Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ31353

An Improved Grey Wolf Optimizer Based on Elite Opposition-Based Learning Strategy and Non-Linear Parameters

Jiayi Miu

Department of Information Science and Technology, Nanjing Forestry University, Nanjing, China
18015515826@163.com

Abstract. The Grey Wolf Optimizer (GWO) has been extensively applied in meta-heuristic optimization. However, it inherently suffers from several limitations, including inaccurate solution outputs, slow convergence speed, and a high tendency to get trapped in local optima. To resolve these problems, this study introduces two targeted improvements to the GWO algorithm. Firstly, an elite opposition-based learning method is employed for initializing the grey wolf population. This method enhances the diversity of initial individuals, reinforces the algorithm's global search ability, and accelerates convergence in the early iteration stage. Secondly, nonlinear parameters are incorporated into both the prey encircling and attacking processes of GWO. This modification expands the algorithm's search scope during the early iterations. Ten benchmark test functions with distinct characteristics were used to validate the improved algorithm (IEN-GWO), which was compared with five well-recognized meta-heuristic algorithms. The experimental results demonstrate that IEN-GWO outperforms the compared algorithms in terms of solution precision, stability, and convergence rate.

Keywords: Grey Wolf Optimizer, Elite Opposition-Based Learning Strategy, Non-Linear Parameters, Function Testing

1. Introduction

Swarm intelligence algorithms are an important part of meta-heuristic methods. They get inspiration from the cooperative behaviors of natural groups. People use them to find global optimal solutions. Common examples include Genetic Algorithm (GA) [1], Particle Swarm Optimization (PSO) [2], Whale Optimization Algorithm (WOA) [3], and Harris Hawks Optimization (HHO) [4].

Mirjalili et al. proposed the Grey Wolf Optimizer (GWO) [5] in 2014. GWO imitates the social hierarchy of grey wolves. It also imitates their cooperative hunting behavior. This algorithm is widely used. The reasons are its good convergence, simple parameter settings, and easy implementation. But in practical use, GWO often has problems. These problems include: the solution results are not accurate enough, convergence is slow, and it is easy to get stuck in local optima. Researchers have developed many improved versions to solve these problems. Liu et al. used Chebyshev chaotic mapping to initialize the population. They also used nonlinear strategy adjustments. These adjustments control parameters and position update equations. This method balances exploration and exploitation. It also improves convergence [6].Sun et al. added dynamic

DOI: 10.54254/2755-2721/2026.TJ31353

perturbation coefficients to position updates. They combined improved convergence factors with reverse learning selection. But the solution accuracy of this method was still not good enough [7]. Banaie-Dezfouli et al. put forward a representativeness mechanism. This mechanism guides position updates. It achieves dynamic balance between population diversity and the coordination of exploration and exploitation [8]. Seyedabbasi et al. integrated group coordination and chained following mechanisms to accelerate global search, but this increased dependence on alpha wolves [9]. Meidani et al. developed AGWO through adaptive parameter tuning and termination criteria optimization, effectively boosting global optimization performance [10].

Despite these advancements, existing improved GWO algorithms still have room for optimization in core areas: adaptive balance between local exploration and global search, convergence efficiency, and escaping local optima. This study proposes the Elite Opposition-Based Learning and Nonlinear Parameter-Enhanced Grey Wolf Optimizer (IEN-GWO) and validates it using 10 benchmark test functions.

2. Basic Grey Wolf Optimization algorithm

In 2014, researcher S. Mirjalili proposed GWO after in-depth studies on grey wolves' hunting patterns and social structures. In nature, grey wolves have a strict social hierarchy and follow an orderly process of tracking, encircling, and attacking prey. This collaborative behavior forms the biological basis of GWO.

(1) Surrounding Prey. When hunting, grey wolves gradually encircle their prey. In GWO, this behavior is modeled as the convergence of wolves toward the prey, described by the following mathematical equations:

$$d = |C \cdot x_p(t) - x(t)| \tag{1}$$

$$x(t+1) = x_p(t) - A \cdot d \tag{2}$$

Where: t is current iteration number, $x_p(t)$ is prey's position at iteration t, $x(t)$ is grey wolf's current position, d is Euclidean distance between the grey wolf and prey, $x(t+1)$ is grey wolf's updated position at iteration $t+1$, A and C are coefficient vectors, calculated as:

$$A = 2k \cdot s_1 - k, C = 2 \cdot s_2 \tag{3}$$

Here, k linearly decreases from 2 to 0 as iterations progress, and s_1 and s_2 are random vectors within [0, 1].

(2) Leadership and Position Update. Grey wolf hunting is led by alpha (α) wolves, with occasional assistance from beta (β) and delta (δ) wolves. In the search space, the prey's exact position is unknown. Thus, GWO assumes that α, β, and δ wolves have the best knowledge of potential prey positions. The algorithm retains the top three optimal solutions and guides other wolves to update their positions based on these leaders:

$$\begin{cases} d_\alpha = |C_1 \cdot x_\alpha - x| \\ d_\beta = |C_2 \cdot x_\beta - x| \\ d_\delta = |C_3 \cdot x_\delta - x| \end{cases} \rightarrow \begin{cases} x_1 = x_\alpha - A_1 \cdot d_\alpha \\ x_2 = x_\beta - A_2 \cdot d_\beta \\ x_3 = x_\delta - A_3 \cdot d_\delta \end{cases} \rightarrow x\left(t+1\right) = \frac{x_1+x_2+x_3}{3} \tag{4}$$

DOI: 10.54254/2755-2721/2026.TJ31353

Where: A_1, A_2, A_3 and C_1, C_2, C_3 are coefficient vectors for α, β, δ wolves; $d_\alpha, d_\beta, d_\delta$ are distances between ordinary wolves and the three leaders; x_1, x_2, x_3 are candidate positions guided by α, β, δ wolves; $x(t+1)$ = final updated position.

(3) Grey wolves confirm the prey's position through attacks, obtaining the optimal solution. During iterations, the gradual decrease of A simulates the wolf pack's approach to prey. The value of A is controlled by parameter a:

$$a = 2 - \frac{2t}{T} \tag{5}$$

Where T = maximum number of iterations. As t increases from 0 to T, a linearly decreases from 2 to 0, and A follows the same trend.

3. Improved Grey Wolf Optimization algorithm

3.1. Population initialization via elite opposition-based learning strategy

The initial population of the standard GWO algorithm is generated randomly. This random generation leads to uneven distribution and poor diversity of the population. We introduce an elite opposition-based learning strategy to fix this problem.

This strategy uses two key features. First, reverse solutions are often closer to the global optimal solution. Second, elite individuals hold high-quality information. For an elite individual $x_e = [x_{e,1}, x_{e,2}, \ldots, x_{e,D}]$ (where D stands for the number of dimensions), its opposition solution is defined as:

$$x_{opp,i} = LB_i + UB_i - x_{e,i} \tag{6}$$

Here, LB_i and UB_i are the lower and upper bounds of the i-th dimension. If $x_{opp,i}$ goes beyond the boundary (either $x_{opp,i} < LB_i$ or $x_{opp,i} > UB_i$), we reset it randomly with this formula:

$$x_{opp,i} = LB_i + rand() \cdot (UB_i - LB_i) \tag{7}$$

We generate opposition solutions for all elite individuals first. Then we combine these opposition solutions with the original population. Finally, we select the top individuals with the best fitness values for the next iteration. This process boosts population diversity and makes early convergence faster.

3.2. Introducing a nonlinear function to improve parameter

In the standard GWO algorithm, parameter a decreases linearly from 2 to 0. This linear decrease cannot fully simulate the complex hunting behavior of grey wolves. It also limits the algorithm's ability to balance exploration and exploitation.

To solve this issue, we design a nonlinear function to adjust parameter a:

$$a = a_{min} + (a_{max} - a_{min}) \cdot e^{-\varepsilon \cdot (t/T)^2} \tag{8}$$

DOI: 10.54254/2755-2721/2026.TJ31353

In this formula: $a_{max} = 2$, $a_{min} = 0.1$; $\varepsilon = 4$ which is the shape control parameter; t is the current iteration; T is the maximum number of iterations.

As shown in Figure 1, the nonlinear variation of a expands the exploration range in the early iterations. This helps the algorithm avoid local optima. In the later iterations, a decreases rapidly. The algorithm then focuses on local exploitation to refine solutions.

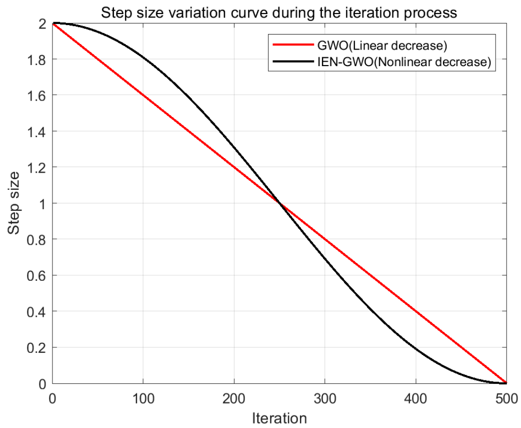

Figure 1. Step size variation curves during GWO and improved GWO iterations

3.3. Implementation procedure of the enhanced Grey Wolf Optimization algorithm

We add elite opposition-based learning and nonlinear parameter tuning to the original Grey Wolf Optimization (GWO) algorithm. This modified algorithm can avoid falling into local optima. It also achieves faster convergence in the later iteration stages. We call this improved algorithm the Improved Grey Wolf Optimization Algorithm (IEN-GWO). The steps of the IEN-GWO algorithm are as follows:

1.Initialization: Set population size (N), maximum iterations (T), dimension (D), and variable bounds (LB, UB).

2.Elite Opposition Population Generation: Generate the initial population, identify elite individuals, compute their opposition solutions using Equations (6) and (7), merge populations, and select the top N individuals by fitness.

3.Leader Selection: Calculate fitness values for all individuals, and select the top three as α, β, δ wolves.

4.Parameter and Position Update: Compute nonlinear parameter a using Equation (8), calculate coefficient vectors A and C via Equation (3), and update positions using Equations (1), (2), and (4).

5.Fitness Evaluation: Compute fitness values for updated individuals and update α, β, δ wolves if new optimal solutions are found.

6.Termination Check: If $t = T$, output the global optimal solution; otherwise, return to Step 3.

4. Experiments and results analysis

4.1. Experimental setup

To gauge how well the IEN - GWO algorithm performs in optimization tasks, 10 classic benchmark test functions were adopted for experimental simulation and comparative assessment. Among these selected functions, F1-F7 are defined as unimodal functions, serving the purpose of testing the

DOI: 10.54254/2755-2721/2026.TJ31353

algorithm's local optimization ability, whereas F8-F10 are multimodal functions that are designed to evaluate its global optimization performance. The specific parameters and characteristics of these functions are summarized in Table 1 as follows.

Table 1. Standard test function information table

No.	Function name	fmin
F1	Sphere Function	0
F2	Schwefel’s problem 2.22	0
F3	Schwefel’s problem1.2	0
F4	Schwefel’s problem 2.21	0
F5	Generalized Rosenbrock’s Function	0
F6	Step function	0
F7	Quartic Function	0
F8	Generalized Penalized Function1	0
F9	Generalized Penalized Function2	0
F10	Shekel’s Family	−10.5363

We wanted to check how well the IEN-GWO algorithm performs in optimization. So we did a comparative analysis with ten benchmark test functions. We chose five typical meta-heuristic algorithms as comparison standards. These algorithms are the original Grey Wolf Optimizer (GWO) [5], Multi-Verse Optimizer (MVO) [11], Slime Mould Algorithm (SMA) [12], Whale Optimization Algorithm (WOA) [3], and Harris Hawks Optimization (HHO) [4].

We set the experiment parameters clearly. The maximum number of iterations is 500, and the population size is 30. All comparative algorithms use the same parameter settings as their original papers. To avoid interference from random factors and experimental bias, we ran each benchmark function independently 30 times. We picked three key metrics for evaluation. The first is the optimal value, which measures how close the result is to the global optimum. The second is the mean value, which shows the convergence precision. The third is the standard deviation, which indicates the algorithm’s stability. We provide detailed test results and related analyses in Table 2.

4.2. Comparative analysis with other intelligent algorithms

Table 2 summarizes the optimal values, mean values, and standard deviations of IEN-GWO and the comparative algorithms. These data come from 30 independent runs on each test function. Meanwhile, Figure 2 shows the partial convergence curves of all algorithms involved in the comparison.

Functions F1 to F7 are unimodal functions. They are used to assess the local exploitation capability of algorithms. As shown in Table 2, the proposed IEN-GWO has the best stability and highest solution accuracy among all unimodal functions. Only on F3, its average accuracy and standard deviation are slightly worse than those of HHO. It’s worth noting that IEN-GWO successfully reaches the theoretical optimal value of F1. This proves its strong local exploitation performance.Moreover, the convergence curves of F1 to F5 show two advantages of IEN-GWO. It converges quickly in the early stage. It also keeps a sustainable search capability in later iterations. These advantages come from the elite opposition-based learning strategy. This strategy expands the search space in the early and middle stages. It helps the algorithm quickly find the optimal search

DOI: 10.54254/2755-2721/2026.TJ31353

direction. It also speeds up convergence. We added nonlinear parameters to the prey encircling and attacking phases. These parameters match GWO's iterative optimization logic. They support continuous refinement of solutions in later stages. They also prevent the algorithm from stopping the search prematurely.

Functions F8 to F10 are multimodal functions. They are used to evaluate the global exploration capability of algorithms. As seen in Table 2, IEN-GWO is better than all comparative algorithms in three aspects: optimal value, mean value, and standard deviation. This shows that IEN-GWO has excellent global search ability. It also works well in solving large-scale optimization problems. Meanwhile, the convergence curves of F8 and F9 tell us more. IEN-GWO converges faster. After convergence, it keeps a stable and continuous downward trend. It also achieves higher final optimization accuracy than other algorithms. This further proves that our improvement strategies are effective. It also shows that the modified IEN-GWO has better performance. As clearly shown in F8's convergence curve, IEN-GWO and the basic GWO converge almost at the same time at the 100th iteration. After that, the basic GWO tends to stabilize. But IEN-GWO escapes local optima at the 300th iteration. It continues searching thanks to its nonlinear search mechanism. This effectively reduces premature convergence. IEN-GWO thus gets better optimization results than other comparative algorithms.

Table 2. Comparison table of test function experimental results

No.	Indicator	IEN-GWO	GWO	MVO	SMA	WOA	HHO
	Best	0.00E+00	5.84E-93	2.23E-03	0.00E+00	7.15E-06	2.25E-203
F1	Mean	6.33E-299	1.51E-86	4.59E-03	3.99E-03	3.98E-04	1.25E-182
	SD	0.00E+00	4.10E-86	1.63E-03	2.00E-02	4.86E-04	0.00E+00
	Best	1.17E-207	1.08E-52	1.29E-02	6.75E-79	6.09E-04	3.67E-109
F2	Mean	1.41E-142	1.04E-49	1.98E-02	6.14E-04	9.58E-03	2.16E-96
	SD	6.29E-142	1.76E-49	4.90E-03	3.35E-03	8.41E-03	1.18E-95
	Best	6.47E-198	2.84E-48	5.87E-03	1.36E-114	4.72E+02	5.10E-192
F3	Mean	3.25E-118	1.48E-41	2.33E-02	3.77E-01	2.14E+03	3.18E-166
	SD	1.27E-117	4.62E-41	1.24E-02	1.31E+00	6.91E+02	0.00E+00
	Best	5.70E-212	7.53E-31	2.76E-02	6.00E-139	3.74E+00	4.45E-101
F4	Mean	8.67E-157	3.32E-28	4.62E-02	4.39E-05	3.51E+01	3.13E-90
	SD	4.75E-156	1.09E-27	1.36E-02	2.22E-04	1.43E+01	1.32E-89
	Best	1.38E-10	4.92E+00	7.11E-01	2.34E-04	6.63E-03	7.89E+00
F5	Mean	3.30E-07	6.08E+00	1.40E+02	4.30E-01	4.06E+01	8.54E+00
	SD	3.01E-07	7.16E-01	3.62E+02	5.53E-01	1.00E+02	3.01E-01
	Best	3.78E-12	8.00E-07	1.72E-03	3.12E-06	6.15E-04	8.12E-03
F6	Mean	2.22E-10	1.92E-06	4.80E-03	8.93E-02	3.62E-03	4.48E-02
	SD	1.16E-10	5.85E-07	2.20E-03	1.23E-01	2.26E-03	2.59E-02
	Best	4.58E-07	2.87E-05	1.40E-04	5.05E-04	1.93E-03	1.31E-07
F7	Mean	3.07E-05	2.20E-04	1.15E-03	6.79E-03	1.33E-02	4.31E-05
	SD	2.91E-05	1.55E-04	7.84E-04	9.10E-03	9.06E-03	4.91E-05

Table 2. (continued)

	Best	9.46E-14	1.06E-07	2.48E-05	8.00E-07	1.98E-03	4.59E-04
F8	Mean	1.59E-10	6.62E-04	2.09E-02	4.33E-03	1.81E+00	2.54E-03
	SD	1.41E-10	3.62E-03	7.91E-02	5.78E-03	3.26E+00	9.22E-04
	Best	6.92E-12	7.05E-07	1.33E-04	2.05E-06	2.57E-04	7.89E-03
F9	Mean	1.04E-09	3.33E-03	1.72E-03	1.88E-02	3.22E-02	2.35E-02
	SD	9.22E-10	1.82E-02	3.49E-03	2.84E-02	2.66E-02	9.43E-03
	Best	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
F10	Mean	-1.05E+01	-1.03E+01	-9.20E+00	-1.05E+01	-1.05E+01	-6.22E+00
	SD	5.64E-08	1.44E+00	2.50E+00	1.33E-02	1.08E-03	2.50E+00

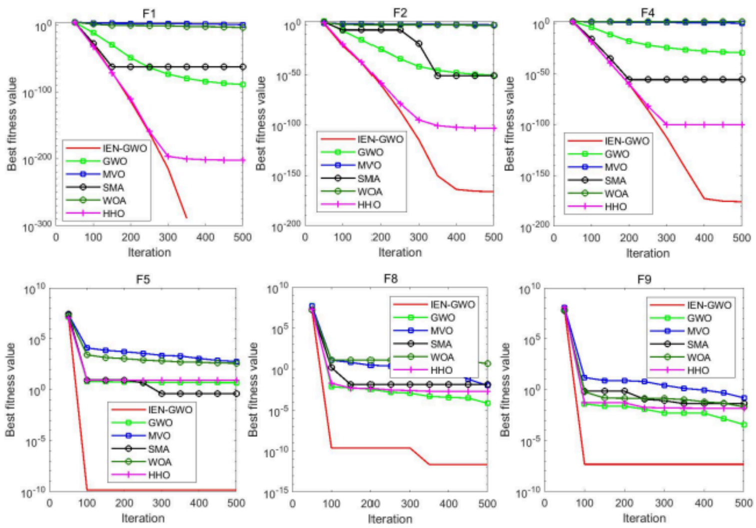

Figure 2. Partial convergence curve diagrams for each comparison algorithm

5. Conclusion

We developed an improved Grey Wolf Optimizer (IEN-GWO) in this paper. We added two strategies: elite opposition-based learning and nonlinear parameter adjustment. The elite opposition-based learning does two things. It increases the diversity of the initial population. It also speeds up early convergence. The nonlinear parameter adjustment balances global exploration and local exploitation. It helps the algorithm escape local optima. Tests on 10 benchmark functions show that IEN-GWO is better than the original GWO and other advanced algorithms. It performs well in solution precision, stability, and convergence rate. Future work will focus on applying IEN-GWO to real-world engineering optimization problems. These problems include feature selection and path planning.

References

[1] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.
[2] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). ieee.
[3] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

DOI: 10.54254/2755-2721/2026.TJ31353

[4] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.
[5] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
[6] Liu, W., Guo, Z. Q., Jiang, F., et al. (2023). Improved grey wolf algorithm based on cooperative siege strategy and its PID parameter optimization. Journal of Computer Science and Exploration, 17(3), 620-634.
[7] Sun, B., Zhou, J. K., Zhao, Y. Q., et al. (2024). Global path planning of robots based on improved grey wolf optimization algorithm. Science Technology and Engineering, 24(33), 14287-14297.
[8] Banaie-Dezfouli, M., Nadimi-Shahraki, M. H., & Beheshti, Z. (2021). R-GWO: Representative-based grey wolf optimizer for solving engineering problems. Applied Soft Computing, 106, 107328.
[9] Seyedabbasi, A., & Kiani, F. (2021). I-GWO and Ex-GWO: Improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. Engineering with Computers, 37(1), 509-532.
[10] Meidani, K., Hemmasian, A., Mirjalili, S., & Barati Farimani, A. (2022). Adaptive grey wolf optimizer. Neural Computing and Applications, 34(10), 7711-7731.
[11] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural computing and applications, 27(2), 495-513.
[12] Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. Future generation computer systems, 111, 300-323.