Improved Swarm Intelligence Algorithm Based on Novel Nonlinear Multi-Strategy Optimisation

Xiaoran He[1],Jianuo Hou[1],Jianrong Li[1](✉) and Chuanlei Zhang[1](✉)
[1] College of Artificial Intelligence, Tianjin University of Science and Technology Tianjin, China
lisa_ljr@tust.edu.cn

Abstract. Aiming at the problems of low solution accuracy of the grey wolf optimization algorithm and the tendency to fall into local optimality in the later stage, an improved swarm intelligence algorithm (NSGWO) based on a novel nonlinear multi-strategy optimization is proposed : firstly, Logistic mapping is utilized to make the initial particles as uniformly distributed as possible, so as to provide a balanced search space; improved Gaussian distributions are used to adjust the convergence factor, and a positional The hyperbolic tangent function is introduced into the update formula. Thus, the search efficiency and accuracy of the algorithm are improved with the help of the nonlinear property of the logarithmic function. Finally, the optimal position perturbation mechanism is introduced, which prompts the algorithm to jump out of the local optimal solution by perturbing the optimal position in a small range, thus further improving the optimisation performance of the algorithm. Several classical unimodal and multimodal test functions are used to verify the optimisation performance of NSGWO. The experimental results show that compared with other classical optimisation algorithms, NSGWO has a greater advantage in terms of convergence speed and solution accuracy.

Keywords: Logistic mapping, convergence factor, position update, optimal position perturbation,GWO.

1 Introductory

Swarm intelligence algorithms are a class of optimization algorithms inspired by the behaviour of a group of people, aiming to find the optimal solution to a complex optimization problem. Among them Grey Wolf Optimizer , GWO, is a heuristic optimisation algorithm, proposed by Mirjalili et al.[1] in 2014.

It has been widely used in various aspects such as parameter optimisation[2]. In addition, the GWO algorithm supports the collaboration of multiple intelligences and is able to search the problem solution space in a wider range.

Since its proposal, the grey wolf optimisation algorithm has attracted much attention and research. Some studies have refined the algorithm by improving the position update strategy in the grey wolf optimization algorithm.Hou[3] used a dynamic proportional weighting strategy to update the positions of grey wolves, which accelerated the convergence of the algorithm.Pan[4] proposed a competitive guiding strategy to update the positions of the individuals, which made the algorithm's search more flexible.Liu[5] introduced an evolutionary boundary constraint processing mechanism The position of grey wolves is updated in time to retain the position information of the optimal individual, which enhances the search accuracy of the algorithm.Yao[6] optimized the position update equation in the original grey wolf optimization algorithm, and introduced the concept of information entropy to maintain the diversity of the population in the iterative process to avoid the algorithm falling into a local optimum.

There are also some studies to improve the performance of the algorithm by improving the convergence strategy in the grey wolf optimization algorithm. Li et al.[7] introduced the cosine control factor to balance the global and local exploration ability of the algorithm and improve the convergence speed.Song et al.[8] improved the global search ability of the grey wolf optimization algorithm by adjusting the linear convergence factor to nonlinear.Jia et al.[9] changed the method of decreasing the convergence factor of grey wolf optimization algorithm by adopting the nonlinear decreasing method, which improves the search ability of the algorithm.Zhang et al.[10] proposed an adaptive convergence factor adjustment strategy and adaptive weighting factor to update individual positions, which improves the convergence accuracy, speed and stability of the algorithm.Zhou et al.[11] proposed a method to optimize the initial population distribution by using a chaotic mapping, and at the same time, combined with a random walk algorithm, which enhanced the local search capability of the traditional grey wolf optimization algorithm.

In addition, some scholars have introduced other algorithmic ideas to enhance the grey wolf optimization algorithm.Du and Zhang[12] used a fuzzy C-mean clustering algorithm to group wolves based on the traditional grey wolf optimization algorithm to increase the diversity of the population, and used the Harris Hawk optimization algorithm to design the optimal escape position of the prey, which reduces the probability of the local optimum, and used the Particle Swarm Optimization algorithm and the Bat optimization algorithm to design the movement patterns of different positions. to design movement patterns at different locations, increasing the ability to find the global optimum.Lghamrawy[13] combined the genetic algorithm with the grey wolf optimisation algorithm, using genetic crossover and mutation operations to speed up the exploration process of the algorithm and improve the efficiency of the algorithm.Liu[14] integrated the lion optimisation algorithm and the dynamic weights into the original grey wolf algorithm, improving the searching ability of the algorithm, and at the same time avoiding falling into local optimum.Liu[15] combined the grey wolf optimization algorithm with differential evolutionary algorithm and OTSU algorithm, and the accuracy and stability of the improved algorithm were improved.Liang et al.[16] also combined the differential evolutionary algorithm and grey wolf optimization algorithm, which improved the searching ability of the algorithm.Zhao et al.[17] improved the grey wolf optimization algorithm's hunting formula to increase the diversity of the population through the interference of other individuals, which improved the convergence, robustness and efficiency of the improved algorithm in solving complex problems.Wang[18] introduced differential evolutionary algorithm to guide the evolution of the wolf population, which improved the search ability of the algorithm, updated the wolf population according to the principle of survival of the fittest, and avoided the algorithm from falling into a local optimum.Yang[19] proposed to combine grey wolf optimization algorithm with the dynamic weighting algorithm combined to improve the convergence accuracy and convergence speed of the algorithm.

In order to further optimise the GWO algorithm, this paper is dedicated to improving the global search capability and performance optimisation of the algorithm by targeting key aspects such as algorithm initialisation, convergence factor, position update and optimal position perturbation.

2 Related work

2.1 Grey Wolf Optimisation Algorithm

GWO[1] simulates grey wolf group predation behaviour and achieves optimisation based on the mechanism of wolf group collaboration.The basic principle of the grey wolf optimisation algorithm is as follows(see Fig. 1).

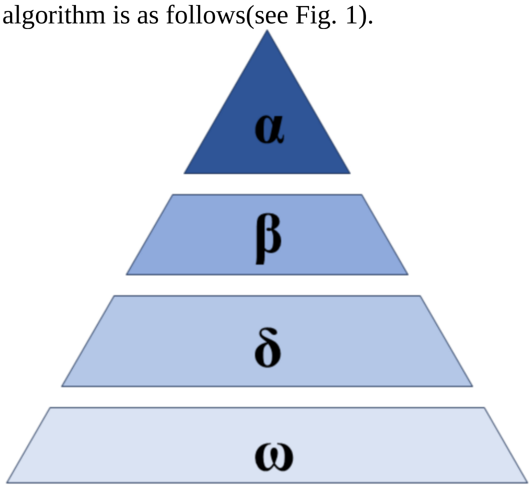

Fig. 1. Wolf pack rank distribution map

In GWO, grey wolves use the following position update formula to achieve prey encirclement during hunting:

$$\vec{D}=\left|\vec{C}*\vec{X}_p(t)-\vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1)=\vec{X}_p(t)-\vec{A}\vec{D} \tag{2}$$

Eq. (1) is the distance between the grey wolf and the prey, Eq. (2) is the position update formula of the grey wolf, and are the position vectors of the prey and the grey wolf, respectively, and t is the current iteration number. Sum are the determined coefficients, which are calculated by the formulas, respectively:

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

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

where A, C are vectors of random numbers with two one-dimensional components taking values within [0,1], which are used to model the attack behaviour of a grey wolf on its prey, which is affected by the values it takes.

During the evolutionary computation, the location of the prey (optimal solution) is unknown, so in GWO this paper considers the optimal grey wolf as α, the second best as β, the third best as δ, and the rest of the grey wolves as ω. The model is built based on the property that α (the potential optimal solution), β, and δ have more knowledge about the location of the prey, and the iterative process employs α , β, and δ to direct the movement of ω to achieve global optimisation . Using the positions of α, β and δ, the positions of all grey wolves are updated using the following equation:

$$\vec{D}_\alpha = \left|\vec{C}_1 * \vec{X}_\alpha - \vec{X}\right| \tag{5}$$

$$\vec{D}_\beta = \left|\vec{C}_2 * \vec{X}_\beta - \vec{X}\right| \tag{6}$$

$$\vec{D}_\sigma = \left|\vec{C}_3 * \vec{X}_\sigma - \vec{X}\right| \tag{7}$$

The above represents the distance of individual grey wolves from tier packs, tier packs, and tier packs, respectively.

$$\vec{X}_1 = \left|\vec{X}_\alpha - A_1 * \vec{D}_\alpha\right| \tag{8}$$

$$\vec{X}_2 = \left|\vec{X}_\beta - A_2 * \vec{D}_\beta\right| \tag{9}$$

$$\vec{X}_3 = \left|\vec{X}_\sigma - A_3 * \vec{D}_\sigma\right| \tag{10}$$

Where X_1, X_2 and X_3 denote the positions of individual X grey wolves that need to be adjusted due to the influence of α, β and δ wolves, respectively, and the average value is taken here(see Fig. 2).

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{11}$$

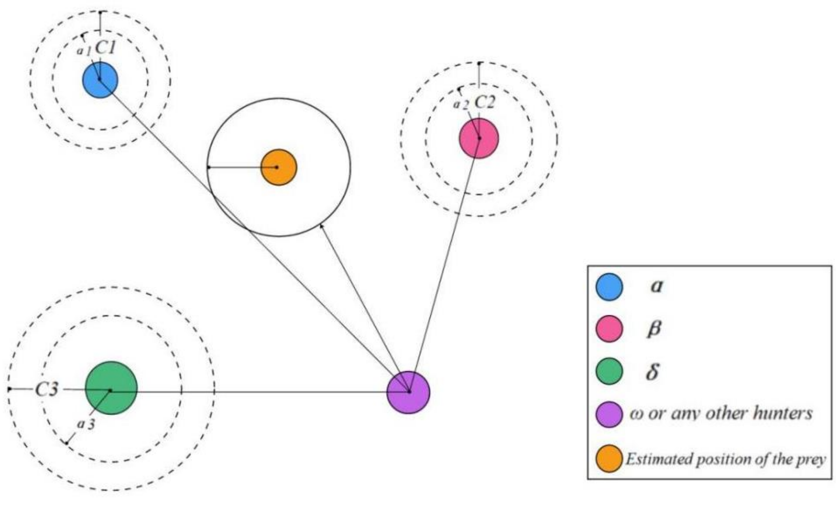

Fig. 2. Grey Wolf Optimisation Algorithm Location Update Chart

In the following equation, t denotes the current number of iterations and T is the set maximum number of iterations. a larger value of a causes the grey wolves to move away from the prey in the hope of finding a more suitable prey, thus prompting the pack to perform a global search, and if a smaller value of a causes the grey wolves to move closer to the prey, prompting the pack to perform a local search.

$$a = 2 - 2 * t/T \tag{12}$$

2.2 Normal Distribution

The normal distribution, also known as Gaussian distribution[20], can be used to generate standard normally distributed random numbers using the function transformation method. If the random variable X obeys a Gaussian distribution with mathematical expectation μ and standard variance σ2 , denoted as:X~N(μ,σ^2), then its probability density function is.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{13}$$

The normal distribution is a strictly stable probability distribution(see Fig. 3).

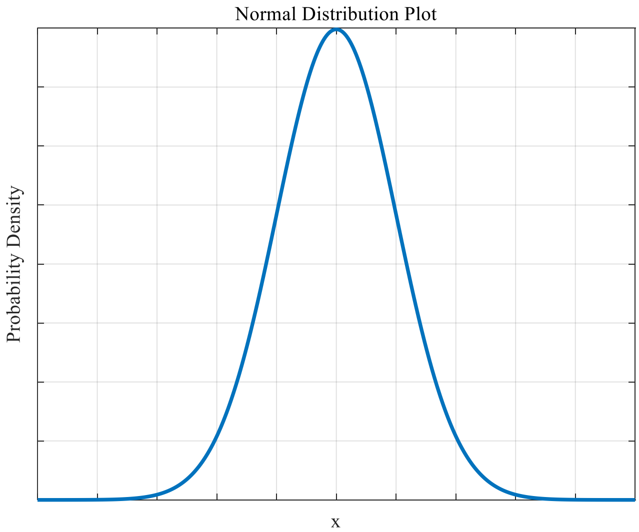

Fig. 3. normal distribution chart

3 Algorithmic Model

3.1 Initialisation

The first improvement is related to initialisation.Logistic mapping is a relatively simple one-dimensional chaotic system. Chaos is an inherent property of nonlinear dynamical systems and is a common phenomenon in nonlinear systems.Various forms of definitions of Logistic mappings exist, one form of definition is shown below[21]:

$$X_{n+1} = X_n \mu (1 - X_n), \mu \in [0, 4], X_n \in (0, 1) \tag{14}$$

Logistic mappings are not necessarily in a chaotic state, which is related to the value of μ. Only when 3.5699456 < μ ≤ 4[22], Logistic mapping has chaotic nature. When μ is taken as 4, better results can be obtained, so in this paper, μ = 4 is taken(see Fig. 4).

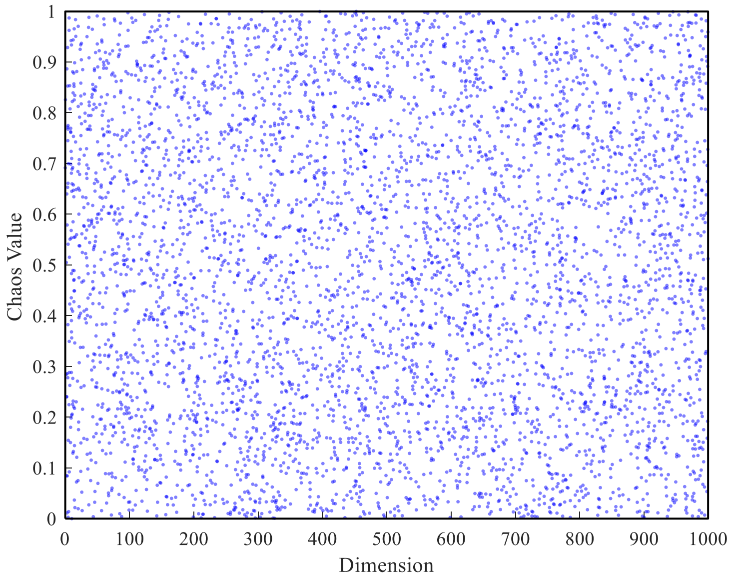

Fig. 4. Initialise the particle distribution map

3.2 Convergence factor

The second improvement is related to the convergence factor. The convergence factor obeys a normal distribution , which will improve the global search capability of the algorithm. In this paper, we refer to[23] for the analysis of the convergence factor:

$$a(k) = (a_{\max} - a_{\min}) \exp\left[- \frac{k^2}{(\alpha * t_{\max})^2} \right] + a_{\min} \tag{15}$$

Equation (15) is based on the exponential function, but it may have limitations in some cases. In order to further improve the convergence performance of the algorithm, a hyperbolic tangent function with saturation property is introduced in this paper. It helps to find the global optimal solution faster. Its defining equation is as follows.

$$a(k) = (a_{\max} - a_{\min}) \cdot \tanh\left(\frac{-k^3}{a \cdot t_{\max}} \right) + a_{\min} \tag{16}$$

Where α is the expansion constant for changing the curve and k is the current number of iterations. As shown in the figure, the curve of a obeys a normal distribution(see Fig. 5).

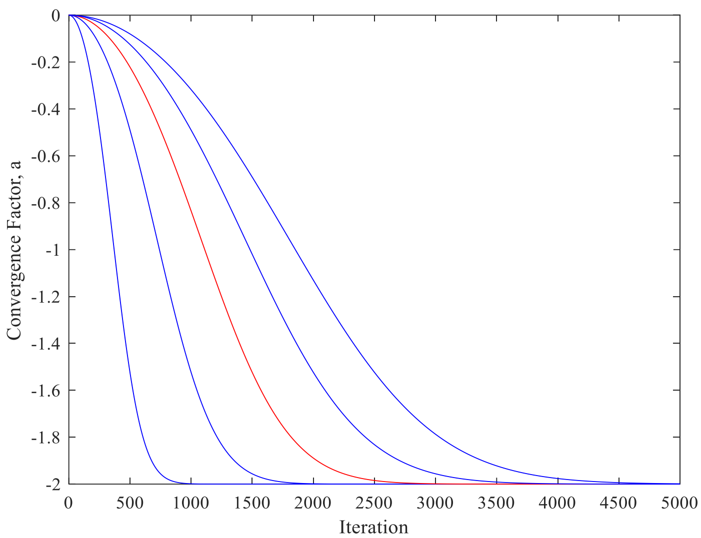

Fig. 5. Convergence factor convergence curve

Figure 3 clearly shows that the decreasing rate decreases rapidly in the first 1000 iterations, but the decreasing rate slows down after more than 1000 iterations. This tendency can quickly approach the optimal solution at the beginning of the iteration, so its value range is set to [0.1, 0.5] to make full use of this property.

3.3 Location Updates

X_1, X_2, and X_3 denote the positions of individual ω grey wolves that need to be adjusted due to the influence of α, β, and δ wolves, respectively. Using the positions of α, βand δ, the positions of all grey wolves were updated using the following equation:

$$A = 2 \cdot a \cdot rand() - a \qquad (16\ 7)$$

$$C = 2 \cdot rand() \qquad (18)$$

Where a is the convergence factor and the A and C coefficients are generated according to a, rand ().

$$m_1 = \log_3 (f_i)^2 \qquad (19)$$

$$m_2 = \log_6 (f_i)^2 \qquad (17\ 0)$$

$$m_3 = \log_9 (f_i)^2 \qquad (21)$$

$$J_i = \frac{m_i}{\sum_{i=1}^{3} m_i} \qquad (22)$$

Where J_1 is the coefficient of contraction for α-wolves, J_2 is the coefficient of contraction for β-wolves, and J_3 is the coefficient of contraction for δ-wolves.

$$\vec{X}_1 = J_1 \cdot \vec{X} - A_1 \cdot \vec{D}_\alpha \qquad (2$$

18)

$$\vec{X}_2 = J_2 \cdot \vec{X} - A_2 \cdot \vec{D}_\beta$$ (2
19)

$$\vec{X}_3 = J_3 \cdot \vec{X} - A_3 \cdot \vec{D}_\delta$$ (2
20)

Where X_1, X_2 and X_3 denote the positions of individual grey wolves that need to be adjusted due to the influence of α wolves, β wolves and δ wolves, respectively.

3.4 Optimal position perturbation

By referring to[24] for the analysis of the optimal position perturbation, the

$$U(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 \times |X_{\text{rand}}(t) - 2 \times r_2 \times X(t)|, r_5 \geq 0.5 \\ X_\alpha(t) - X_{\text{avg}}(t) - r_3 \times (lb + r_4 \times (ub - lb)), r_5 < 0.5 \end{cases} \tag{26}$$

In this paper, more stochastic parameters and dynamic adjustments are introduced to provide more flexible location update strategies.

$$R(t+1) = \begin{cases} X_{\text{rand}(t)} - r_1 \times |X_{\text{rand}(t)} - 2r_2 \times X(t)| + \frac{1}{2} \times (ub - lb) \times r_3 & \text{if } r_s \geq 0.5 \\ x_a(t) - X_{\text{ang}(t)} - \frac{1}{3} \times (lb + r_4 \times (ub - lb)) + r_s \times (ub - lb) & \text{if } r_s < 0.5 \end{cases} \tag{27}$$

Where R(t + 1) denotes the updated position vector, and t denotes the current iteration number. xrand (t) denotes the position vector of a randomly selected grey wolf, and Xrand (t) is the position of a randomly selected individual from the population in the current iteration. r_1, r_2, r_3, r_4, and rs denote the random parameters, which take the values in the range of [0,1], and these parameters are used to control the position magnitude and direction of the update.

The importance of the perturbed grey wolf position being better than the original target position cannot be overstated. To ensure this, this paper introduces a greedy strategy after the update operation. This strategy simply compares the fitness values of the new and old target positions and decides whether to update the target position based on the comparison. Doing so ensures that the search process is always moving in a more favourable direction.

$$X(t+1) = \begin{cases} R(t+1), f(R(t+1) \leq f(X(t))) \\ X(t), \text{ otherwise} \end{cases} \tag{28}$$

The variation of the optimal position perturbation can help the algorithm to better explore the search space and increase the search diversity of the algorithm, thus improving the global search capability of the algorithm.

Combining the above analyses, this paper proposes the following pseudo-code flow chart for NSGWO:

Input: N,pop, dim, maxIter
Output: Best_fitness,Best_index,Best_Pos
Begin
  Initializing (Using Logistic mapping to distribute N wolves uniformly in the search space).
For i = 1 to pop

```
        Calculating their fitness and Find Xα Xβ and Xδ.
    End for
    For t =1 to maxIter
      For i= 1 to pop
          For j=1 to dim
              Computing a; by using Eq.(16).
          Updating X; by Eq(17)-(28).
        End for
        For z = 1 to pop
          Computing U;by using Eq. (30).
            Comparing Position and U and Selecting Best_Pos;by using
              Eq.(31).
          Updating Best_Pos.
        End for
      End for
    End for
    Return the Best_fitness,best_index,Best_Pos.
    End
```

4 Numerical Experiments And Analysis Of Results

4.1 Test Functions and Parameter Settings

In order to verify the effectiveness of the NSGWO algorithm proposed in this paper, 6 basic test functions[1] are selected to verify the proposed improved algorithm. The parameter settings of the test functions are shown in Tables 1, where F1-F6 are unimodal functions .

Table 1. 6 unimodal test functions.

Function	Dim	Range	fmin
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]	0
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$f_4(x)=\max_i\{\lvert x_i\rvert, 1\leqslant i\leqslant n\}$	30	[-10,10]	0
$f_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
$f_6(x)=\sum_{i=1}^{n} i x_i^4+\mathrm{random}[0,1)$	30	[-1.28,1.28]	0

Images of the objective functions F1-F6(see Fig. 6).

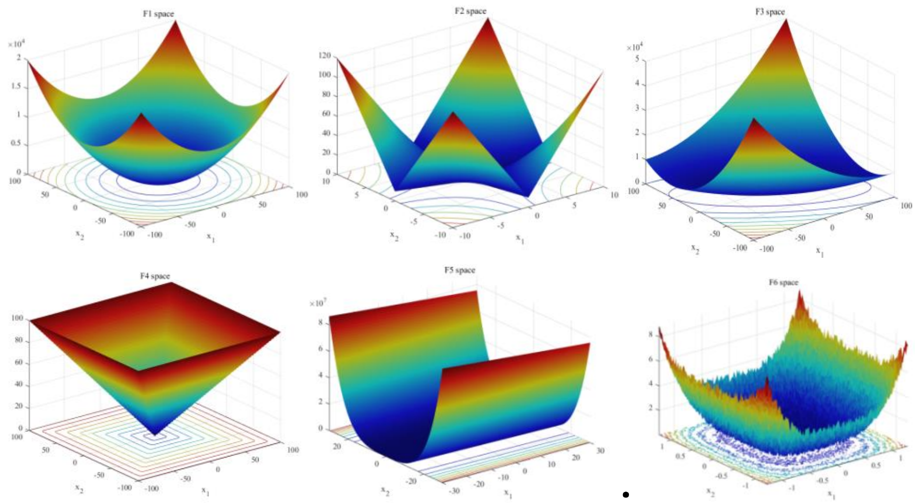

Fig. 6. Images of the objective functions F1-F6

4.2 Average fitness values of the NSGWO algorithm at time for different values of α

In this paper, the effect of parameter α on the performance of the algorithm is contemplated through the study of the convergence factor formulation in the improved grey wolf optimisation algorithm (see equation (16)). By using different values for the parameter α, differences in the performance of the improved grey wolf optimisation algorithm were observed. In order to determine the optimal value of the parameter α, this paper chooses the six standard test functions in Table 1 for multiple tests and averages the fitness values obtained from each run of the algorithm as a comparison result. The specific test results are shown in Table 2.

Table 2. 6 Different α in the test function run results.

α	0.1	0.2	0.3	0.4	0.5
F1	-91.413489045782775	**-93.589560243293500**	-89.338516195241610	-85.788886664829860	-82.223654290345370
F2	-52.799616970950375	**-62.326178091277440**	-58.088011755190730	-52.524711143772880	-57.465046590268930
F3	-90.524503450978690	**-1.007885732395940e+02**	-95.837703372384280	-91.425620641138170	-94.664706760062490

F4	-37.149154635156208	**-52.517308833937420**	-50.323843121710880	-48.023657638930670	-45.709004145301556
F5	1.461890748295926	1.461679566873616	1.461775138716486	**1.461467760724498**	1.461938351920743
F6	**-2.235534749657929**	-1.431952671203699	-1.720423689016609	-2.030622054344956	-1.713682013933701

From the experimental results in Table 2, it can be seen that when the parameter α takes different values, the improved algorithm has different mean values for the 6 objective functions adaptation. According to the data in the table, the parameter α = 0.2 is selected for the improved algorithm in order to optimise the overall performance of the algorithm.

4.3 Comparison with related improved algorithms

In order to verify the effectiveness of this paper's algorithm for test functions of different modes, six unimodal test functions in Table 1 are used to test RUN[25], DBO[26], TSA[27], DE[28], GOA[29], and GWO[1] with this paper's improved algorithm (NSGWO), respectively, and each algorithm is tested with 150 iterations and run for 200 runs, respectively, to take the average of the adaptation values for comparison. The average of the fitness values reflects the solution accuracy of the algorithms, and the specific experimental results are listed in Table 3, where the best results are blackened.

Table 3. Comparison of test results of different intelligent algorithms for unimodal functions.

function	RUN	DBO	TSA	DE	GOA	GWO	NSGWO
F1	-56.718087260211490	-43.494733613369690	-3.597690684770630	-1.685583508400887	3.592793856535685	-7.109741080478434	**-1.103779030586058e+02**
F2	-31.998081146595300	-21.488018491907390	-2.636960656984202	-0.120509019693695	18.989935714170390	-4.734705982672060	**-55.930217171594950**
F3	-50.716688846070580	-24.468539030585134	0.102332179479164	0.395989658166135	3.714840244358193	0.893800708157472	**-1.006478237955283e+02**
F4	-35.324046066527150	-17.899608274900660	-0.350074033847738	-0.532617891412471	1.419858771940058	-2.174297892682040	**-53.741814395557150**

F5	**1.393035082909629**	1.428800338018144	1.464183380784476	1.693356548616003	8.162911781920434	1.459713043551130	1.461668236099607
F6	**-2.834194405575526**	-2.141282776312391	-1.612582698494702	-1.070873840539553	7.731335536150558	-2.096183514989884	-2.135558961183047

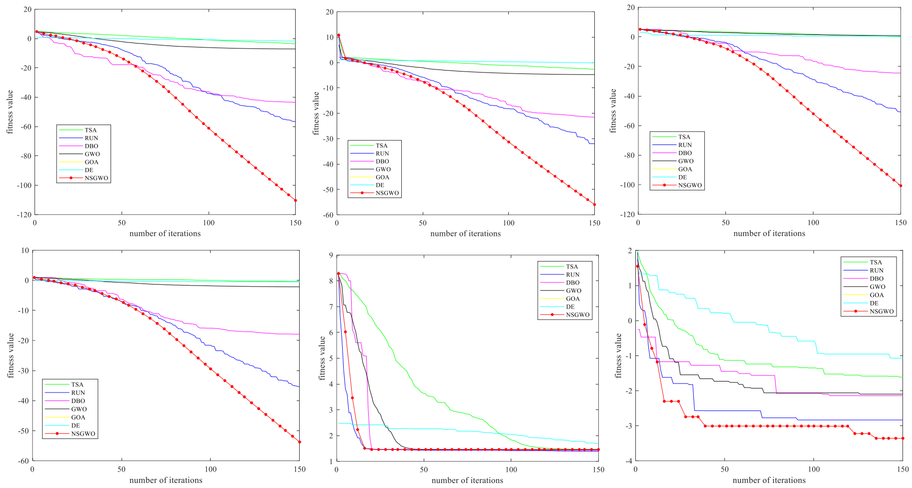

Fig. 7. Evolutionary curves of fitness averages for 7 algorithms on 6 test functions

A comparison of the data in Table 3 and Figure 7 shows that compared with the other five swarm intelligence algorithms and the unimproved Grey Wolf optimization algorithm, the NSGWO algorithm has the smallest fitness value on the state functions F1-F4; on the functions F5 and F6, NSGWO's fitness is slightly worse than that of RUN, but the convergence speed of NSGWO is much higher than that of the other algorithms. In addition, according to the graphical data, NSGWO is much higher than the unimproved GWO algorithm, both in terms of adaptation for the test functions and in terms of convergence speed. Taken together, this shows that the performance aspect of the improved NSGWO algorithm has been greatly improved, and its stability and solution accuracy are higher than the other six algorithms.

5 Summaries

In this paper, through a series of comparative experiments and practical application tests, the significant effect of NSGWO in enhancing the performance of traditional GWO is fully verified. In particular, NSGWO shows obvious advantages when dealing with high-dimensional, nonlinear, nonconvex and other complex optimisation problems.NSGWO is able to show better robustness and adaptivity when dealing with different types of problems, demonstrating its excellent global search capability and fast convergence characteristics. In view of the above research results, our future work plan will focus on the following directions:

1) Further application of NSGWO in the field of smart grids, especially in power system load balancing and optimal scheduling of distributed generation systems to enhance energy efficiency and reduce waste.

2) Explore the application of NSGWO in the field of environmental sciences to optimise environmental models, such as the prediction of air quality models and the management and allocation of water resources.

Overall, this paper effectively improves the grey wolf optimization algorithm and significantly enhances its ability and efficiency in solving complex problems, while the research in this paper provides new perspectives and methods for the further development of the GWO algorithm in practical applications. In the future, we will continue to study other potential advantages and possibilities of the Grey Wolf Optimization Algorithm in depth, so as to play a powerful role in more fields and promote its application and development in practical problem solving.

References

1. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software 69, 46–61 (2014).
2. Ebrahimi, S.M., Hasanzadeh, S., Khatibi, S.: Parameter identification of fuel cell using Repairable Grey Wolf Optimization algorithm. Applied Soft Computing 147, 110791 (2023).
3. Hou, Y., Gao, H., Wang, Z., Du, C.: Improved Grey Wolf Optimization Algorithm and Application. Sensors 22, 3810 (2022).
4. Pan, H., Chen, S., Xiong, H.: A high-dimensional feature selection method based on modified Gray Wolf Optimization. Applied Soft Computing 135, 110031 (2023).
5. Liu, X., Li, G., Yang, H., Zhang, N., Wang, L., Shao, P.: Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm. Expert Systems with Applications 233, 120946 (2023).
6. Yao, K., Sun, J., Chen, C., Cao, Y., Xu, M., Zhou, X., et al.: An information entropy-based grey wolf optimizer. Soft Computing 27(8), 4669–84 (2023).
7. Li, Y., Lin, X., Liu, J.: An Improved Gray Wolf Optimization Algorithm to Solve Engineering Problems. Sustainability 13, 3208 (2021).
8. Song, C., Wang, X., Liu, Z., Chen, H.: Evaluation of axis straightness error of shaft and hole parts based on improved grey wolf optimization algorithm. Measurement 188, 110396 (2022).
9. Jia, S., Sun, S., Wang, J., Xu, F.: Research Based on The Improved Grey Wolf Optimization Algorithm. Journal of Physics: Conference Series 1883(1), 012036 (2021).
10. Zhang, W., Zhang, S., Wu, F., Wang, Y.: Path Planning of UAV Based on Improved Adaptive Grey Wolf Optimization Algorithm. IEEE Access 9, 89400–89411 (2021).
11. Zhou, H., Wu, X., Li, Y., Fan, Z., Chen, W., Mao, J., et al.: Model optimization of a high-power commercial PEMFC system via an improved grey wolf optimization method. Fuel 357, 129589 (2024).

12. Du, J., Zhang, Z., Li, M., et al.: RETRACTED ARTICLE: Optimal scheduling of integrated energy system based on improved grey wolf optimization algorithm. Sci Rep 12, 7095 (2022).
13. Ighamrawy, S.M., Hassanien, A.E.: A hybrid Genetic–Grey Wolf Optimization algorithm for optimizing Takagi–Sugeno–Kang fuzzy systems. Neural Comput & Applic 34, 17051–17069 (2022).
14. Liu J, Wei X, Huang H: An Improved Grey Wolf Optimization Algorithm and its Application in Path Planning. IEEE Access 9, 121944-121956 (2021), doi: 10.1109/ACCESS.2021.3108973
15. Liu, Y., Sun, J., Yu, H., Wang, Y., Zhou, X.: An Improved Grey Wolf Optimizer Based on Differential Evolution and OTSU Algorithm. Appl. Sci. 10, 6343 (2020).
16. Liang, Y., Hu, S., Guo, W., Tang, H.: Abrasive tool wear prediction based on an improved hybrid difference grey wolf algorithm for optimizing SVM. Measurement 187, 110247 (2022).
17. Zhao, Yt., Li, Wg., Liu, A.: Improved grey wolf optimization based on the two-stage search of hybrid CMA-ES. Soft Comput 24, 1097–1115 (2020).
18. Wang, J-S., Li, S-X.: An Improved Grey Wolf Optimizer Based on Differential Evolution and Elimination Mechanism. Scientific Reports 9(1), 7181 (2019).
19. Yang, Q., Liu, J., Wu, Z., He, S.: A fusion algorithm based on whale and grey wolf optimization algorithm for solving real-world optimization problems. Applied Soft Computing 146, 110701 (2023).
20. Liu, Y., Jiang, Y., Zhang, X., Pan, Y., Qi, Y.: Combined Grey Wolf Optimizer Algorithm and Corrected Gaussian Diffusion Model in Source Term Estimation. Processes 10(7), 1238 (2022).
21. Zhao, K., Liu, Y., Hu, K.: Optimal Pattern Synthesis of Linear Array Antennas Using the Nonlinear Chaotic Grey Wolf Algorithm. Electronics 12(19), 4087 (2023).
22. Ye, G., Jiao, K., Huang, X., Goi, B-M., Yap, W-S.: An image encryption scheme based on public key cryptosystem and quantum logistic map. Scientific Reports 10(1), 21044 (2020).
23. Shi, C., Dou, Z., Sangaiah, A.K., et al.: Cognitive decision engine based on binary particles swarm optimization with non-linear decreasing inertia weight. Concurrency and Computation: Practice and Experience 33(12), e4975 (2021).
24. Li, K., Li, S., Huang, Z., et al.: Grey Wolf Optimization algorithm based on Cauchy-Gaussian mutation and improved search strategy. Scientific Reports 12(1), 18961 (2022).
25. Li, M., Liu, Z., Song, H.: An improved algorithm optimization algorithm based on RungeKutta and golden sine strategy. Expert Systems with Applications 247, 123262 (2024).
26. Xue, J., Shen, B.: Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. The Journal of Supercomputing 79(7), 7305–36 (2023).
27. Kiran, M.S.: TSA: Tree-seed algorithm for continuous optimization. Expert Systems with Applications 42(19), 6686–98 (2015).
28. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–59 (1997).