Journal of Computers Vol. 33 No. 2, April 2022, pp. 51-67
doi: 10.53106/199115992022043302005

A Guided Mutation Grey Wolf Optimizer Algorithm Integrating Flower Pollination Mechanism

Jing-Sen Liu[1,2], Qing-Qing Liu[2], Fang Zuo[1,2*]

[1] Henan International Joint Laboratory of Theories and Key Technologies on Intelligence Networks, Henan University, Kaifeng, 475004, China
{ljs, zuofang}@henu.edu.cn
[2] College of Software, Henan University, Kaifeng, 475004, China
lqqhenu@163.com

Received 1 June 2021; Revised 7 September 2021; Accepted 7 October 2021

Abstract. The basic Grey Wolf Optimizer (GWO) has some shortcomings, for example, the convergence speed is slow, it is easy to fall into local extremum, and high-dimensional optimization ability is poor and so on. In response to these shortcomings, an improved grey wolf algorithm which combines flower pollination mechanism, teaching mechanism and polynomial variation is proposed in this study. The flower pollination mechanism is integrated with GWO algorithm, Levy distribution is introduced into the global search of grey wolf population. And the double random mechanism is added in the local search, for these improvements, this algorithm's overall optimization performance is improved. The teaching mechanism is added to α wolf to improve the algorithm's convergence speed. Polynomial mutation is applied to the individuals with poor optimization effect to improve the algorithm's accuracy and its ability to jump out of local extremum. Theoretical analysis shows that the time complexity of the improved algorithm is the same as that of the basic algorithm. The test results of five representative comparison algorithms on multiple different characteristics and different dimensions of CEC2017 benchmark functions and two classical engineering problems show that FMGWO algorithm has high optimization accuracy, convergence speed and solution stability. Therefore, it has obvious advantages in global optimization.

Keywords: grey wolf optimization algorithm, flower pollination algorithm, teaching mechanism, polynomial mutation, CEC2017

1 Introduction

With the development of society, the computing scope and complexity of all kinds of application problems are increasing day by day. Therefore, it is urgent to explore more efficient optimization techniques to solve these complex problems effectively. In recent years, the heuristic intelligent optimization algorithm based on bionics was favored by many scholars because of its simple operation and efficient solution. For instance, the foraging behavior of birds in nature inspires the Particle Swarm Optimization (PSO) [1] algorithm. Cuckoo Search Algorithm (CSA) [2] algorithm is urged by brooding behavior of cuckoo parasitism. Flower Pollination Algorithm (FPA) [3] is enlightened by natural flower pollination course. The special predation behavior of whales inspires Whale Optimization Algorithm (WOA) [4]. Salp Swarm Algorithm (SSA) [5] comes from the foraging behavior of colefish in the sea. These heuristic algorithms offer assistance for solving complex optimization problems. Moreover, these algorithms have extensive use in many application fields.

Grey Wolf Optimizer (GWO) is a new swarm intelligence optimization algorithm proposed by S. Mirjalili et al in 2014. The hierarchical and predatory behavior of the grey wolf in nature is simulated in this algorithm. The behavior of hunting prey and attacking prey of wolves correspond to the global and local search of GWO algorithm respectively. Moreover, the process of hunting prey is corresponding to the algorithm's optimization process. The principle of GWO algorithm is simple, and it has less parameter to be adjusted and it is easy to implement. Currently, GWO has been applied in many fields. Such as solving economic dispatch [6], pipeline scheduling [7], power failure risk prevention [8], multi-sensor training [9], UAV path planning [10] and so on.

Although GWO algorithm has good performance and other advantages, it still has some shortcomings. For example, its convergence rate is slow in the later stage and will easily fall into the local extremum. Moreover, sometimes there is unstable optimization accuracy. For these reasons, many experts have improved the shortcomings of GWO algorithm. M.H. Qais et al. [11] improved the solution accuracy and the algorithm's execution efficiency by modifying control parameters and the position updating method. The improved algorithm had been successful-

* Corresponding Author

ly applied to the power system. S. Gupta et al. [12] proposed the GWO algorithm based on memory. According to the optimal location of each individual grey wolf, the location updating formula was adjusted by combining cross selection and greedy selection. M.H. Nadimi-Shahraki et al. [13] improved the Grey Wolf Optimizer (I-GWO) for solving global optimization and engineering design problems. The IGWO algorithm benefited from a new movement strategy named dimension learning-based hunting (DLH). This improvement was proposed to alleviate the lack of population diversity and premature convergence of the GWO algorithm. T.H. Jiang et al. [14] raised a two-stage individual position encoding / decoding mechanism. Moreover, they designed the individual position updating method with weight coefficient and a nonlinear adjustment strategy of convergence factor to improve the optimization accuracy and the algorithm's convergence rate. W. Long et al. [15] proposed a new parameter C strategy and opposition-learning strategy based on optical lens imaging principle to improve the grey Wolf optimization algorithm. Moreover，it could balance the exploration and mining capabilities of the algorithm and avoid the algorithm falling into local optimum. B.P. Sahoo et al. [16] updated the position by neglecting the unimportant δ wolf in the algorithm's hunting phase, which made the algorithm's structure simpler and improved the algorithm's efficiency. Improved algorithm was successfully applied to the frequency regulation in power system. W. Long et al. [17] introduced the nonlinear adjustment strategy of control parameters in GWO algorithm, which well-balanced the process of the algorithm's local and global search. Furthermore, they raised the optimal position to propose a new position update formula, which enhanced the ability of the algorithm to get rid of the local extremum. A. Rodríguez et al. [18] improved a modified version of the Grey Wolf Optimizer called Group-based Synchronous-Asynchronous Grey Wolf Optimizer. The proposed algorithm presented a better balance between exploration and exploitation. And there was an increment in the accuracy and the ability to avoid the convergence in local minima. C. Lu et al. [19] studied a new multi-objective cellular GWO algorithm and used it to handle the hybrid flow shop scheduling problem. G.W. Huang et al. [20] put forward the moving average adaptive grey wolf updating strategy and grey wolf genetic strategy, which improved the algorithm's global convergence ability. B. Zeng et al. [21] combined GWO with differential evolution (DE) algorithm. Then a fault diagnosis model by using improved algorithm was established, and the model was used to transformer fault diagnosis. G. Natesan et al. [22] proposed the mean value GWO algorithm, improved the hunting and siege formula of grey wolf, which could improve the accuracy of the algorithm. Finally, the algorithm was used to solve the task scheduling problem. A.A. Bilal et al. [23] proposed the distributed Grey Wolf optimizer (DGWO). DGWO used the largest order value (LOV) method to convert the continuous candidate solutions produced by DGWO to discrete candidate solutions. And it applied to Optimal scheduling of workflows in cloud computing environments. K.W. Liu et al. [24] adopted a hybrid individual update strategy integrating Lévy flight and multigroup reorganization strategy when the grey wolf individual position was updated. Consequently, it enhanced the search ability and algorithm's optimization performance. C. Lu et al. [25] controlled convergence factor of GWO by chaos mapping to improve the algorithm's optimization performance.

These improvements make the algorithm have better optimization performance in the corresponding field. However, the optimization accuracy, stability and ability to trip out local extremum still need to be improved. In this paper, dominant variation GWO algorithm which combined flower pollination mechanism with teaching mechanism was proposed. The innovations of this paper mainly include: 1) The mechanism of flower pollination algorithm is combined with GWO algorithm. The Levy distribution is introduced into the prey searching stage of grey wolf population, which could make the individual of grey wolf carry out global search more fully. More importantly, the double random mechanism is added into the algorithm's local search phase to improve the ability to jump out of local extremum. 2) The teaching mechanism is added into α wolf to enhance the leadership ability of the α wolf. It can speed up the grey wolf individual to approach the optimal value region and accelerate the algorithm's convergence rate. 3) Polynomial mutation is introduced in the process of optimization, and the position mutation is carried out for the individuals with poor optimization effect. 4) Theoretical analysis shows that the time complexity of FMGWO algorithm is consistent with the basic grey wolf optimization algorithm, and does not reduce the execution efficiency of the algorithm. The FMGWO algorithm and four representative comparison algorithms are tested on cec217 benchmark function and some engineering design optimization problems. The experimental results show that the optimization performance, solution stability and applicability to different problems of FMGWO algorithm are better than the other four comparison algorithms.

The rest of the paper is structured as follows:

Section 2 mainly introduces the basic principle of grey wolf optimization algorithm. In Section 3, FMGWO algorithm is proposed and the innovations are described. The pseudo code flow and time complexity of FMGWO algorithm are analyzed in Section 4, which shows that the improvement of the algorithm does not affect the operation efficiency. Section 5 introduces and analyzes the test results of five representative comparison algorithms on multiple different characteristics and different dimensions of CEC2017 benchmark functions. The test results show FMGWO algorithm has high optimization accuracy, convergence speed and solution stability. At the same time, the applicability of FMGWO algorithm in dealing with practical application problems is verified by two engineering design constraint optimization problems of pressure vessel and tensile tension/compression spring design. Section 6 summarizes the research content of this paper, and puts forward ideas and prospects for the next work.

2 The Basic Grey Wolf Optimizer

The flow of the basic Grey Wolf Optimizer is as follows:

Step 1: Initialize the population of the grey wolf individuals, the initial position of each grey wolf is generated randomly in the solution space $\overrightarrow{X_i}(i=1,2,\cdots,n)$, initialize parameter vectors $\vec{a}$, $\vec{A}$ and $\vec{C}$, and maximum iterations is represented as Max_iter.

Where, $\vec{A}=2\cdot\vec{a}\cdot\vec{r_1}-\vec{a}$, $\vec{C}=2\cdot\vec{r_2}$, the value of $\vec{a}$ decreases linearly from 2 to 0 as the number of iterations increases, $\vec{r_1}$ and $\vec{r_2}$ are two numbers generated randomly in the range of [0,1]. When $|A|\le 1$, grey wolf individuals conduct local search and attack prey intensively. When $|A|>1$, the wolves will disperse, and then they conduct a global search.

Step 2: Calculate the fitness values of each grey wolf.

Step 3: Identify the current optimal solution α wolf, suboptimal solution β wolf and the third best solution δ wolf.

Step 4: Calculate the distance between individual and α, β δ wolves in the population according to equation (1).

$$\vec{D}_\alpha=|\vec{C}_1\cdot\vec{X}_\alpha-\vec{X}|,\ \vec{D}_\beta=|\vec{C}_2\cdot\vec{X}_\beta-\vec{X}|,\ \vec{D}_\delta=|\vec{C}_3\cdot\vec{X}_\delta-\vec{X}| \tag{1}$$

Step 5: The individual position of grey wolf is updated according to equations (2) and (3).

$$\vec{X}_1=\vec{X}_\alpha-\vec{A}_1\cdot(\vec{D}_\alpha),\vec{X}_2=\vec{X}_\beta-\vec{A}_2\cdot(\vec{D}_\beta),\vec{X}_3=\vec{X}_\delta-\vec{A}_3\cdot(\vec{D}_\delta) \tag{2}$$

Where $\vec{A}_1$, $\vec{A}_2$ and $\vec{A}_3$ are the three vectors of $\vec{A}$.

$$\vec{X}_i'=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \tag{3}$$

Step 6: Update parameter vectors $\vec{a}$, $\vec{A}$ and $\vec{C}$.

Step 7: Compare whether the current iteration number is within the maximum iteration number range, when $t<Max\_iter$, return Step 2.

Step 8: Determine the final optimal value and output.

3 A Guiding Optimization Mutation Grey Wolf Optimization Algorithm Integrating Flower Pollination Mechanism (FMGWO)

3.1 Integration of Flower Pollination Mechanism

1) Introducing Levy flight mechanism into global search.

Flower pollination algorithm (FPA) is a meta-heuristic algorithm to simulate the pollination of phanerogam in nature. The process of cross-pollination and self-pollination corresponds to the global search process and local search process respectively, and the two search methods are balanced by the conversion probability.

In the global search process of FPA, the step size of pollinator's jumping and flying obeys the Levy distribution. The large jump and uneven random moving step size produced by Levy flight make FPA have the strong global optimization ability. In this paper, the mechanism of FPA is integrated into the stage of searching prey of Grey Wolf Algorithm, which makes the individual of grey wolf carry out global search more fully, so as to improve the convergence performance of the algorithm. The mathematical model is as follows:

$$\vec{X}_i^t = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \vec{X}_i^{t+1} = \vec{X}_i^t + L \cdot (\vec{X}_{best}^t - \vec{X}_i^t) \tag{4}$$

Where X_i^{t+1}, X_i^t represent the solution of the $t+1$ generation and the t generation, respectively. The X_{best}^t is the global optimal solution of the generation t. Moreover, L is the levy step, and its calculation formula is as follows:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda / 2)}{\pi} \cdot \frac{1}{s^{1+\lambda}} \tag{5}$$

Where $\lambda = 3/2$, $\Gamma(\lambda)$ is standard gamma function. s is the moving step size and its calculation formula is as follows:

$$s = \frac{U}{|V|^{1/\lambda}}, \quad U \sim N(0, \sigma^2) \quad V \sim N(0,1) \tag{6}$$

Where U obeys the Gaussian distribution with a mean of zero and a standard deviation of δ^2. Moreover, V obeys the standard Gaussian distribution.

2) Adding double random mechanism to local search.

The double random mechanism is used in local search process of FPA, which has better ability to get rid of local extremum. Integrating this mechanism into the stage of surrounding prey of Grey Wolf Algorithm can improve the local search ability of Grey Wolf Algorithm, enhance the population activity and avoid falling into local optimum. The mathematical model after introducing this mechanism is as follows:

$$\vec{X}_i^t = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \vec{X}_i^{t+1} = \vec{X}_i^t + \varepsilon(\vec{X}_k^t - \vec{X}_j^t) \tag{7}$$

Where X_i^{t+1}, X_i^t represent the solutions of the $t+1$ generation and the t generation of individual i, respectively. X_k^t, X_j^t are the solutions different from X_i^t randomly selected from the population. ε is the reproduction probability, which is a random number uniformly distributed in [0,1].

3.2 Increase Teaching Mechanism for Alpha Wolf

The Grey Wolf Algorithm adopts a hierarchical system, and the first three best wolves are considered as α, β and δ, which lead the other wolves to search the region close to optimal solution in the search space. In this paper, the algorithm combines teaching and learning to optimize the position update strategy in the teaching stage [26, 27]. The optimal average difference mechanism is added into the influence of the optimal solution wolf on the individual position update, so that other wolves can update the position according to the average difference between the α wolf and the individual position of the population. Therefore, it can speed up the grey wolf to approach the optimal value region, which could perfect the algorithm's convergence performance. The mathematical model is as follows:

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_1 = \vec{X}_1 + rand \cdot (\vec{X}_\alpha - TF \cdot Mean) \tag{8}$$

Where teaching factors $TF{=}round[1+rand(0,1)]$. Mean is average value of individual position in population.

3.3 Adding Polynomial Mutation Mechanism

In the optimization process of swarm intelligence algorithm, when it appears premature convergence, the mutation operation is used to randomly disturb the solution, which can increase the diversity of solutions in the population. Moreover, it could improve the algorithm's activity and the ability to get rid of local extremum. Polynomial mutation is an effective mutation mechanism for multi-objective optimization [28, 29]. For solving the problem that GWO is easy to fall into local extremum and low optimization accuracy, the polynomial mutation process is added after grey wolf individual position update, and the mutation formula is as follows:

$$x_i^{t+1} = x_i^t + \delta \cdot (ub - lb) \tag{9}$$

Where *ub* and *lb* are the upper and lower bounds of the location boundary, respectively. δ is disturbance factor, the calculation formula is as follows:

$$\delta = \begin{cases} [2u + (1-2u)(1-\delta_1)^{\eta+1}]^{1/(\eta+1)} - 1, & u \le 0.5 \\ 1 - [2(1-u) + 2(u-0.5)(1-\delta_2)^{\eta+1}]^{1/(\eta+1)}, & u > 0.5 \end{cases} \tag{10}$$

Where u is a random number evenly distributed in the range of [0,1]. δ_1=(x$_{best}^t$-lb)/(ub-lb), δ_2=(ub-x$_{best}^t$)/(ub-lb). η is distribution index, and $\eta = \frac{t}{T_{\max}} \cdot \eta_{\max}$, $\eta_{\max}$ generally take 30-50 [28, 29].

In the improved algorithm, the above polynomial mutation mechanism is adopted for the bad solution. If the mutated solution is better, the original solution will be replaced, otherwise it will not be replaced. That is, if the updated objective function value of the i grey wolf is worse than that of the i-1 grey wolf, the new position is obtained by polynomial mutation of the updated position of the i grey wolf. Then the objective function values before and after individual i position variation are compared. If the result after mutation is better, the result will be accepted, otherwise the original solution is retained. This mechanism not only enhances the population activity, but also retains the good mutation results, which improves the optimization performance of the algorithm as a whole.

4 Pseudo Code and Time Complexity Analysis of FMGWO

4.1 Pseudo Code of FMGWO

Algorithm 1. Pseudo code of FMGWO

```
Generate initial population of wolves individuals X_i (i = 1,2,...,n)
   Initialize parameter vector quantity a, A and C
   Calculate the fitness value of the objective function for each wolves
   Find the location of wolf α, wolf β and wolf δ
   t =0
   while（t < Max_iter）
      for i = 1:n
            According to equation (1) calculating the distance between individual and α, β, δ
            According to equation (8) calculating X_1
            According to equation (2) calculating X_2 and X_3
```

```
if |A| > 1
    Update the position of individual wolf by the equation (4)
  else
    Update the position of individual wolf by the equation (7)
end if
  if i > 1
    if Leader_score ≥ Leader_score_last
      Perform position mutate according to equation (9)
      Calculate the objective function value after position mutate newLeader_score
        if newLeader_score < Leader_score
              Retain the mutated solution
      else
              Retain the original solution
      end if
    end if
  end if
end for
Update a, A and C
Calculate the fitness value of the objective function for all wolves
t=t+1
end while
Output optimization result X_α
return
```

4.2 Analysis of Time Complexity

To evaluate an algorithm's advantages and disadvantages, it is essential to consider both optimization performance and time complexity. The time complexity can reflect the algorithm's efficiency. The time complexity of Firefly Algorithm and Cuckoo Search Algorithm are analyzed respectively in reference [30] and reference [31]. The same idea is used to analyze the time complexity of FMGWO in this study.

In the basic GWO algorithm, if the population scale is N, n is the dimension of individual position. The time for setting the initial parameters is t_0, the initialization time of each dimension in the individual position of grey wolf is t_1.Therefore, the time complexity in this period is as follows:

$$T_1 = O(t_0 + N \cdot n \cdot t_1) = O(n) \tag{11}$$

After entering iteration, the total number of iterations is Max_iter. It is assumed that the time of boundary treatment for each dimension of grey wolf in the population is t_2. The time to calculate the coefficient vectors $\vec{A}$ and $\vec{C}$ is t_3, $f(n)$ is the time for calculating the fitness value of the objective function. Finally, the time of α wolf, β wolf, δ wolf is t_4 according to the fitness value. Then the time complexity in this phase is as follows:

$$T_2 = O(N(n \cdot t_2 + t_3 + f(n)) + t_4) = O(n + f(n)) \tag{12}$$

It is assumed that according to equation (1), the time of distance between each individual and wolf α, β, δ is t_5. The time of position updating by equation (2) is t_6. And then the average time obtained by equation (3) is t_7. As a result, the time complexity in this phase is as follows:

$$T_3 = O(N(3 \cdot t_5 + 3 \cdot t_6 + t_7) = O(1) \tag{13}$$

To sum up, the total time complexity of WGO is as follows:

$$T=T_1+Max\_iter(T_2+T_3)=O(n+f(n)) \quad \textbf{(14)}$$

In the FMGWO algorithm, N is the population scale and the dimension is n, which are same as GWO algorithm. Moreover, the initialization process of the two algorithms and the time complexity in this stage are the same, as follows:

$$T_1=O(t_0+N\cdot n\cdot t_1)=O(n) \quad \textbf{(15)}$$

After entering iteration, the total number of iterations is Max_iter. The time of boundary treatment for each dimension of grey wolf in the population is t_2. The time to calculate the coefficient vectors $\vec{A}$ and $\vec{C}$ is t_3. The time for calculating the objective function's fitness value is $f(n)$. The time of α wolf, β wolf, and δ wolf according to the fitness value is t_4. Then the time complexity in this phase is as follows:

$$T_2=O(N(n\cdot t_2+t_3+f(n))+t_4)=O(n+f(n)) \quad \textbf{(16)}$$

In the same way, the time of distance between each individual and wolf α, β, δ is t_5. It is assumed that the time for calculating $\overrightarrow{X_1}$ by equation (8) is η_1. And then the time for calculating $\overrightarrow{X_2}$ or $\overrightarrow{X_3}$ by equation (2) is η_2. If the number of individuals for global search is m ($0\le m\le N$), then the number of individuals for local search in the population is $(N-m)$. It is assumed that the time for generating the control step size of Levy flight mechanism from equation (5) is η_3, and the time of introducing levy to update the position by equation (4) is η_4. Moreover, the time of location update by introducing double random mechanism by equation (7) is η_5. As a result, the time complexity in this phase is as follows:

$$T_3'=O(N(3\cdot t_5+\eta_1+2\cdot\eta_2)+m\cdot(\eta_3+\eta_4)+(N-m)\cdot\eta_5)=O(1) \quad \textbf{(17)}$$

It is assumed that the time for calculating the fitness value of individual objective function after location update is $f(n)$.The time to compare fitness values of two individuals is η_6. The number of individuals performing polynomial variation is q ($0\le q<N$). The time for calculating the variation disturbance factor δ by equation (10) is η_7. The time of mutation for each individual by equation (9) is η_8. Therefore, the time complexity in this phase is as follows:

$$T_4'=O(N\cdot(f(n)+\eta_6)+q\cdot(\eta_7+\eta_8)=O(f(n)) \quad \textbf{(18)}$$

In summary, the total time complexity of FWGO is as follows:

$$T'=T_1+Max\_iter(T_2+T_3'+T_4')=O(n+f(n)) \quad \textbf{(19)}$$

Based on this, the improved algorithm FGWO and GWO algorithm have the same time complexity, and the efficiency of the algorithm is not debased.

5 Simulation Experiment

Fourteen CEC2017 benchmark functions with different optimization characteristics are selected to test the optimization ability of the improved algorithm. The FMGWO algorithm in this study, the basic Grey Wolf algorithm (GWO, 2014), Augmented Grey Wolf Oprimizer Algorithm (AGWO, 2018) [11], memory-based Grey Wolf Optimizer (mGWO, 2020) [12] and Salp Swarm Algorithm (SSA, 2017) are compared and tested on 10, 50 and 100 dimensions.

To ensure the fairness and the experiment's objectivity, the five comparison algorithms use the same software and hardware platform, the running environment is Windows10 operating system, and the programming language is MATLAB R2019a. In the simulation experiment, the five algorithms run independently for 50 times under the same conditions, 30 is the population size, and the maximum evolutionary algebra $MaxG = 1000$. The parameter settings of FMGWO, AGWO and mGWO are consistent with the original setting of basic Grey Wolf algorithm (GWO), and the SSA algorithm does not need to set additional parameters.

5.1 Test Function

The specific test functions are described in Table 1. The 14 test functions in Table 1 are all CEC 2017 benchmark functions. Among them, $f_1(x) \sim f_5(x)$ is a complex unimodal function, which has no local minimum. It is mainly used to verify the accuracy and convergence rate of the algorithm. $f_6(x) \sim f_{14}(x)$ is a complex multimodal function, which has many local minima. It can be used to test the algorithm's ability to jump out of local extremum and global search ability.

Table 1. Test function

Serial number	Function	Function formula	Optimal value
$f_1(x)$	Bent Cigar	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	0
$f_2(x)$	Sum of Different Power	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	0
$f_3(x)$	Zakharov	$f_3(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$	0
$f_4(x)$	High Conditioned Elliptic	$f_4(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	0
$f_5(x)$	Discus	$f_5(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$	0
$f_6(x)$	Rosenbrock's	$f_6(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2) + (x_i - 1)^2)$	0
$f_7(x)$	Rastigin's	$f_7(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	0
$f_8(x)$	Expanded Schaffer's F6	$g(x,y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1 + 0.001(x^2+y^2))^2}$	0
		$f_8(x) = g(x_1,x_2) + g(x_2,x_3) + \ldots + g(x_{D-1},x_D) + g(x_D,x_1)$	0
$f_9(x)$	Non-continuous Rotated Rastrigin's	$f_9(x) = \sum_{i=1}^{D} \left[y_i^2 - 10\cos(2\pi y_i) + 10 \right]$	0

		$y_i = \begin{cases} x_i \rightarrow \lvert x_i \rvert < \frac{1}{2} \\ round(2x_i)/2 \rightarrow \lvert x_i \rvert >= \frac{1}{2} \end{cases}$	
$f_{10}(x)$	ACKley's	$f_{10}(\mathrm{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	0
$f_{11}(x)$	Griewank's	$f_{11}(\mathrm{x}) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	0
$f_{12}(x)$	HappyCat	$f_{12}(x) = \left\lvert \sum_{i=1}^{D} x_i^2 - D \right\rvert^{\frac{1}{4}} + (0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/\mathrm{D} + 0.5$	0
$f_{13}(x)$	HGBat	$f_{13}(x) = \left\lvert (\sum_{i=1}^{D} x_i^2)^2 - (\sum_{i=1}^{D} x_i)^2 \right\rvert^{\frac{1}{2}} + (0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/\mathrm{D} + 0.5$	0
$f_{14}(x)$	Schaffer's	$f_{14}(\mathrm{x}) = \left[\frac{1}{D-1}\sum_{i=1}^{D-1}(\sqrt{s_i} * (\sin(50.0 s_i^{0.2}) + 1))\right]^2$	0

5.2 Analysis of Optimization Accuracy

To test the improved algorithm's optimization performance, the dimension of the test function $f_1(x)$, $f_{14}(x)$ is set to d = 10 / 50 / 100, and tested in three different dimensions: low, medium and high. Table 2 shows the best solution, average value and variance of the optimization results obtained by the five algorithms running independently 50 times in different dimensions of each function.

Table 2. Comparison of optimization performance of five algorithms under fixed iteration times

Function	Algorithm	dim=10			dim=50			dim=100		
		Optimal solution	Average value	Variance	Optimal solution	Average value	Variance	Optimal solution	Average value	Variance
$f_1(x)$	FMGWO	2.59E-234	1.31E-227	0	2.83E-116	3.80E-113	1.69E-224	5.07E-93	2.16E-90	3.56E-179
	AGWO	3.65E-183	5.14E-151	9.15E-300	1.58E-64	7.99E-61	6.61E-120	2.49E-40	1.88E-37	1.40E-73
	mGWO	8.45E-133	1.04E-122	5.33E-243	1.07E-45	1.85E-43	3.90E-85	1.16E-29	2.32E-28	1.10E-55
	GWO	2.69E-121	3.24E-113	2.16E-224	1.05E-41	8.81E-40	2.99E-78	7.25E-27	1.37E-25	1.65E-50
	SSA	2.12E-02	1.67E+01	4.84E+02	6.08E-04	7.09E+00	1.22E+02	2.37E+03	1.92E+04	2.15E+08
$f_2(x)$	FMGWO	0	0	0	0	0	0	0	0	0
	AGWO	0	0	0	0	0	0	0	0	0
	mGWO	0	0	0	0	0	0	0	0	0
	GWO	0	0	0	0	0	0	0	0	0
	SSA	3.53E-21	2.02E-16	1.55E-31	3.46E-20	1.83E-16	2.77E-31	1.79E-20	2.18E-16	2.82E-31
$f_3(x)$	FMGWO	6.98E-169	4.22E-159	6.21E-316	3.96E-56	2.05E-49	5.42E-97	2.79E-33	3.55E-24	5.94E-46
	AGWO	6.46E-132	1.58E-122	6.38E-243	6.25E-22	1.07E-16	4.24E-31	1.01E-06	6.18E-03	2.88E-04
	mGWO	4.61E-89	7.34E-80	6.27E-158	3.49E-13	6.09E-10	4.98E-18	1.33E-02	4.13E-01	3.70E-01
	GWO	2.66E-76	6.17E-70	9.70E-138	1.15E-10	9.16E-08	8.63E-14	8.53E-02	1.73E+00	8.52E+00
	SSA	3.82E-12	1.25E-11	2.03E-23	6.45E+01	1.42E+02	2.09E+03	8.79E+02	1.31E+03	3.51E+04
$f_4(x)$	FMGWO	7.88E-237	1.79E-226	0	7.42E-117	6.08E-114	9.30E-226	2.33E-93	3.73E-91	6.13E-181
	AGWO	3.19E-176	4.54E-150	8.64E-298	2.52E-64	3.89E-61	3.17E-120	6.43E-42	1.23E-38	1.52E-75
	mGWO	1.69E-133	1.64E-124	4.20E-247	1.45E-46	3.98E-44	1.01E-86	4.79E-30	5.70E-29	2.57E-57
	GWO	7.49E-120	3.76E-113	1.79E-224	1.76E-42	2.62E-40	1.58E-79	2.25E-27	3.50E-26	1.48E-51
	SSA	6.09E+03	3.69E+05	9.08E+10	3.56E+06	1.47E+07	4.08E+13	2.32E+07	5.04E+07	2.77E+14
$f_5(x)$	FMGWO	1.65E-236	8.38E-228	0	8.06E-120	1.52E-117	1.67E-233	1.80E-96	2.60E-94	3.91E-187
	AGWO	1.82E-173	5.34E-154	1.18E-305	3.78E-68	1.02E-63	1.47E-125	1.42E-43	1.84E-41	1.65E-81
	mGWO	5.84E-135	1.49E-127	8.13E-253	2.72E-49	3.16E-47	1.42E-92	1.78E-33	5.38E-32	6.51E-63
	GWO	6.00E-122	4.68E-116	4.96E-230	4.49E-45	2.42E-43	1.73E-85	2.06E-30	2.54E-29	8.96E-58
	SSA	5.48E+02	5.47E+03	1.43E+07	9.14E+03	3.83E+04	5.35E+08	2.58E+04	8.58E+04	2.00E+09
$f_6(x)$	FMGWO	0	9.94E-02	3.87E-01	0	0	0	0	0	0
	AGWO	0	0	0	0	2.85E-08	1.42E-06	0	2.73E-10	2.19E-18
	mGWO	3.29E-28	2.70E-01	1.02E+00	6.18E-30	3.53E-05	1.44E-03	1.30E-27	1.05E-08	5.46E-15
	GWO	1.98E-15	1.66E+00	7.10E+00	7.42E-30	4.16E-06	1.13E-04	6.12E-26	5.91E-10	8.56E-18
	SSA	5.12E-03	1.60E+01	9.29E+02	3.24E+01	1.19E+02	1.00E+04	3.85E+02	6.44E+02	3.43E+04

$f_7(x)$	FMGWO	0	0	0	0	0	0	0	0	0
	AGWO	0	0	0	0	0	0	0	0	0
	mGWO	0	3.24E+00	8.17E+00	0	1.19E+01	3.94E+01	5.33E-15	1.37E+01	1.03E+02
	GWO	0	6.30E-02	1.98E-01	0	7.44E-01	3.5250	0	4.88E-01	2.1821
	SSA	4.97E+00	1.65E+01	5.58E+01	4.48E+01	8.92E+01	5.58E+02	8.65E+01	1.71E+02	2.12E+03
$f_8(x)$	FMGWO	9.59E-06	1.69E-01	4.78E-02	3.34E-01	8.38E+00	2.97E+01	1.42E-04	1.20E+01	2.47E+02
	AGWO	0	6.75E-01	6.15E-01	3.65E+00	1.50E+01	1.17E+01	1.07E+01	3.47E+01	5.94E+01
	mGWO	3.91E-02	3.83E-01	1.04E-01	8.46E+00	1.35E+01	4.86E+00	2.48E+01	3.19E+01	1.28E+01
	GWO	4.12E-02	2.93E-01	3.53E-02	5.88E+00	1.21E+01	8.00E+00	1.83E+01	2.91E+01	2.50E+01
	SSA	1.36E+00	2.4617	2.08E-01	1.18E+01	1.70E+01	3.25E+00	2.96E+01	3.61E+01	9.2385
$f_9(x)$	FMGWO	0	0	0	0	0	0	0	0	0
	AGWO	0	0	0	0	7.17E-01	1.02E+01	0	1.81E+00	8.76E+01
	mGWO	0	5.80E-01	2.2894	2.00E+00	1.81E+01	8.70E+01	8.00E+00	3.18E+01	6.41E+02
	GWO	0	1.1432	5.1622	0	5.53E+00	6.42E+01	2.49E-14	8.68E	6.57E+01
	SSA	6.00E+00	3.80E+01	4.20E+02	4.65E+02	7.28E+02	1.88E+04	2.03E+03	2.93E+03	2.10E+05
$f_{10}(x)$	FMGWO	3.55E-15	3.55E-15	1.29E-59	7.11E-15	7.25E-15	1.01E-30	7.11E-15	1.17E-14	9.29E-30
	AGWO	0	2.91E-15	6.02E-30	7.11E-15	9.17E-15	8.29E-30	1.42E-14	1.79E-14	1.80E-29
	mGWO	3.55E-15	3.98E-15	1.36E-30	1.42E-14	2.44E-14	1.94E-29	4.97E-14	6.48E-14	4.36E-29
	GWO	0	3.48E-15	2.52E-31	2.84E-14	3.14E-14	1.25E-29	9.95E-14	1.12E-13	8.36E-29
	SSA	6.46E-06	5.88E-01	7.47E-01	1.27E+00	3.38E+00	8.56E-01	5.01E+00	7.23E+00	8.07E-01
$f_{11}(x)$	FMGWO	0	1.55E-03	2.83E-05	0	0	0	0	0	0
	AGWO	0	8.18E-02	2.57E-02	0	0	0	0	8.25E-04	1.68E-05
	mGWO	0	5.27E-02	1.11E-03	0	5.15E-03	5.66E-05	0	3.89E-03	5.98E-05
	GWO	0	1.62E-02	5.75E-04	0	1.34E-03	1.67E-05	0	0	0
	SSA	2.70E-01	3.13E-02	2.96E-02	5.40E-04	1.03E-02	7.51E-05	2.75E-01	7.15E-01	4.82E-02
$f_{12}(x)$	FMGWO	2.42E-02	6.35E-02	5.19E-04	4.45E-02	1.24E-01	1.79E-03	6.94E-02	1.68E-01	2.70E-03
	AGWO	3.52E-02	2.72E-01	1.45E-02	1.11E-01	2.31E-01	6.74E-03	1.79E-01	2.80E-01	4.22E-03
	mGWO	3.57E-02	1.11E-01	2.12E-03	1.14E-01	2.68E-01	1.11E-02	1.31E-01	2.69E-01	5.22E-03
	GWO	2.57E-02	7.78E-02	8.52E-04	8.57E-02	2.14E-01	4.15E-03	1.35E-01	2.61E-01	4.68E-03
	SSA	1.19E-01	4.02E-01	2.70E-02	3.75E-01	5.97E-01	1.11E-02	4.52E-01	5.98E-01	7.00E-03
$f_{13}(x)$	FMGWO	5.05E-03	2.89E-02	1.53E-03	1.40E-02	1.01E-01	2.00E-03	3.78E-02	1.36E-01	4.00E-03
	AGWO	8.45E-06	9.85E-02	6.68E-03	7.18E-02	1.57E-01	3.26E-03	9.09E-02	2.08E-01	5.79E-03
	mGWO	1.76E-02	1.32E-01	6.82E-03	6.23E-02	1.44E-01	1.63E-03	7.82E-02	1.39E-01	1.53E-03
	GWO	8.42E-03	9.67E-02	6.26E-03	7.76E-02	1.46E-01	1.33E-03	8.00E-02	1.38E-01	9.52E-04
	SSA	1.34E-01	4.08E-01	6.09E-02	3.04E-01	6.60E-01	9.41E-02	3.31E-01	5.54E-01	4.37E-02
$f_{14}(x)$	FMGWO	9.07E-121	5.11E-116	2.39E-230	6.45E-65	1.17E-63	4.10E-126	2.25E-53	3.06E-52	1.29E-103
	AGWO	4.13E-133	1.12E-111	6.05E-221	4.50E-42	3.00E-39	2.80E-77	5.37E-28	6.99E-27	7.36E-53
	mGWO	1.12E-82	1.65E-75	7.79E-149	1.67E-29	1.47E-27	1.59E-53	2.86E-20	8.80E-19	2.78E-36
	GWO	3.04E-67	5.95E-64	1.63E-126	8.29E-26	9.23E-25	1.11E-48	8.84E-18	3.73E-17	9.66E-34
	SSA	6.94E-01	6.10E+00	1.85E+01	1.54E+01	2.16E+01	9.89E+00	1.90E+01	2.40E+01	6.71E+00

By observing the data in Table 2, it can be seen that the FMGWO algorithm has higher optimization accuracy in different dimensions of each function. Moreover, its optimization effect is obviously better than AGWO, mGWO, GWO and SSA. Especially in the various dimensions of function $f_2(x)$, $f_7(x)$, $f_9(x)$, the optimal solution and average value of the optimization results obtained by 50 runs are theoretical optimal values.

The optimization results of five algorithms in 10 dimensions could be observed from Table 2. For $f_2(x)$, $f_7(x)$, $f_9(x)$ three functions, the best solution and average value of FMGWO algorithm running 50 times are theoretical optimal values. For $f_6(x)$, $f_{10}(x)$ functions，the optimization result of FMGWO algorithm is slightly lower than that of AGWO algorithm. However, it is better than mGWO, GWO and SSA algorithms. For the $f_8(x)$, $f_{14}(x)$ functions, the accuracy of the best solution obtained by 50 runs of FMGWO algorithm is slightly lower than that of AGWO algorithm, and higher than that of mGWO, GWO and SSA algorithms. However, the average value and variance of FMGWO algorithm are better than the four comparison algorithms. For the six functions $f_1(x)$, $f_3(x)$, $f_5(x)$, $f_{11}(x)$, $f_{13}(x)$, FMGWO algorithm is far better than the other four algorithms in terms of optimal solution, mean value and variance, and has better optimization ability. Although AGWO and mGWO are superior to GWO and SSA in most functions, they are generally inferior to FMGWO. Moreover, the variance of the results of 50 runs of FMGWO on each function is generally smaller than that of AGWO and mGWO. In contrast, FMGWO algorithm has better optimization ability and better stability.

The five algorithms' optimization results under the conditions of 50 and 100 dimensions could be observed in Table 2. For the functions $f_2(x)$, $f_6(x)$, $f_7(x)$, $f_9(x)$, $f_{11}(x)$, the optimal solution and average value obtained by FMGWO algorithm running 50 times in 50 and 100 dimensions are both theoretical optimal values. For function $f_8(x)$, the optimization accuracy of FMGWO algorithm in 50 dimensions is slightly lower than that of the other four comparison algorithms. However, the optimal solution, average value and variance of FMGWO algorithm in 100 dimensions are better than those of the other four comparison algorithms. For the six functions such as $f_1(x)$, $f_3(x)$, $f_5(x)$, $f_{10}(x)$, $f_{12}(x)$, $f_{14}(x)$, when the five algorithms run 50 times in 50 and 100 dimensional conditions, the optimal solution, average value and variance of FMGWO algorithm are better than the other four comparison algorithms. For most functions in the two dimensions, AGWO and mGWO algorithms have higher optimization accuracy than GWO and SSA algorithms. However, the optimization accuracy of AGWO and mGWO algorithms is higher than that of FMGWO algorithm only in the condition of 50 dimension function. In other cases, the optimization accuracy of the two algorithms is lower than that of FMGWO algorithm. More importantly, the optimal solution and average value of mGWO algorithm are even lower than those of GWO algorithm under 50 and 100 dimensions of $f_6(x)$, $f_7(x)$, $f_9(x)$ functions. It shows that the adaptability and stability of high-dimensional solution are not enough. The experimental results show that as the dimensions rise, the optimization accuracy of the five algorithms will decline in most functions. However, the FMGWO algorithm still has strong solution ability and high stability in high-dimensional optimization.

It could be described from the above optimization results and analysis that the FMGWO's accuracy and stability are better than the other four algorithms on three different dimensions of $d = 10/50/100$. Consequently, it can solve the problems of low precision and poor stability of high-dimensional solution in complex function optimization problems.

5.3 Analysis of Convergence Curve

To compare the optimization performance of the five algorithms more intuitively，the convergence rate and the ability to get rid of local extremum are analyzed. Fig. 1 to Fig. 9 describe the solving convergence curves of FMGWO, AGWO, mGWO, GWO and SSA algorithms for $f_1(x) \sim f_{14}(x)$ functions with the same iteration times and dimension of 100. Due to the complexity of multi-dimensional and multi-modal functions, the algorithm will sink into the local optimal value easily. Therefore, the convergence of these functions can better illustrate the algorithm's optimization ability. The following only lists the convergence curves of all 9 multimodal functions among the 14 test functions, while the convergence curves of unimodal functions are relatively simple and the results are similar to those of multimodal functions, so they are not redundant.

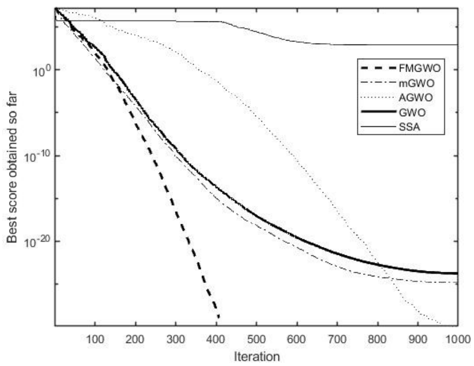

Fig. 1. The function $f_6(x)$'s convergence cure

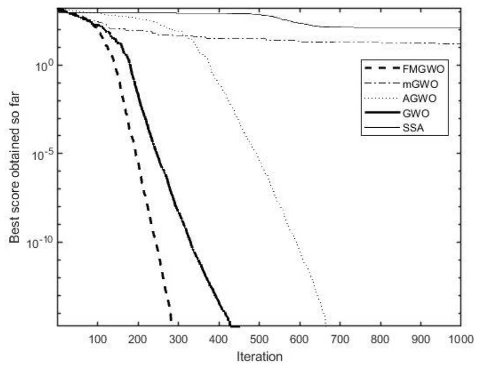

Fig. 2. The function $f_7(x)$'s convergence curve

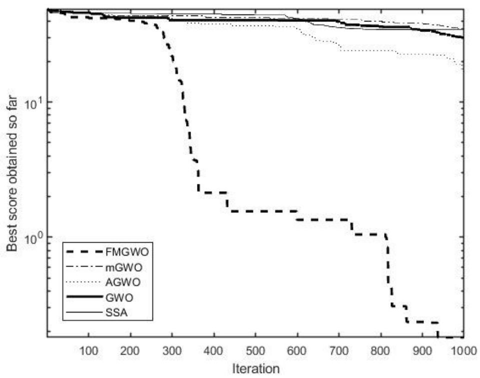

Fig. 3. The function $f_8(x)$'s convergence curve

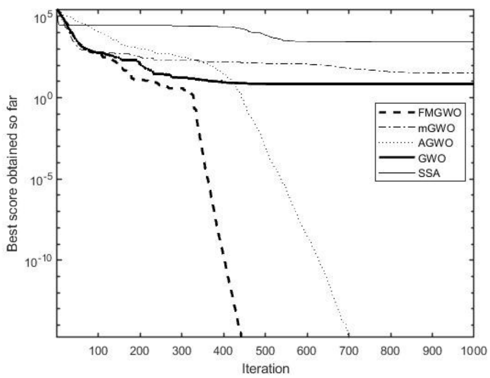

Fig. 4. The function $f_9(x)$'s convergence curve

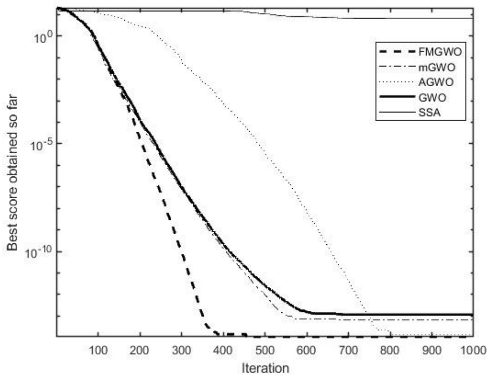

Fig. 5. The function $f_{10}(x)$'s convergence curve

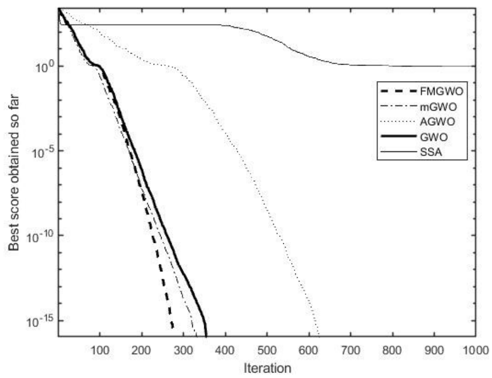

Fig. 6. The function $f_{11}(x)$'s convergence curve

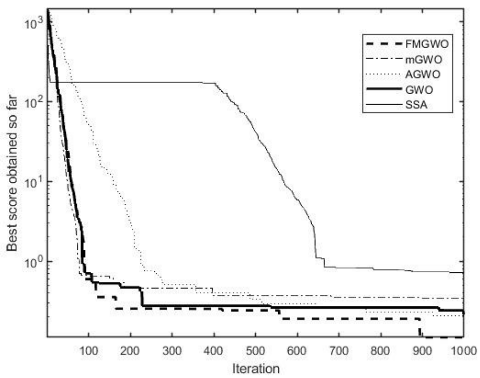

Fig. 7. The function $f_{12}(x)$'s convergence curve

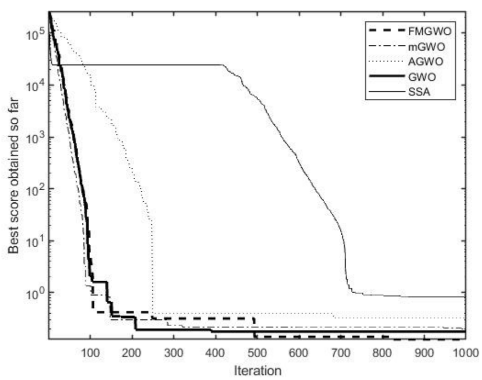

Fig. 8. The function $f_{13}(x)$'s convergence curve

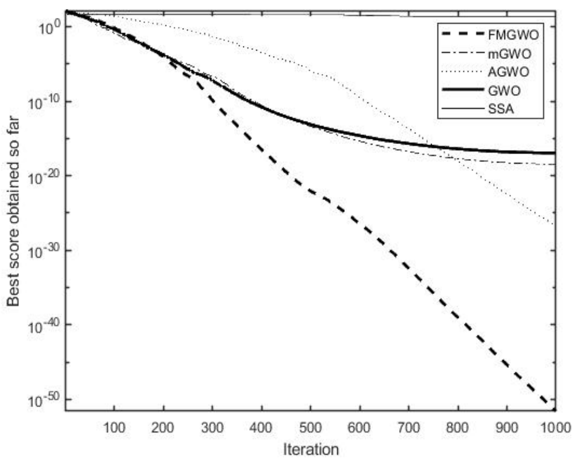

Fig. 9. The function $f_{14}(x)$'s convergence curve

For Fig. 1, Fig. 2, Fig. 4 and Fig. 6, the FMGWO algorithm has always maintained a fast convergence rate and smooth convergence curve. Fig. 1 shows that it converges to the theoretical optimal solution in the 400th generation. From Fig. 2 and Fig. 6, it converges to the theoretical optimal solution in the 300th generation. In Fig.4, it is found to the theoretical optimal solution in the 450th generation. The convergence velocity of AGWO algorithm is generally slower than that of GWO algorithm in the prophase of iteration. However, it can jump out of local extremum in the later stage. In Fig. 1, it converges to the ground of theoretical optimal value at the finish of iteration. Fig. 2 and Fig. 4 show that it could converge to the ground of theoretical optimal value at the 700th generation. It could be seen from Fig. 6 that it converges to the ground of theoretical optimal value at the 600th generation. The mGWO algorithm and SSA algorithm in Fig. 2 and Fig. 4 have fallen into local extremum from the beginning of iteration, and the downward trend of convergence curve is not obvious. Fig. 1 and Fig. 6 show that the convergence velocity of mGWO algorithm is faster than that of GWO, AGWO and SSA algorithm. However, it is still slower than FMGWO algorithm. As for Fig. 3, the convergence rate of GWO, AGWO, mGWO and SSA algorithms is very slow from the beginning to the end of iteration. The downward trend of convergence curve is not obvious, and the final convergence accuracy is not high. However, FMGWO algorithm converges quickly after the 300th generation. It can jump out of local optimal value in the late iteration, and converge to the ground of theoretical optimal solution near the 1000th generation. Fig. 5 shows that the FMGWO algorithm converges quickly. Moreover, it converges to the ground of theoretical optimal solution when the iteration reaches the 400th generation. The mGWO and GWO algorithms sink into the local optimal value in the 600th generation, which cannot jump out until the finish of the iteration. Although the convergence rate of AGWO algorithm is slow, it can converge to the ground of theoretical optimal solution at the 800th generation. The convergence curve of SSA algorithm is almost horizontal and the downward trend is not obvious. As for Fig. 7 and Fig. 8, the convergence rates of FMGWO, mGWO and GWO algorithms are basically the same in the prophase of iteration. In the later stage of iteration, FMGWO algorithm can jump out of local extremum and converge to the ground of theoretical optimal solution at the end of iteration. However, GWO, AGWO and mGWO algorithms fall into local extremum and cannot jump out at the 800th generation. SSA algorithm falls into local extremum when iteration reaches the 400th generation. Moreover, it does not jump out until the end of iteration, and the optimization accuracy is low. Fig. 9 describes that FMGWO algorithm has the fastest convergence speed among the five algorithms. The optimal solution found at the end of iteration is near the theoretical optimal solution. The convergence rate of GWO, AGWO and mGWO algorithms are relatively slow. They all fail to reach the theoretical optimal solution at the end of iteration. However, the convergence curve of SSA algorithm is almost horizontal.

From the above analysis, it can be proposed that the FMGWO algorithm's optimization capability is obviously better than the other four comparative algorithms in high-dimensional conditions. This is mainly because in the process of optimization, the solution accuracy and algorithm's convergence speed are improved through adding teaching mechanism to α wolf and using Levy flight mechanism in global search. Adding double random mechanism in local search and adding polynomial variation mechanism after position updating can effectively avert the problem that the basic algorithm will fall into local extremum easily.

In conclusion, the FMGWO algorithm raised in this paper has better performance in three dimensions of low, medium and high. Moreover, its solving accuracy, convergence rate and optimization stability are better than AGWO, mGWO, GWO and SSA.

5.4 FMGWO for Solution and Analysis of Classical Engineering Problems

In order to verify the effectiveness of FMGWO algorithm in solving engineering constrained optimization problems, well-known standard engineering optimization design problems such as Pressure vessel design and tension/compression spring design. The results of the FMGWO are compared with the state-of-the-art metaheuristic algorithms: Grey Wolf Algorithm (GWO, 2014), Augmented Grey Wolf Oprimizer Algorithm (AGWO, 2018), memory-based Grey Wolf Optimizer (mGWO, 2020) and salp group algorithm (SSA, 2017). These two engineering problems have their own different constraints. This section uses a general death penalty function mechanism to deal with these constraints, so that the algorithm automatically discards infeasible solutions during the optimization process.When solving these engineering optimization problems, each algorithm runs independently 50 times to select the best optimization design result.In all experiments, the parameters of the comparison algorithms were the same as those recommended in the original books.

1) Pressure vessel design

For the common pressure vessel structure in engineering, the dynamic model that can be established is shown in Fig. 10.

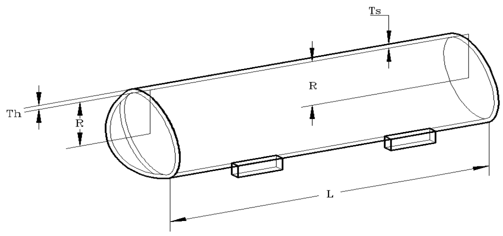

Fig. 10. Schematic diagram of pressure vessel structure

The goal of pressure vessel design problem is to design the pressure vessel with the lowest manufacturing cost by looking for the global optimal value. This problem has four constraints and four design variables. The design variables are shell thickness $T_s(0 \le T_s \le 99)$, head thickness $T_h(0 \le T_h \le 99)$, shell radius $R(10 \le R \le 200)$ and cylindrical section length $L_s(10 \le L_s \le 200)$, and the specific mathematical model is as follows:

Consider $x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L_s]$

Minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2$

Subject to $g_1(x) = -x_1 + 0.0193x_3 \le 0$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(x) = x_4 - 240 \le 0$$

Table 3 lists the best solutions obtained when the five algorithms run independently for 50 times to solve the pressure vessel problem. It can be seen from the data in the table that the minimum cost $f(x)$=5887.825646 obtained by the algorithm FMGWO in this paper at T_s = 0.778400, T_h= 0.385147, R= 40.324845, L_s= 199.931022 is the minimum cost of these algorithms to solve the pressure vessel design problem. The cost of the FMGWO algorithm is slightly lower than that of the mGWO algorithm and significantly lower than the other three algorithms. It shows that the FMGWO algorithm has a better ability to solve such engineering design constraint optimization problems.

Table 3. Results for the pressure vessel problem

Algorithm	x_1	x_2	x_3	x_4	Optimum Cost
FMGWO	0.778400	0.385147	40.324845	199.931022	5887.825646
AGWO	0.781128	0.386234	40.453843	200.000000	5933.504730
mGWO	0.778533	0.384998	40.335777	199.852079	5888.352199
GWO	0.779070	0.385062	40.353781	199.634728	5890.825931
SSA	0.782775	0.386926	40.558289	197.238984	5904.863276

2) Tension/compression spring design

For the common tensile tension/compression spring design structure in engineering, the dynamic model that can be established is shown in Fig. 11.

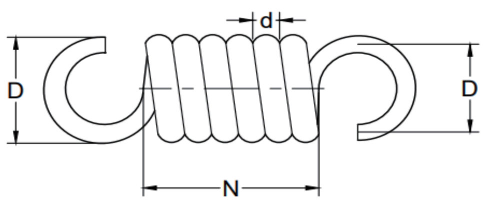

Fig. 11. Schematic diagram of tensile tension/compression spring

The design of tension spring is a problem of minimizing constraints. Its purpose is to design a tension spring with the lightest weight and meeting the four constraints of deflection, shear stress, fluctuation frequency and outer diameter. The problem has three design variables: wire diameter $d(0.05 \le d \le 2.00)$, mean coil diameter $D(0.25 \le D \le 1.30)$, number of active coils $N(2.00 \le N \le 15.00)$ and the mathematical model of the tension spring design problem is as follows:

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

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

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

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

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

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

Table 4 shows the optimal solutions of the minimum weight and corresponding design variables of FMGWO algorithm and other four algorithms to solve the design problem of tension spring. It can be seen from the data in this table that the final design results obtained by other algorithms are very different except that the solution results of AGWO algorithm are relatively poor. However, the spring weight solved by FMGWO algorithm is the smallest of the five algorithms, which shows that FMGWO algorithm still has good performance for solving this kind of engineering design problems.

Table 4. Results for the Table 3, results for the pressure vessel problem

Algorithm	x_1	x_2	x_3	Optimum Cost
FMGWO	0.050899	0.337592	12.533948	0.012672
AGWO	0.051190	0.344423	12.080950	0.012709
mGWO	0.052108	0.366792	10.725862	0.012674
GWO	0.052564	0.378106	10.152263	0.012695
SSA	0.051066	0.341912	12.216184	0.012675

The test results of the above engineering constrained optimization problems show that the improved algorithm FMGWO can find better solutions and provide superior solutions when solving engineering optimization design problems. The algorithm has good convergence performance and application potential.

6 Conclusions

In this article, the GWO algorithm is improved for its low accuracy of high-dimensional solution, slow convergence speed and easy to sink into local extremum. Firstly, the flower pollination mechanism is combined with GWO algorithm. The Levy distribution is introduced into the global search of grey wolf population. Moreover, the double random mechanism is added into the local search. Because of these improvements, the overall optimization performance of the algorithm and the ability to jump out of local extremum are improved. Secondly, the teaching mechanism is adopted for α wolf to make the individual of grey wolf approach the optimal value region to speed up the algorithm's convergence rate. Finally, polynomial mutation is introduced to improve the performance of the algorithm. It is proved that FMGWO algorithm and GWO algorithm have the same time complexity through theoretical analysis. The test results describe that the improved algorithm has obvious effectiveness in solving function optimization problems and engineering design constrained optimization problems. Although FMGWO algorithm has good optimization ability, it still cannot solve the theoretical optimal value in some test functions. It shows that there is still room for further improvement of the algorithm, which needs further research in the future. At the same time, with the continuous emergence and development of practical engineering problems, the optimization of problem solving in the latest field will still be an important aspect of future research work. The next work to be done is to continue to improve the grey wolf optimization algorithm and continuously improve the optimization performance of the algorithm. More importantly, it could be reasonably applied to the solution of more engineering problems.

7 Acknowledgement

This work is jointly supported by the Key R&D and Promotion Projects in Henan Province, China (grant number 182102310886), the Key Scientific Research Project of Universities in Henan Province (grant number 19A520016), the Postgraduate Education Innovation and Quality Improvement Project of Henan University (Grant No. SYL18020105 and SYL19050104), the Henan Higher Education Teaching Reform Research and Practice Project (grant number 2019SJGLX080Y).

References

[1] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks (1995) 1942-1948.
[2] X.S. Yang, S. Deb, Engineering Optimisation by Cuckoo Search, International Journal of Mathematical Modelling & Numerical Optimisation 1(4)(2010) 330-343.
[3] J.S. Liu, L. Liu, Y. Li, A Differential Evolution Flower Pollination Algorithm with Dynamic Switch Probability, Chinese Journal of Electronics 28(4)(2019) 737-747.
[4] S. Mirjalili, A. Lewis. The Whale Optimization Algorithm, Advances in Engineering Software 95(2016) 51-67.
[5] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Advances in Engineering Software (114)(2017) 163-191.
[6] H.M. Song, M.H. Sulaiman, M.R. Mohamed, An application of grey wolf optimizer for solving combined economic emission dispatch problems, International Review on Modelling and Simulations (IREMOS) 7(5)(2014) 838-844.
[7] G.M. Komaki, V. Kayvanfar, Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time, Journal of Computational Science (8)(2015) 109-120.
[8] W. Elloumi, H.E. Abed, A. Abraham, A.M. Alimi, A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP, Applied Soft Computing 25(2014) 234-241.
[9] S. Mirjalili, How effective is the grey wolf optimizer in training multilayer perceptrons, Applied Intelligence 43(2015) 150-161.
[10]P. Yao, H.L. Wang, Three-dimensional path planning for UAV based on improved interfered fluid dynamical system and grey wolf optimizer, Control and Decision 31(4)(2016) 701-708.
[11]M.H. Qais, H.M. Hasanien, S. Alghuwainem, Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems, Applied Soft Computing 69(2018) 504-515.
[12]S. Gupta, K. Deep, A memory-based Grey Wolf Optimizer for global optimization tasks, Applied Soft Computing 93(2020) 106367.
[13]M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Systems With Applications 166(2021) 113919.
[14]T.H. Jiang, Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm, Control and Decision 33(3)(2018) 503-508.
[15]W. Long, T.B. Wu, M.Z. Tang, M. Xu, S.H. Cai, Grey Wolf Optimizer Algorithm Based on Lens Imaging Learning Strategy, Journal of Automatica Sinica 46(10)(2020) 2148-2164.
[16]B.P. Sahoo, S. Panda, Improved grey wolf optimization technique for fuzzy aided PID controller design for power system frequency control, Sustainable Energy, Grids and Networks 16(2018) 278-299.
[17]W. Long, J.J. Jian, X.M. Liang, M.Z. Tang, Inspired grey wolf optimizer for solving large-scale function optimization problems, Applied Mathematical Modelling 60(2018) 112-126.
[18]A. Rodríguez, O. Camarena, E. Cuevas, I. Aranguren, A. Valdivia-G, B. Morales-Castañeda, D. Zaldívar, M. Pérez-Cisneros, Group-based Synchronous-Asynchronous Grey Wolf Optimizer, Applied Mathematical Modelling 93(2021) 226-243.
[19]C. Lu, L. Gao, Q.K. Pan, X.Y. Li, J. Zheng, A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution, Applied Soft Computing 75(2019) 728-749.
[20]G.W. Huang, Y.G. Cai, Y.H. Qi, H.R. Chen, S.H. Wang, Adaptive Genetic Grey Wolf Optimizer Algorithm for Capacitated Vehicle Routing Proble, Acta Electronica Sinica 47(12)(2019) 2602-2610.
[21]B. Zeng, J. Guo, W.Q. Zhu, Z.H. Xiao, F. Yuan, S.X. Huang, A Transformer Fault Diagnosis Model Based On Hybrid Grey Wolf Optimizer and LS-SVM, Energies 12(21)(2019) 4170
[22]G. Natesan, A. Chokkalingam, Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm, ICT Express 5(2)(2019) 110-114.
[23]A.A. Bilal, A.N. Aldeen, Distributed Grey Wolf Optimizer for scheduling of workflow applications in cloud environments, Applied Soft Computing 102(2021) 107113.

[24]K.W. Liu, Z.C. Cao, Energy-optimized task scheduling of automated warehouse based on improved grey wolf optimizer, Computer Integrated Manufacturing System 26(2)(2020) 376-383.
[25]C. Lu, L. Gao, X.Y. Li, C.Y. Hu, X.S. Yan, W.Y. Gong, Chaotic-based grey wolf optimizer for numerical and engineering optimization problems, Memetic Computing 12(2020) 371-398.
[26]R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Computer-Aided Design 43(3)(2011) 303-315.
[27]S.Y. Liu, A.Z. Jin, A Co-evolutionary Teaching-learning-based Optimization Algorithm for Constrained Optimization Problems, Acta Automatica Sinica 44(9)(2018) 1690-1697.
[28]Q.B. Huang, J. X. Li, C.Y Song, C. H. Xu, X.F. Lin, Whale optimization algorithm based on cosine control factor and polynomial mutation, Journal of Control and Decision 35(3)(2020) 559-568.
[29]G.Q. Zeng, J. Chen, L.M. Li, M.R. Chen, L. Wu, Y.X. Dai, C.W. Zheng, An improved multi-objective population-based extremal optimization algorithm with polynomial mutation, Information Sciences 330(2016) 49-73.
[30]J.S. Liu, Y.N. Mao, X.Z. Liu, Y. Li, A dynamic adaptive firefly algorithm with globally orientation, Mathematics and Computers in Simulation 174(2020) 76-101.
[31]J.S. Liu, X.Z. Liu, Y. Li, Two Subpopulations Cuckoo Search Algorithm Based On Mean Evaluation Method For Function Optimization Problems, International Journal of Pattern Recognition and Artificial Intelligence 34(8)(2020) 2059027.