ISSN 2616-5775 Vol. 7, Issue 3: 32-37, DOI: 10.25236/AJCIS.2024.070304

Modified Grey Wolf Optimization Hybridized with Teaching and Learning Mechanism for Solving Optimization Problems

Zehua Lin[1,*]

[1]*College of Science, Tianjin University of Commerce, Tianjin, China*
[*]*Corresponding author*

Abstract: *The Grey Wolf Optimization (GWO) algorithm, inspired by grey wolf social behaviors, has shown excellent performance in various optimization problems. However, it faces limitations in handling dynamic optimization problems. To address this, we propose an enhanced version, Merged Teaching and Learning Grey Wolf Optimization (MTLGWO). MTLGWO introduces a two-phase teaching and learning strategy, improving global exploration and local exploitation capabilities. The core improvements include using Latin Hypercube Sampling for better population initialization and adopting a group teaching mechanism to simulate diverse teaching strategies. Through comprehensive performance testing on CEC2017 basic test functions, MTLGWO demonstrates superior performance in terms of convergence accuracy, stability, and convergence speed. Compared with other classical heuristic optimization algorithms, MTLGWO proves its potential and reliability as an efficient tool for solving optimization problems. These results highlight MTLGWO's potential as an efficient tool for practical optimization problems.*

Keywords: *Bionic intelligent computing, Grey Wolf Optimizer, modified Grey Wolf Optimizer, teaching-learning-based optimization*

1. Introduction

Optimization algorithms represent one of the most active and rapidly evolving fields in today's domains of science, technology, and engineering. With continual advancements in computer science, industrial engineering, mechanical engineering, and management science, optimization problems and their solutions play increasingly vital roles in various aspects of production, life, and scientific research. Notably, metaheuristic algorithms have emerged as crucial tools for addressing NP-hard (Non-deterministic Polynomial-time hard) problems, adapting well to the uncertainties and complexities inherent in optimization processes.

Metaheuristic algorithms, inspired by natural phenomena or physical principles, share the common characteristic of not imposing specific requirements on the objective function or constraint conditions of a problem. They do not rely on precise analytical expressions or mathematical models. Instead, these algorithms effectively handle the uncertainties associated with optimization problems and are adaptable to a variety of complex optimization scenarios.

In recent years, as research has deepened and technologies have advanced, numerous novel metaheuristic algorithms have been introduced. Seyedali Mirjalili[1] introduced the Grey Wolf Optimization algorithm (GWO), inspired by the hierarchical leadership and hunting mechanisms of grey wolf populations, featuring alpha, beta, delta, and omega wolves to simulate leadership hierarchies. Building upon the original GWO, many scholars have continually improved the algorithm. Mohammad H. Nadimi-Shahraki[2] proposed the Improved Grey Wolf Optimizer (I-GWO) to address global optimization and engineering design problems, alleviating issues of population diversity, imbalance between exploration and exploitation, and premature convergence in the GWO algorithm. Nitin Mittal[3] presented the modified GWO (mGWO), focusing on achieving an appropriate balance between exploration and exploitation for optimal algorithm performance. Xianqiu Meng[4] addressed issues of local stagnation and premature convergence when handling specific datasets, proposing an Advanced Grey Wolf Optimization algorithm (AGWO) with elastic, cyclic, and attack mechanisms. Mehak Kohli[5] introduced chaos theory to the GWO algorithm, proposing the Chaos Grey Wolf Optimization algorithm (CGWO) for solving constrained optimization problems. Rahul Kumar Vijay[6] developed the Quantum

Grey Wolf Optimizer (QGWO) algorithm, applying it to seismic activity modeling in various regions. Ashutosh Bhadoria[7] developed a hybrid version, combining the Grey Wolf Optimization algorithm with simulated annealing (hGWO-SA), for solving nonlinear, highly constrained, and non-convex engineering design and optimization problems.

This paper delves into a comprehensive study of the Grey Wolf Optimization algorithm (GWO), focusing on improvements in population initialization, parameters, and search mechanisms. A novel approach, named Merged Teaching and Learning Grey Wolf Optimization (MTLGWO), is proposed.

The enhancements presented in this paper are divided into two main parts. In the first part, addressing population initialization, the Latin Hypercube Sampling method (LHS)[8] is employed to initialize the grey wolf population. This ensures that the algorithm's initial search points comprehensively cover the solution space, mitigating the risk of falling into local optima and increasing the likelihood of finding global optimum solutions.The second part introduces an iterative process divided into education and learning stages. In the education stage, the population is split into two subsets, A and B, based on the individual positions. Subset A comprises individuals with better positions, while subset B contains those with poorer positions. Different updating mechanisms are applied to transform the positions of individuals in subsets A and B. The learning stage is further divided into individual self-learning and mutual learning phases. In the self-learning phase, grey wolves update their positions based on their historical optimal positions. In the mutual learning phase, wolves randomly learn from individuals (neighbors) within the population who outperform themselves, or update their positions based on the overall population's positions.

2. Merged Teaching and Learning Grey Wolf Optimization (MTLGWO)

The Grey Wolf Optimization (GWO) algorithm, while widely recognized for its simplicity, minimal parameter tuning requirements, and robust global search capability, may exhibit slower convergence speeds, particularly when faced with complex or high-dimensional optimization problems. Additionally, the algorithm may sometimes prematurely converge to local optimal solutions, especially in the presence of vast search spaces or structurally intricate problems. Despite having fewer parameters, the performance of GWO remains sensitive to parameter settings, and inappropriate configurations can lead to a significant degradation in performance. When dealing with multi-peaked problems containing multiple local optimal solutions, GWO might struggle to distinguish and select the global optimum.

In addressing these issues, this section introduces a teaching and learning mechanism into the Grey Wolf Optimization algorithm, proposing a Hybrid Teaching and Learning Grey Wolf Optimization algorithm (MTLGWO). This section provides a detailed introduction to the fundamental concepts of MTLGWO.MTLGWO consists of two stages: the teaching phase and the learning phase.

1) Teaching phase

During the teaching phase of the grey wolf population, based on the individual positions (determined by their fitness values), the population is divided into two parts: a superior subset and an inferior subset. The individuals in the superior subset have better positions, while those in the inferior subset have poorer positions. The positions of the superior subset are primarily influenced by the position of the alpha wolf, whereas the positions of the inferior subset are influenced not only by the position of the alpha wolf but also by the average position of the wolf pack. Different updating mechanisms are applied to these two subsets of individuals for position transformation, with the update formulas as follows:

$$X_{new,i} = \omega_1 \times X_i + \omega_2 \times X_p - A \times D \quad \text{if } f\left(X_i\right) < f\left(X_{\text{mean}}\right) \tag{1}$$

$$X_{new,i} = (X_i + (C-1) \times E) \times \omega_1 + diff \times \omega_2 \quad \text{if } f\left(X_i\right) > f\left(X_{\text{mean}}\right) \tag{2}$$

$X_{new,i}$ represents the updated position of the grey wolf individual, X_p is the position of the alpha wolf (prey), X_{mean} is the average position of the current population, w_1 and w_2 are weighting factors used to adjust the weights of the current position and the alpha wolf's position, A is a random number controlling the intensity or step size of the search, C is a random number in the [0,2] interval, D represents the distance between the current wolf and the target (prey), simulating the interaction and hunting behavior among wolf pack members, E represents the distance between the current wolf and the average position of the population, and $diff$ represents the distance between the prey and the average position of the grey wolf population. The calculation formulas are as follows:

$$A = a \times (2 \times r - 1) \tag{3}$$

ISSN 2616-5775 Vol. 7, Issue 3: 32-37, DOI: 10.25236/AJCIS.2024.070304

$$D = \left| C \cdot X_p - X_i \right| \tag{4}$$

$$E = X_{mean} - X_i \tag{5}$$

$$diff = r \times \left(X_p - T_{\text{factor}} \times X_{mean} \right) \tag{6}$$

$$\omega_1 = \sin\left(\frac{\pi}{2} \times \frac{iter}{MaxIter} \right) \tag{7}$$

$$\omega_2 = \cos\left(\frac{\pi}{2} \times \frac{iter}{MaxIter} \right) \tag{8}$$

$$a = a_{\max} - ((a_{\max} - a_{\min}) \times \frac{iter}{MaxIter}) \tag{9}$$

In this context, *iter* represents the current iteration count, *MaxIter* denotes the maximum number of iterations, *r* is a random number within the [0,1] interval, *a* is a dynamic decay factor whose magnitude depends on the values of a_{min} and a_{max}.

Finally, as each grey wolf individual is guided by three alpha, beta, and delta wolves, they obtain three positions. The average of these three positions serves as the final position. The specific updating formula is as follows:

$$X_{new,i} = \frac{\left(X_1 + X_2 + X_3 \right)}{3} \tag{10}$$

2) learning phase

The learning phase consists of two parts: the individual self-learning phase (first phase) and the inter-group mutual learning phase (second phase). In the individual self-learning phase, grey wolves update their current positions based on their own historical best positions. The formula is as follows:

$$X_{new,i} = X_i + \left(X_{i,best} - X_i \right) \times a \times r \tag{11}$$

Where $X_{i,best}$ represents the individual grey wolf's own historical best position (optimal fitness), *a* is a dynamic decay factor, and *r* is a random number within the [0,1] interval.

During the inter-group mutual learning phase, grey wolves randomly learn from individuals within the population who have superior fitness (neighbors), or they update their positions based on the overall position of the population. The formula is as follows:

$$X_{new,i} = X_i + (X_{\text{neighbour}} - X_i) \times a \times r \quad \text{if} \quad f\left(X_i\right) > f\left(X_{\text{neighbour}}\right) \tag{12}$$

$$X_{new,i} = X_i + (X_{best} - X_{worst}) \times (X - 1) \quad \text{if} \quad f\left(X_i\right) < f\left(X_{\text{neighbour}}\right) \tag{13}$$

Where $X_{\text{neighbour}}$ represents a randomly selected neighbor within the population, X_{best} denotes the current best position of the population, and X_{worst} represents the worst position within the current population.

3. Algorithm performance testing

To validate the effectiveness of MTLGWO, this paper continues testing using the CEC2017 basic test functions [9], denoted as TF1-TF20. The theoretical optimal values for these functions are all 0. The mathematical formulas for the test functions are presented in the table 1 below:

Table 1: CEC2017 basic test functions

function	Dimension	Range	Global solution
$TF_1(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	D	$[-10,10]^D$	0
$TF_2(x) = \sum_{i=1}^{D} \left\| x_i \right\|^{i+1}$	D	$[-100,100]^D$	0

ISSN 2616-5775 Vol. 7, Issue 3: 32-37, DOI: 10.25236/AJCIS.2024.070304

$TF_3(x)=\sum_{i=1}^{D}x_i^2+\left(\sum_{D}^{i=1}0.5x_i\right)^2+\left(\sum_{D}^{i=1}0.5x_i\right)^4$	D	[-5,10]^D	0
$TF_4(x)=\sum_{i=1}^{D-1}\left(100\left(x_i^2-x_{i+1}\right)^2+\left(x_i-1\right)^2\right)$	D	[-10,10]^D	0
$TF_5(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos\left(2\pi x_i\right)+10\right)$	D	[-5.12,5.12]^D	0
$TF_6(x)=g\left(x_1,x_2\right)+g\left(x_2,x_3\right)+\ldots+g\left(x_{D-1},x_D\right)+g\left(x_D,x_1\right)$ $g(x,y)=0.5+\frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right)-0.5\right)}{\left(1+0.001\left(x^2+y^2\right)\right)^2}$	D	[-10,10]^D	0
$TF_7(x)=\min\left(\sum_{i=1}^{D}\left(\hat{x}_i-\mu_0\right)^2,dD+s\sum_{i=1}^{D}\left(\hat{x}_i-\mu_1\right)^2\right)+10\left(D-\sum_{i=1}^{D}\cos\left(2\pi\overline{z}_i\right)\right)$ $\mu_0=2.5,\mu_1=-\sqrt{\frac{\mu_0^2-d}{s}},s=1-\frac{1}{2\sqrt{D+20}-8.2},d=1$ $y=\frac{10(\mathbf{x}-\mathbf{o})}{100},\hat{x}_i=2\operatorname{sign}\left(x_i^*\right)y_i+\mu_0,\text{ for } i=1,2,\ldots,D$ $z=\Lambda^{100}\left(\hat{\mathbf{x}}-\mu_0\right)$	D	[-5,5]^D	0
$TF_8(x)=\sum_{i=1}^{D}\left(z_i^2-10\cos\left(2\pi z_i\right)+10\right)+TF_{13}$	D	[-10,10]^D	0
$TF_9(x)=\sin^2\left(\pi w_1\right)+\sum_{i=1}^{D-1}\left(w_i-1\right)^2\left[1+10\sin^2\left(\pi w_t+1\right)\right]+\left(w_D-1\right)^2\left[1+\sin^2\left(2\pi w_D\right)\right]$ $w_i=1+\frac{x_i-1}{4},\forall i=1,\ldots,D$	D	[-10,10]^D	0
$TF_{10}(\mathbf{x})=418.9829\times D-\sum_{i=1}^{D}g\left(z_i\right),\quad z_i=x_i+4.209687462275036\mathrm{e}+002$ $g\left(z_i\right)=\begin{cases}z_i\sin\left(\lvert z_i\rvert^{1/2}\right) & \text{if } \lvert z_i\rvert\le 500\\ \left(500-\operatorname{mod}\left(z_i,500\right)\right)\sin\left(\sqrt{\lvert 500-\operatorname{mod}\left(z_i,500\right)\rvert}\right)-\frac{\left(z_i-500\right)^2}{10000D} & \text{if } z_i>500\\ \left(\operatorname{mod}\left(\lvert z_i\rvert,500\right)-500\right)\sin\left(\sqrt{\lvert\operatorname{mod}\left(\lvert z_i\rvert,500\right)-500\rvert}\right)-\frac{\left(z_i+500\right)^2}{10000D} & \text{if } z_i<-500\end{cases}$	D	[-10,10]^D	0
$TF_{11}(x)=\sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}}\mathbf{x}_i^2$	D	[-10,10]^D	0
$TF_{12}(x)=10^6x_1^2+\sum_{i=2}^{D}x_i^2$	D	[-10,10]^D	0
$TF_{13}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right)+20+\mathrm{e}$	D	[-32,32]^D	0
$TF_{14}(x)=\sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(x_i+0.5\right)\right)\right]\right)-D\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\cdot 0.5\right)\right]$ $a=0.5,b=3,k\max=20$	D	[-10,10]^D	0
$TF_{15}(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	D	[-600,600]^D	0
$TF_{16}(x)=\frac{10}{D^2}\prod_{i=1}^{D}\left(1+i\sum_{j=1}^{32}\frac{\lvert 2^jx_i-\operatorname{round}\left(2^jx_i\right)\rvert}{2^j}\right)^{\frac{10}{D^{12}}}-\frac{10}{D^2}$	D	[-10,10]^D	0
$TF_{17}(x)=\left\lvert\sum_{i=1}^{D}x_i^2-D\right\rvert^{1/4}+\left(0.5\sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}x_i\right)/D+0.5$	D	[-10,10]^D	0
$TF_{18}(x)=\left\lvert\left(\sum_{i=1}^{D}x_i^2\right)^2-\left(\sum_{i=1}^{D}x_i\right)^2\right\rvert^{1/2}+\left(0.5\sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}x_i\right)/D+0.5$	D	[-10,10]^D	0
$TF_{19}(x)=TF_7\left(TF_4\left(x_1,x_2\right)\right)+TF_7\left(TF_4\left(x_2,x_3\right)\right)+\ldots$ $+f_7\left(f_4\left(x_{D-1},x_D\right)\right)+TF_7\left(TF_4\left(x_D,x_1\right)\right)$	D	[-10,10]^D	0
$TF_{20}(x)=\left[\frac{1}{D-1}\sum_{i=1}^{D-1}\left(\sqrt{s_i}\cdot\left(\sin\left(50.0s_i^{0.2}\right)+1\right)\right)\right]^2,s_i=\sqrt{x_i^2+x_{i+1}^2}$	D	[-10,10]^D	0

In this section, these test functions are utilized to verify the performance of MTLGWO. The population size D is set to 30, and the maximum number of iterations is set to 500. To reduce statistical errors, the algorithm independently runs 30 times, and the average and standard deviation of the optimal fitness values from these 30 runs are calculated. The average represents convergence accuracy, and the standard deviation indicates the stability of the algorithm. Smaller average values and standard deviations correspond to better algorithm performance.

On the test functions, MTLGWO is compared for performance against six classical heuristic optimization algorithms: Cat Swarm Optimization (CSO) [10], Particle Swarm Optimization (PSO)[11], Cuckoo Search (CS)[12], Artificial Bee Colony (ABC)[13], Whale Optimization Algorithm (WOA)[14], and

ISSN 2616-5775 Vol. 7, Issue 3: 32-37, DOI: 10.25236/AJCIS.2024.070304

Artificial Fish Swarm Algorithm (AFSA)[15]. The testing results are presented in the table 2 below:

Table 2: Testing results of MTLGWO and six classical heuristic optimization algorithms on CEC2017 functions

function		MTLGWO	CSO	PSO	CS	ABC	WOA	AFSA
TF1	Ave	2.45e-244	2.77e-09	2.32e-07	3.77e-07	2.45e-06	5.40e-71	2.74e-02
	Std	0.00e+00	5.34e-08	1.54e-07	5.33e-06	1.44e-06	9.80e-71	3.69e-03
TF2	Ave	1.43e-177	2.47e-47	6.66e-24	3.84e-14	8.27e-22	1.28e-68	4.64e-04
	Std	0.00e+00	7.05e-47	1.67e-25	5.84e-14	2.46e-23	3.15e-68	1.68e-04
TF3	Ave	5.26e-126	2.84e-07	3.17e-02	7.83e-01	3.17e-02	4.88e-02	3.46e-01
	Std	1.58e-125	5.40e-07	1.47e-02	1.92e-01	2.92e-01	9.41e-01	3.62e-01
TF4	Ave	0.00e+00	5.44e-01	1.40e-01	8.54e-01	1.14e-01	0.00e+00	9.86e-07
	Std	0.00e+00	1.41e-01	1.05e-01	9.71e-02	3.52e-01	0.00e+00	1.53e-06
TF5	Ave	0.00e+00	3.24e-01	6.93e-01	2.15e-01	3.84e-02	0.00e+00	1.02e-09
	Std	0.00e+00	1.09e-01	5.70e-01	2.59e-01	2.62e-02	0.00e+00	1.62e-10
TF6	Ave	4.11e+00	4.48e+00	2.41e+00	3.35e+00	2.49e+00	3.79e+00	3.51e-01
	Std	5.00e-01	9.76e-01	8.80e-01	4.28e-01	3.59e-01	1.95e+00	6.54e-02
TF7	Ave	0.00e+00	2.80e-01	1.68e-01	2.76e-01	1.28e-01	0.00e+00	1.51e-08
	Std	0.00e+00	6.89e-01	4.62e-01	1.26e-01	2.31e-01	0.00e+00	2.31e-09
TF8	Ave	0.00e+00	1.27e-02	2.92e-01	2.21e-01	8.26e-02	0.00e+00	9.39e-10
	Std	0.00e+00	3.47e-01	3.22e-01	3.09e-01	4.86e-02	0.00e+00	1.72e-10
TF9	Ave	9.87e-01	6.05e-01	1.16e-01	2.04e-01	4.25e-01	4.17e-01	1.79e-02
	Std	3.13e-01	2.09e-01	3.48e-00	5.81e-01	1.63e-01	1.89e-01	3.58e-02
TF10	Ave	3.82e-04	3.31e-03	1.07e-02	5.85e-01	5.16e+00	3.82e-04	3.82e-04
	Std	0.00e+00	1.06e-03	1.59e-02	4.08e+00	2.77e+00	1.78e-12	3.59e-09
TF11	Ave	6.12e-245	1.29e-08	1.02e-06	2.07e-05	2.45e-04	2.92e-70	5.41e-02
	Std	0.00e+00	5.33e-07	9.57e-05	6.14e-04	2.77e-04	7.56e-70	1.94e-02
TF12	Ave	3.08e-247	8.34e-02	2.64e-02	4.13e-01	3.75e-01	6.04e-76	8.05e+00
	Std	0.00e+00	1.94e-02	9.90e-01	5.94e-01	5.72e-01	1.64e-75	1.03e+00
TF13	Ave	4.44e-16	1.98e-01	8.95e-02	3.19e-01	5.77e-01	3.29e-15	2.65e-01
	Std	0.00e+00	5.71e-01	1.46e-02	2.30e-01	7.90e-01	2.13e-15	4.26e-01
TF14	Ave	0.00e+00	1.19e-01	4.56e-01	2.82e-01	1.06e-01	0.00e+00	4.75e-01
	Std	0.00e+00	1.19e-01	9.70e-01	1.53e-01	1.20e-01	0.00e+00	9.88e-02
TF15	Ave	6.19e-03	1.38e-04	1.39e-02	7.31e-03	7.45e-01	3.14e-02	1.52e-02
	Std	1.75e-02	4.04e-03	9.46e-01	9.80e-02	3.28e-01	9.43e-02	1.08e-02
TF16	Ave	3.54e-14	0.00e+00	2.77e-02	4.53e-02	1.44e-02	0.00e+00	1.31e-02
	Std	1.06e-13	0.00e+00	3.60e-02	6.89e-03	3.98e-03	0.00e+00	2.26e-03
TF17	Ave	6.19e-02	7.58e-02	3.66e-02	1.04e-01	1.11e-01	5.67e-02	1.05e-02
	Std	1.28e-02	3.35e-02	1.05e-02	1.63e-02	1.53e-02	1.13e-02	2.46e-03
TF18	Ave	3.74e-02	5.01e-02	3.68e-02	7.28e-02	6.11e-02	4.19e-02	1.04e-03
	Std	1.34e-02	2.46e-02	1.02e-02	1.24e-02	1.03e-02	9.92e-03	4.61e-04
TF19	Ave	0.00e+00	1.56e-01	4.79e-01	9.49e-01	5.01e-02	0.00e+00	1.11e-17
	Std	0.00e+00	7.68e-01	2.04e-01	2.15e-01	1.09e-02	0.00e+00	3.33e-17
TF20	Ave	1.57e-124	5.40e-01	2.37e-01	1.33e-02	1.53e+00	2.90e-58	2.27e-01
	Std	2.73e-124	7.87e-01	2.40e-01	3.16e-01	3.50e-01	8.10e-58	1.98e-02

Based on the testing results from TF1 to TF20, we can conduct a performance analysis. Regarding optimization accuracy, MTLGWO demonstrates high precision across multiple test functions, particularly achieving average values close to the mathematical limits in TF1, TF2, TF11, and TF12. This surpasses the performance of other algorithms, indicating that MTLGWO can approach global optimum solutions very closely for these problems. In terms of stability and consistency, MTLGWO shows extremely low standard deviations, close to zero or zero, across almost all test functions. This highlights the algorithm's exceptional stability and consistency, suggesting that MTLGWO reliably achieves consistently high-precision results in repeated runs.

Comparing with other algorithms, MTLGWO's average values outperform CSO, PSO, CS, ABC, and WOA algorithms across most test functions. This demonstrates the superiority of MTLGWO in these optimization problems, particularly in handling complex problems or those requiring high precision. It's noteworthy that in certain test functions, such as TF4, TF5, TF9, TF10, etc., other algorithms also show good performance, but MTLGWO generally maintains strong performance or performs equally well

ISSN 2616-5775 Vol. 7, Issue 3: 32-37, DOI: 10.25236/AJCIS.2024.070304

compared to other algorithms.

4. Conclusions

The main work and innovations of this study are summarized as follows:

1) Improvement of Optimization Algorithm: Effective enhancements were made to the original Grey Wolf Optimization (GWO) algorithm, resulting in the proposed MTLGWO algorithm. The new algorithm introduces the Latin Hypercube Sampling (LHS) method during population initialization, significantly enhancing the algorithm's initial search capabilities. Through the implementation of a grouping teaching mechanism and a novel individual position updating strategy, the algorithm's performance in global exploration and local exploitation has been strengthened.

2) Validation and Comparative Analysis of Algorithm Performance: The superiority of the MTLGWO algorithm was validated through performance testing on the CEC2017 benchmark functions. In comparison to six classical heuristic optimization algorithms, MTLGWO demonstrated significant advantages in terms of convergence accuracy and stability.

References

[1] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in engineering software, 2014, 69: 46-61.

[2] Nadimi-Shahraki M H, Taghian S, Mirjalili S. An improved grey wolf optimizer for solving engineering problems[J]. Expert Systems with Applications, 2021, 166: 113917.

[3] Mittal N, Singh U, Sohi B S. Modified grey wolf optimizer for global engineering optimization[J]. Applied Computational Intelligence and Soft Computing, 2016,2016.

[4] Meng X, Jiang J, Wang H. AGWO: Advanced GWO in multi-layer perception optimization[J]. Expert Systems with Applications, 2021, 173: 114676.

[5] Kohli M, Arora S. Chaotic grey wolf optimization algorithm for constrained optimization problems[J]. Journal of computational design and engineering, 2018, 5(4): 458-472.

[6] Vijay R K, Nanda S J. A quantum grey wolf optimizer based declustering model for analysis of earthquake catalogs in an ergodic framework[J]. Journal of Computational Science, 2019, 36: 101019.

[7] Bhadoria A, Marwaha S, Kamboj V K. A solution to statistical and multidisciplinary design optimization problems using hGWO-SA algorithm[J]. Neural Computing and Applications, 2021, 33: 3799-3824.

[8] Olsson A M J, Sandberg G E. Latin hypercube sampling for stochastic finite element analysis[J]. Journal of Engineering Mechanics, 2002, 128(1): 121-125.

[9] Wu G, Mallipeddi R, Suganthan P N. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization [J]. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 2017.

[10] Bahrami M, Bozorg-Haddad O, Chu X. Cat swarm optimization (CSO) algorithm[J]. Advanced optimization by nature-inspired algorithms, 2018: 9-18.

[11] Shi Y. Particle swarm optimization[J]. IEEE connections, 2004, 2(1): 8-13.

[12] Yang X S, Deb S. Cuckoo search: recent advances and applications[J]. Neural Computing and applications, 2014, 24: 169-174.

[13] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. Journal of global optimization, 2007, 39: 459-471.

[14] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[15] Li X L. A new intelligent optimization-artificial fish swarm algorithm[J]. Doctor thesis, Zhejiang University of Zhejiang, China, 2003, 27