot x_i $	[-10, 10]	0

TABLE 2: Performance comparison with standard WOA.

Functions	Evaluation index	WOA	IWOA
F ₁	Mean	1.23E - 101	1.29E - 137
	Std.	2.5117E - 101	1.8965E - 137
F ₂	Mean	2.89E - 60	1.34E - 77
	Std.	6.3447E - 60	2.24328E - 77
F ₃	Mean	1.94E + 04	9.27E - 100
	Std.	6.894E + 03	1.2311E - 99
F ₄	Mean	26.05163423	25.47650265
	Std.	0.448316347	0.265685577
F ₅	Mean	4.37E - 99	5.30E - 140
	Std.	1.07959E - 98	5.7842E - 140
F ₆	Mean	0	0
	Std.	0	0
F ₇	Mean	0	0
	Std.	0	0
F ₈	Mean	3.22E - 61	6.15E - 78
	Std.	5.09486E - 61	6.91291E - 78

g10, IWOA is more stable than ICA. The COMDE [23] algorithm shows good stability, although the standard deviations of the IWOA are less than that of COMDE, but for g01, g02, g03, g05, g07, g09, g10, and g13, their FEs are less than that of COMDE, indicating that the calculation cost of IWOA is less than that of COMDE. The NDE [24] is able to find the global optimal solution consistently in 12 out of 13 comparison functions with the exception of g02, but their FEs are set to 240,000. Compared with IPES [35], the IWOA algorithm finds more optimal solutions, and there are better mean values on g01, g02, g05, g10, and g13. For g03, g08, g11, and g12, the mean values of the two are equivalent. For g01 and g13, IWOA and IFOA [36] have almost the same mean values and standard deviations. On the whole, IFOA has strong stability, but its ability to find the best solutions is poor.

Figures 6–8 show the convergence graphs of $\log_{10}(f(x) - f(x^*))$ over FEs with 20 runs, where $f(x^*)$ is the known optimal solution. Following the previous rule, the dimension of g01, g02, g03, and g07 is greater than 10, so the FEs are set to 80,000, and the others are set to 64,000.

As shown in Table 3, IWOA is able to find the global optimal solution consistently in 10 out of 13 test functions over 20 runs with the exception of test functions (g02, g03, and g07).

In order to test the effect of various parameters on IWOA, we compared the methods before and after the

improvement. Among them, the initial population produced without a good point set method is called IWOA-1, the method without nonlinear convergence factor is called IWOA-2, the method without improved mutation operation is called IWOA-3, the method without constraint processing technology is called IWOA-4, and the method without out-of-bound processing is called IWOA-5. Each algorithm runs 20 times under the same conditions. Table 4 shows the best, mean values, and standard deviations.

It can be seen from Table 4 that the IWOA-2 without nonlinear convergence factor is the worst. It shows that, in the IOWA algorithm, the linear convergence factor has an obvious influence on it. For g01, g04, g05, g07, and g10 with many constraints, IWOA-5 without out-of-bound processing is not ideal, indicating that the more the constraints, the more sensitive the boundary. For IWOA-4, it can be seen that the experimental results are not as good as those with constraint processing technology, indicating that the constraint processing technology adopted in this paper is effective. For g01, g02, g03, g04, g07, g09, g10, and g13 with more variables, the effect of using the good point set method to generate the initial solutions is obviously better than the IWOA-1 without the good point set method. It shows that increasing the population diversity can improve the optimization ability. At the same time, it can be seen that the mutation operation that adds evolution information is better

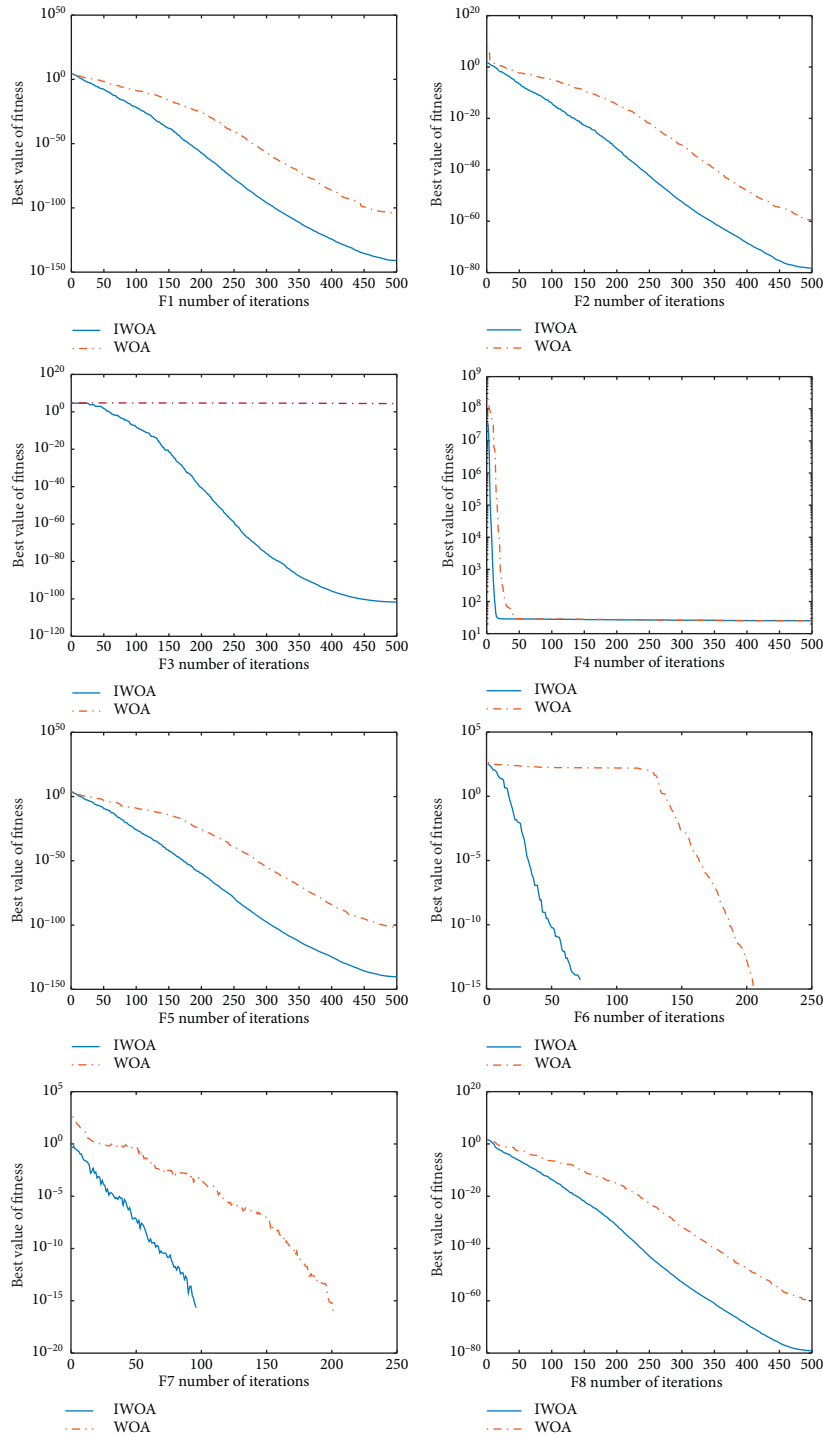


FIGURE 5: The evolution curve of finding the optimal solution of equation F1–F8.

TABLE 3: Comparison of IWOA with other algorithms.

Function/optimal	Methods	Best	Mean	Std.	FEs
g01/−15.000	IPES	−14.999	−14.999	—	240000
	IFOA	−15	−15	0	164111
	COMDE	−15	−15	$7.4E-01$	130000
	ICA	−15	−15	0	200000
	IWOA	−15	−14.998	$1.97E-13$	80000

TABLE 3: Continued.

Function/optimal	Methods	Best	Mean	Std.	FEs
g02/-0.803619	IPES	-0.803607	-0.792771	—	240000
	IFOA	-0.803603	-0.803617	2.4E-06	553916
	COMDE	-0.803619	-0.801238	5.0E-03	200000
	ICA	-0.803619	-0.791906	5.1E-02	200000
	IWOA	-0.803576	-0.801395	6.9E-03	80000
g03/-1.000	IPES	-1	-1	—	240000
	IFOA	-1.00052	-1.00052	9.9E-16	434916
	COMDE	-1.0000	-1.0000	3.03E-08	150000
	ICA	-1.0005	-1.0004	6.2E-05	200000
	IWOA	-1.0001	-0.9973	3.9E-03	80000
g04/-30665.539	IPES	-30665.539	-30665.539	—	240000
	IFOA	-30665.538	-30665.538	2.2E-11	134756
	COMDE	-30665.539	-30665.539	0	50000
	ICA	-30665.539	-30665.539	0	200000
	IWOA	-30665.539	-30664.771	1.7E-01	64000
g05/5126.498	IPES	5126.498	5139.003	—	240000
	IFOA	5126.495	5126.495	2.5E-08	786696
	COMDE	5126.498	5126.498	0	200000
	ICA	5126.498	5126.498	3.8E-01	200000
	IWOA	5126.498	5126.804	3.5E+00	64000
g06/-6961.814	IPES	-6961.814	-6961.814	—	240000
	IFOA	-6961.814	-6961.814	1.8E-12	129950
	COMDE	-6961.814	-6961.814	0	12000
	ICA	-6961.814	-6961.814	0	200000
	IWOA	-6961.814	-6952.593	9.9E+00	64000
g07/24.306	IPES	24.307	24.316	—	240000
	IFOA	24.3068	24.3064	1.29E-04	695875
	COMDE	24.3062	24.3062	4.7E-07	200000
	ICA	24.306	24.3704	3.2E-01	200000
	IWOA	24.3056	24.3705	2.4E-02	80000
g08/-0.095825	IPES	-0.095825	-0.095825	—	240000
	IFOA	-0.095825	-0.095825	8.33E-17	10331
	COMDE	-0.095825	-0.095825	9.0E-18	4000
	ICA	-0.095825	-0.095825	0	200000
	IWOA	-0.095825	-0.095823	4.8E-06	64000
g09/680.630	IPES	680.630	680.630	—	240000
	IFOA	680.632	680.647	2.02E-08	559717
	COMDE	680.632	680.630	4.07E-13	70000
	ICA	680.630	679.429	4.5E-02	200000
	IWOA	680.630	680.616	9.1E-02	64000
g10/7049.331	IPES	7051.341	7210.360	—	240000
	IFOA	7049.2844	7049.2607	6.4E-03	883825
	COMDE	7049.248	7049.248	1.5E-04	200000
	ICA	7049.248	7053.721	1.7E+02	200000
	IWOA	7049.331	7046.627	3.1E+01	64000
g11/0.7499	IPES	0.75	0.75	—	240000
	IFOA	0.75	0.75	2.7E-11	372647
	COMDE	0.7499	0.7499	0	50000
	ICA	0.74990	0.74995	0	200000
	IWOA	0.7499	0.7495	2.9E-06	64000
g12/-1.000	IPES	-1.000	-1.000	—	240000
	IFOA	-1.000	-1.000	0	4505
	COMDE	-1	-1	0	6000
	ICA	-1	-1	0	200000
	IWOA	-1.000	-0.998	2.3E-03	64000

TABLE 3: Continued.

Function/optimal	Methods	Best	Mean	Std.	FEs
g13/0.05395	IPES	0.053950	0.0514626	—	240000
	IFOA	0.0543003	0.0539588	$6.98E-05$	904975
	COMDE	0.0539415	0.0539415	$1.4E-17$	150000
	ICA	—	—	—	—
	IWOA	0.05395	0.05394	$3.4E-05$	64000

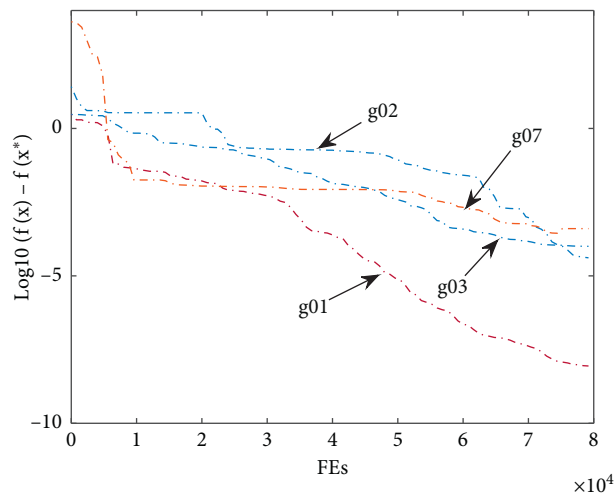


FIGURE 6: Convergence graph for g01–g03 and g07.

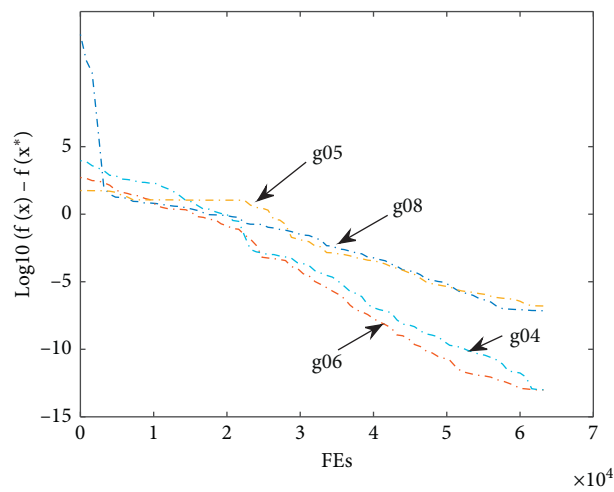


FIGURE 7: Convergence graph for g04–g06 and g08.

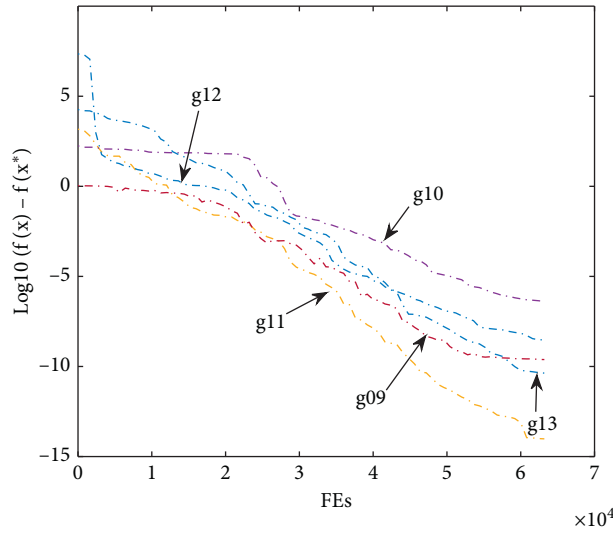


FIGURE 8: Convergence graph for g09–g13.

TABLE 4: Statistical features of the results obtained by various versions of IWOA.

Problem	Features	IWOA	IWOA-1	IWOA-2	IWOA-3	IWOA-4	IWOA-5
g01	Best	-15	-14.998	-14.999	-15	-14.998	-14.999
	Mean	-14.998	-14.998	-14.991	-14.997	-14.994	-14.998
	Std.	1.97E-13	3.24E-04	9.85E-03	7.46E-04	5.27E-03	2.13E-04
g02	Best	-0.803572	-0.8038	-0.7997	-0.8034	-0.8017	-0.8038
	Mean	-0.801395	-0.8006	-0.8076	-0.8012	-0.8007	-0.8023
	Std.	6.9E-03	7.05E-03	1.65E-02	6.67E-03	7.69E-02	6.5E-03
g03	Best	-1.0001	-0.9997	-0.9958	-1.0001	-0.9979	-0.9999
	Mean	-0.9973	-0.9956	-0.9842	-1.0034	-0.9528	-0.9921
	Std.	3.9E-03	2.62E-03	4.8E-03	3.6E-03	5.3E-03	5.3E-03
g04	Best	-30665.539	-30664.66	-30664.41	-30665.88	-30664.32	-30665.51
	Mean	-30664.771	-30663.86	-30664.02	-30664.85	-30661.36	-30664.72
	Std.	1.7E-01	3.5E-01	2.1E-02	2.4E-01	8.6E-01	2.6E-01
g05	Best	5126.498	5126.697	5124.071	5126.421	5125.535	5125.386
	Mean	5126.804	5126.022	5129.948	5126.977	5127.811	5125.846
	Std.	3.5E+00	4.5E+00	6.2E+00	5.4E+00	7.4E+00	1.2E+00
g06	Best	-6961.814	-6961.814	-6958.081	-6961.814	-6959.795	-6961.814
	Mean	-6952.593	-6959.795	-6950.981	-6951.057	-6953.327	-6950.333
	Std.	9.9E+00	6.0E+00	7.3E+00	10E+00	4.6E+00	11E+00
g07	Best	24.3056	24.3037	24.3421	24.3184	24.3228	24.3106
	Mean	24.3705	24.3674	24.4065	24.3436	24.3788	24.3428
	Std.	2.4E-02	1.3E-01	5.2E+00	2.7E-01	1.3E+00	2.2E-02
g08	Best	-0.095825	-0.095825	-0.095802	-0.095824	-0.095822	-0.095825
	Mean	-0.095823	-0.095820	-0.095822	-0.095821	-0.095820	-0.095821
	Std.	4.8E-06	2.7E-05	3.2E-04	6.5E-06	5.2E-05	6.7E-06
g09	Best	680.630	680.638	682.241	680.630	680.633	680.630
	Mean	680.616	680.882	687.076	681.300	683.533	680.605
	Std.	9.1E-02	0.499	3.942	0.291	2.438	8.6E-01
g010	Best	7049.331	7047.405	7058.088	7050.038	7056.534	7049.148
	Mean	7046.627	7043.326	7051.038	7041.397	7042.572	7047.406
	Std.	3.1E+01	5.2E+02	6.3E+02	3.7E+01	2.7E+02	1.8E+01
g011	Best	0.7499	0.7499	0.7501	0.7499	0.7499	0.7499
	Mean	0.7495	0.7487	0.7416	0.7499	0.7494	0.7491
	Std.	2.9E-06	3.5E-06	4.3E-05	3.1E-06	2.5E-06	7.4E-06
g012	Best	-1.000	-1.000	-0.997	-1.000	-1.000	-0.999
	Mean	-0.998	-0.996	-0.991	-0.997	-0.993	-0.995
	Std.	2.3E-03	5.3E-03	3.5E-02	2.6E-03	9.2E-03	6.3E-03

TABLE 4: Continued.

Problem	Features	IWOA	IWOA-1	IWOA-2	IWOA-3	IWOA-4	IWOA-5
g013	Best	0.05395	0.05396	0.05398	0.05396	0.05393	0.05395
	Mean	0.05394	0.05393	0.05388	0.05392	0.05390	0.05391
	Std.	$3.4E-05$	$2.5E-04$	$1.7E-03$	$4.2E-05$	$4.8E-04$	$3.7E-05$

than IWOA-3. On the whole, the most obvious impact on the algorithm is the processing of nonlinear convergence factor and the processing of constraint technology.

In summary, the test results of 13 benchmark constrained optimization problems show that the IWOA algorithm has high accuracy and strong stability, and it is competitive in constrained optimization problems.

5. Conclusion and Future Work

Like other swarm intelligence algorithms, the WOA algorithm also has the disadvantages of slow convergence and easy to fall into local optimum. According to the characteristics of the function optimization problem, this paper improves the standard WOA from three aspects: (i) initial population, (ii) convergence factor, and (iii) mutation operation. An IWOA is proposed. At the same time, in order to solve the constraint optimization problem, the dynamic penalty function method is introduced to transform the constrained optimization problem into an unconstrained optimization problem. The test of 8 benchmark functions shows that the IWOA algorithm has improved global exploration ability and local exploitation ability. Then 13 benchmark constrained optimization problems were tested, and compared with other algorithms in many aspects, the experimental results can show that the IWOA algorithm is competitive.

WOA is a new type of swarm intelligence algorithm; the next work is to further improve the algorithm, so that it can be applied in more fields and solve more constrained optimization problems.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Research Project of Guangxi Universities (KY2015YB521), the Youth Education Teachers' Basic Research Ability Enhancement Project of Guangxi Universities (2019KY1098), and the Key Project in Teaching Reform of Lushan College of Guangxi University Science and Technology (2018JGZ004).

References

- [1] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [2] M. H. Zhong and W. Long, "Whale optimization algorithm based on stochastic adjustment control parameter," *Science Technology and Engineering*, vol. 17, no. 12, pp. 68–73, 2017.
- [3] J. X. Sha, "Application of ameliorative whale optimization algorithm to optimal allocation of multi-objective water resources," *Water Resources and Hydropower Engineering*, vol. 49, no. 4, pp. 18–26, 2018.
- [4] H. H. Mehne and S. Mirjalili, "A parallel numerical method for solving optimal control problems based on whale optimization algorithm," *Knowledge-Based Systems*, vol. 151, no. 7, pp. 114–123, 2018.
- [5] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, no. 1, pp. 441–453, 2018.
- [6] D. Wu, Y. J. Xu, H. M. Zhao et al., "A novel gate resource allocation method using improved PSO-Based QEA," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–9, 2020.
- [7] D. Wu, Y. J. Xu, X. Z. H. Gao et al., "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, pp. 1–10, 2020.
- [8] Y. Song, D. Wu, W. Deng et al., "MPPCEDE: multi-population parallel co-evolutionary differential evolution for parameter optimization," *Energy Conversion and Management*, vol. 228, Article ID 113661, 2021.
- [9] D. Wu, Y. J. Xu, Y. J. Song et al., "An effective improved co-evolution ant colony optimization algorithm with multi-strategies and its application," *International Journal of Bio-Inspired Computation*, vol. 16, no. 3, pp. 158–170, 2020.
- [10] Y. J. Song, D. Q. Wu, W. Ali, W. Mohamed, X. Zhou, and B. Zhang, "Enhanced success history adaptive DE for parameter optimization of photovoltaic models," *Complexity*, vol. 2021, Article ID 6660115, 22 pages, 2020.
- [11] M. Abdel-Basset, L. Abdle-Fatah, and A. K. Sangaiah, "An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment," *Cluster Computing*, vol. 22, no. 4, pp. 8319–8334, 2018.
- [12] W. Long, S. H. Cai, J. J. Jiao et al., "Improved whale optimization algorithm for large scale optimization problems," *Systems Engineering Theory & Practice*, vol. 37, no. 11, pp. 2983–2994, 2017.
- [13] Q. He, K. Y. Wei, and Q. S. H. Xu, "Mixed strategy based improved whale optimization algorithm," *Application Research of Computers*, vol. 36, no. 12, pp. 3647–3665, 2019.
- [14] Wu Z. Q., Mu Y. M. Improved Whale Optimization Algorithm. Application Research of Computers, 2020-03-27: <http://kns.cnki.net/kcms/detail/51.1196.tp.20200326.1103.023.html>.

- [15] L. Liu and Q. He, "Improved whale optimization algorithm for solving function optimization problems," *Application Research of Computers*, vol. 37, no. 4, pp. 1005–1009, 2020.
- [16] G. Y. Ning, D. Q. Cao, and Y. Q. Zhou, "An improved differential evolution algorithm for solving constraint optimization problem," *Mathematics in Practice and Theory*, vol. 47, no. 2, pp. 156–165, 2017.
- [17] L. Chen and Y. L. Huo, "The solution to constrained optimization problem by improved genetic algorithm," *Journal of Chongqing Technology and Business (Natural Sciences Edition)*, vol. 31, no. 9, pp. 64–67, 2014.
- [18] W. Long and W. Z. H. Zhang, "Modified bat algorithm for solving constrained optimization problems," *Application Research of Computers*, vol. 31, no. 8, pp. 2350–2353, 2014.
- [19] Y. Q. Mi and Y. L. Gao, "The improved particle swarm optimization algorithm for solving constrained optimization problems," *Journal of Jiangxi Normal University (Natural Science)*, vol. 39, no. 1, pp. 59–63, 2015.
- [20] D. M. Lei, S. A. N. Q. Cao, and M. Li, "An imperialist competitive algorithm for solving constrained optimization problem," *Control and Decision*, vol. 34, no. 8, pp. 1663–1670, 2019.
- [21] W. Long, S. H. H. Cai, J. J. Jiao et al., "Firefly algorithm for solving constrained optimization problems and engineering applications," *Journal of Central South University (Science and Technology)*, vol. 46, no. 4, pp. 1260–1266, 2015.
- [22] Z. H. Wang and X. F. Li, "Self-adaptive artificial bee colony Algorithm for constrained optimization problem," *Computer Engineering and Applications*, vol. 55, no. 15, pp. 47–57, 2019.
- [23] A. W. Mohamed and H. Z. Sabry, "Constrained optimization based on modified differential evolution algorithm," *Information Sciences*, vol. 194, no. 7, pp. 171–208, 2012.
- [24] A. W. Mohamed, "A novel differential evolution algorithm for solving constrained engineering optimization problems," *Journal of Intelligent Manufacturing*, vol. 29, no. 3, pp. 659–692, 2018.
- [25] A. W. Mohamed, A. K. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems," *International Journal of Applied Metaheuristic Computing*, vol. 10, no. 1, pp. 1–28, 2019.
- [26] W. A. Watkins and W. E. Schevill, "Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis*, *Balaenoptera borealis*, *Megaptera novaeangliae*, and *Balaenoptera physalus*," *Journal of Mammalogy*, vol. 60, no. 1, pp. 155–163, 1979.
- [27] R. Haupt and S. Haupt, *Practical Genetic algorithm*, John Wiley & Sons, New York, NY, USA, 2004.
- [28] C. H. K. Zhang, L. Y. Xu, and H. H. Shao, "Improved chaos optimization algorithm and its application in nonlinear constraint optimization problems," *Journal of Shang Hai Jiao Tong University*, vol. 34, no. 5, pp. 593–595, 2000.
- [29] Z. L. Wei, H. Zhao, M. D. Li et al., "A grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter," *Journal of Air Force Engineering University (Natural Science Edition)*, vol. 17, no. 3, pp. 68–72, 2016.
- [30] G. Y. Ning, D. Q. Cao, and Y. Q. Zhou, "Improved differential evolution algorithm for solving 0-1 programming problems," *Journal of Systems Science and Complexity*, vol. 39, no. 1, pp. 120–132, 2019.
- [31] L. Wu, Y. Wang, S. Zhou et al., "Differential evolution for nonlinear constrained optimization using non-stationary multi-stage assignment penalty function," *Systems Engineering Theory and Practice*, vol. 27, no. 3, pp. 128–133, 2007.
- [32] Parsopoulos K. E., Vrahatis M. N. Particle swarm optimization method for constrained optimization problems [EB/OL]. [2015-01-03], http://www.researchgate.net/publication/2527227_Particle_swarm_optimization_method_for_constrained_optimization_problems.
- [33] J. S. Yong, L. W. Xi, Y. Chen et al., "A modified whale optimization algorithm for large-scale global optimization problems," *Expert Systems with Applications*, vol. 114, pp. 563–577, 2018.
- [34] J. J. Liang, T. P. Runarsson, E. Mezura-Montes et al., "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Technical Report*, vol. 18, pp. 1–25, 2006.
- [35] C. H. G. Cui and X. F. Yang, "Interior penalty rule based evolutionary algorithm for constrained optimization," *Journal of Software*, vol. 26, no. 7, pp. 1688–1699, 2015.
- [36] Shi J. P., Li P. S., Liu G. P., et al. Improved Fruit Fly Optimization Algorithm for Solving Constrained Optimization Problems and Engineering Applications. *Control and Decision* [2019-10-23], <http://kns.cnki.net/kcms/detail/21.1124.TP.20191022.1622.013.html>.