Grey Wolf algorithm based on S-function and particle swarm optimization

ChenYang Liu¹, Yongli Wang¹∗

¹School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
∗yongliwang@njust.edu.cn

Abstract. Based on the analysis of the shortcomings of the grey wolf optimization algorithm, an improved grey wolf optimization algorithm (SGWO) is proposed. The algorithm uses the convergence factor based on S-function change to balance the global search and local search ability of the algorithm. At the same time, the proportion weight based on Euclidean distance of step size and the individual optimal position of the particle swarm optimization algorithm are introduced to update the grey wolf position, thus speeding up the convergence speed of the algorithm to 8. The simulation results of three classical test functions show that the SGWO algorithm has higher accuracy and better stability.

1. Introduction
Biology brings many inspirations to human beings in different ways, which can help people solve problems in life. Swarm intelligence (swarm intelligence) is a kind of phenomenal law of biological collective behavior in nature. Swarm intelligence (swarm intelligence) algorithm has been widely used in reliability optimization and job scheduling.

In reference [3], the grey wolf optimization (GWO) algorithm is proposed. In this paper, multiple benchmark functions are tested to show that for some benchmark function scenarios, GWO algorithm performs better in convergence speed and search ability. Document [6] uses the idea of population dynamic evolution to improve the global search ability of the algorithm. Inspired by the dynamic evolutionary population and dynamic weight, this paper proposes a hybrid improvement strategy, which combines the two to improve the convergence speed and produce better solutions.

2. Basic Grey Wolf Algorithms
Grey wolf optimization algorithm is a new swarm intelligence optimization algorithm proposed by simulating the social hierarchy mechanism and predation behavior of the grey wolf population in nature. The grey wolf population has a strict hierarchy. As shown in Figure 1.1, α is the highest level of grey wolf, and β is the subordinate of α. δ obeys the orders of α and β. The lowest level of grey wolf is called ω.

![Figure 1.1 Grey Wolf Grade Pyramid](image-url)
Grey wolf hunting is mainly divided into three stages: tracking and approaching prey; hunting and encircling prey until it stops moving; attacking prey.

Assuming that the number of gray wolf population is N, the position of the first gray wolf is \( x_i \), the optimal solution of the population is \( \alpha \), the suboptimal solution is \( \beta \), the third optimal solution is \( \delta \), and the other individuals are \( w \), the mathematical model of gray wolf’s predation behavior is described as follows:

\[
D = C \cdot |X_p(t) - X(t)|
\]

\[
X(t + 1) = X_p(t) - A \cdot D
\]

Among them: \( t \) denotes the number of iterations, \( A \) and \( C \) are coefficient vectors, \( X_p(t) \) is the current position vector of prey, \( X(t) \) is the current position vector of a gray wolf.

The expressions of coefficient vectors \( A \) and \( C \) are:

\[
A = a \cdot (2r_2^2 - 1)
\]

\[
C = 2 \cdot r_1
\]

Among them: \( r_1 \) and \( r_2 \) are random variables of \([0,1]\), \( A \) is the convergence factor, and decreases linearly from 2 linear to 0 with the number of iterations. As shown in formula 5, where Max is the maximum number of iterations

\[
a = 2 - \frac{2t}{\text{max}}
\]

The positions of other wolves in the population are determined by the positions of \( \alpha \), \( \beta \) and \( \delta \).

\[
\begin{align*}
D_\alpha &= [C_1 \cdot X_\alpha - X], \quad X_1 = X_\alpha - A_\alpha \cdot D_\alpha \\
D_\beta &= [C_2 \cdot X_\beta - X], \quad X_2 = X_\beta - A_\beta \cdot D_\beta \\
D_\delta &= [C_3 \cdot X_\delta - X], \quad X_3 = X_\delta - A_\delta \cdot D_\delta \\
X(t+1) &= \frac{X_1(t) + X_2(t) + X_3(t)}{3}
\end{align*}
\]

The main steps of grey wolf optimization algorithm are as follows:

1. Random initialization of wolves and parameters of initialization algorithm in upper and lower bounds of wolves variables
2. Calculate the fitness of each individual wolf.
3. Individual wolves with the highest fitness were selected as alpha, beta and delta wolves.
4. According to formula (1), (2), (3), (5) calculate the position of other wolves.
5. Determine whether the maximum number of iterations has been reached, and stop iteration if it has been reached. At this time, the location of alpha is the location of prey; otherwise, proceed to step (2).

3. The improvement of grey wolf optimization algorithm

3.1. Convergence Factor Based on S-type Function

According to literature [1], when gray wolf attacked prey, when \( |A| > 1 \), it indicated that gray wolf moved away from prey, and expanded the search range to find prey, i.e. global search, with fast convergence. When \( |A| < 1 \), it indicated that gray wolf moved towards prey, contracted the search range to attack prey, i.e. local search, with slow convergence speed. Therefore, the size of \( A \) is closely related to the global and local search ability of GWO algorithm (citation: Grey Wolf optimization algorithm with improved convergence factor and proportional weight). Formula (5) shows that \( A \) changes with the change of convergence factor \( a \), and convergence factor \( a \) decreases from 2 linearity to 0 with the number of iterations, but the algorithm is not linear in the process of continuous convergence. Thus, linear decreasing convergence factor \( a \) can not fully reflect the actual optimization search process.

Therefore, a convergence factor based on S-type function is proposed in this paper.
The expression of the S-type function is as follows:

\[ y = \frac{k}{1 + e^{-ax}} \quad (8) \]

Where \( a \) and \( R \) are constant. Let \( x = \frac{1}{t} \), the expression of the inverted S-type function is obtained as follows:

\[ y = \frac{k}{1 + e^{\frac{a}{t}x}} \quad (9) \]

The image of the S-function is shown in Figure 3.1.

![Figure 3.1: S-type functions](image)

The improved inertia weight formula in this paper is as follows:

\[ a = \alpha_{\text{start}} - \left( \alpha_{\text{start}} - \alpha_{\text{end}} \right) \frac{1}{1 + e^{-c dt}} \quad (10) \]

Among them, \( A \) is the convergence factor, \( \alpha_{\text{start}} \) and \( \alpha_{\text{end}} \) are the initial and final values of the convergence factor. In this paper, \( \alpha_{\text{start}} = 2, \alpha_{\text{end}} = 0 \), \( t \) is the number of iterations, \( c = 3.4 \), \( d = 0.07 \). The change diagram of convergence factor before and after improvement is shown in Figure 3.2.

![Figure 3.2: Convergence Factor Change Graph](image)

The improved strategy can better balance the global search and local search ability of the algorithm.

3.2. Dynamic Weighting Strategy Combined with Particle Swarm Optimization

3.2.1. Dynamic Weighting Strategy
Reference [7] gives a proportional weight based on fitness value, which is expressed as follows:

$$W_a = \frac{f_a}{f_a + f_\beta + f_\delta}, \quad W_\beta = \frac{f_\beta}{f_a + f_\beta + f_\delta}, \quad W_\delta = \frac{f_\delta}{f_a + f_\beta + f_\delta}$$

(11)

$$X(t+1) = X_t \cdot W_a + X_\beta \cdot W_\beta + X_\delta \cdot W_\delta$$

(12)

In reference [8], a proportional weight based on Euclidean distance of step size is proposed. The expression is as follows:

$$W_1 = \frac{|X_1|}{|X_1| + |X_2| + |X_3|}, \quad W_2 = \frac{|X_2|}{|X_1| + |X_2| + |X_3|}, \quad W_3 = \frac{|X_3|}{|X_1| + |X_2| + |X_3|}$$

(13)

$$X(t+1) = \frac{X_1 \cdot W_1 + X_2 \cdot W_2 + X_3 \cdot W_3}{3}$$

(14)

$W_1$, $W_2$, $W_3$ represent the learning rates of $\omega$ wolves to $\alpha$, $\beta$, $\delta$ wolves respectively.

By introducing the above four proportional weights, the convergence speed of the algorithm can be accelerated. Through experiments, it can be found that the effect of introducing the proportional weights proposed in reference [9] is better.

Literature [8] validates the effect of proportional weight based on Euclidean distance of step size from the theoretical point of view.

Therefore, this paper introduces the proportion weight based on Euclidean distance of step size proposed in reference [6].

3.2.2. Combining with Particle Swarm Optimization

Particle Swarm Optimization (PSO), also known as Particle Swarm Optimization (PSO), belongs to evolutionary algorithm, similar to Grey Wolf algorithm. It also starts from random solution and searches for the optimal solution through iteration. Velocity Updating Formula and Location Updating Formula

$$v_{i}^{t+1} = wv_{i}^{t} + c_1r_1(x_{pbi}^{t} - x_{i}^{t}) + c_2r_2(x_{gbi}^{t} - x_{i}^{t})$$

(15)

$$x_{i}^{t+1} = x_{i}^{t} + v_{i}^{t+1}$$

(16)

From the formula, it can be seen that the updating of the position of PSO is mainly through perceiving the global optimal position and the individual optimal position. Particle swarm optimization (PSO) can improve the local search ability by introducing individual optimal location.

The basic Grey Wolf algorithm only perceives the relationship between $\alpha$, $\beta$, $\delta$ wolves and their own location. If combined with particle swarm optimization, the introduction of individual optimal location can better enhance the local search ability of the algorithm.

The grey wolf optimization algorithm formula combining individual optimal position and dynamic weight strategy is as follows:

$$W_1 = \frac{|X_1|}{|X_1| + |X_2| + |X_3| + |X_4|}, \quad W_2 = \frac{|X_2|}{|X_1| + |X_2| + |X_3| + |X_4|}, \quad W_3 = \frac{|X_3|}{|X_1| + |X_2| + |X_3| + |X_4|},$$

$$W_4 = \frac{|X_4|}{|X_1| + |X_2| + |X_3| + |X_4|}$$

(17)

$$X(t+1) = \frac{X_1 \cdot W_1 + X_2 \cdot W_2 + X_3 \cdot W_3 + X_4 \cdot W_4}{3}$$

(18)
\[ \begin{align*}
D_\alpha &= |C_1 \cdot X_\alpha - X|, X_1 = X_\alpha - A_1 \cdot D_\alpha \\
D_\beta &= |C_2 \cdot X_\beta - X|, X_2 = X_\beta - A_2 \cdot D_\beta \\
D_\gamma &= |C_3 \cdot X_\gamma - X|, X_3 = X_\gamma - A_3 \cdot D_\gamma \\
D_\delta &= |C_4 \cdot X_\delta - X|, X_4 = X_\delta - A_4 \cdot D_\delta
\end{align*} \]  \tag{19}

Among them, \( X_p \) is the best position for individuals.

### 3.3. CGWO algorithm steps

Based on the above description of the improvement strategy, the steps of the improved grey wolf optimization algorithm (CGWO) proposed in this paper are given. 

- **Step 1**: Set the population size \( N \), the maximum number of iterations \( t_{\text{max}} \), and randomly generate \( a, A, C \) and other parameters.

- **Step 2**: Initialize gray wolf population randomly in search space;

- **Step 3**: Calculate the fitness values of all gray wolves in the population and rank them according to fitness values. Choose the first three best wolves and record their location \( x_\alpha, x_\beta \) and \( x_\gamma \).

- **Step 4**: Use formulas (5), (8) and (9) to update the position of other gray wolf individuals in the population.

- **Step 5**: Calculate \( a \) by formula (7), then update the values of \( A \) and \( C \) by formula (3) and (4).

- **Step 6**: Determine whether the algorithm satisfies the end condition. If the maximum number of iterations \( t_{\text{max}} \) is reached, the calculation will be stopped and the optimal position \( x_\alpha \) will be output. Otherwise, the Step 3-Step 5 will be executed repeatedly.

### 4. Simulation experiment

In order to verify the performance of CGWO algorithm, eight international classical benchmark functions are selected for simulation experiments. The test functions are shown in Table 1. Among them, \( f_1 \sim f_5 \) is a single peak test function and \( f_6 \sim f_8 \) is a multi-peak test function. The algorithm runs 30 times independently, and the average value of the experiment is obtained to reflect the convergence accuracy of the algorithm under a given number of iterations, and the standard deviation is used to reflect the stability of the algorithm.

#### 4.1. Comparisons with improved GWO algorithm with different strategies

Compared with IGWO algorithm, Square GWO algorithm, NGWO algorithm and CGWO algorithm, CGWO algorithm has higher accuracy and better stability.
5. Conclusion
This paper presents an improved grey Wolf algorithm based on the convergence factor of $S$ function change and the idea of combining dynamic weight strategy with particle swarm optimization. The experimental results of the test function show that the combined algorithm has the advantages of both of them, and can ensure that the search accuracy and the convergence speed of the algorithm are greatly improved, and has better search performance. Next, we will apply the optimized algorithm to practical problems such as mobile cloud resource scheduling.

References
[1] JIANGF, XIAH, TRANQA, et al. A new binary hybrid particle swarm optimization with wavelet mutation [J]. Knowledge-Based Systems, 2017, 130: 90—101.
[2] MIRJALILIS, MIRJALILI SM, LEWISA. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.
[3] Rodríguez L, Castillo O, Soria J, et al. A Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm[J]. Applied Soft Computing, 2017, 57: 315-328.
[4] Wang Min, Tang Mingzhu. A new grey wolf optimization algorithm with non-linear convergence factor [J]. Computer Applied Research, 2016, 33 (12): 3648-3653.
[5] Saremi S, Mirjalili S Z, Mirjalili S M. Evolutionary population dynamics and grey wolf optimizer[J]. Neural Computing & Applications, 2015, 26(5):1257-1263.
[6] Nasrabadi M S, Sharafi Y, Tayari M. A parallel grey wolf optimizer combined with opposition based learning[CI]/Swarm Intelligence and Evolutionary Computation. IEEE, 2016:18-23.
[7] Xu Chenhua, Li Chengxian, Wang Youjun, et al. Aluminum oxide quality index prediction model based on chaotic grey wolf optimization algorithm [J]. Journal of Guangxi University (Natural Science Edition), 2016, 41 (6): 1869-1878.
[8] Guo Zhenzhou, Liu Ran, Gong Changqing, et al. Improvement research based on Grey Wolf algorithm [J]. Computer application research, 2017 (12): 3603-3606.
[9] Wang Qiuping, Wang Mengna and Wang Xiaofeng. Grey Wolf Optimization Algorithms for Improving Convergence Factor and Proportional Weight [J/OL]. Computer Engineering and Application.