IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2023, accepted 21 November 2023, date of publication 27 November 2023, date of current version 6 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337209

RESEARCH ARTICLE

Improved Grey Wolf Optimization Algorithm Based on Hyperbolic Tangent Inertia Weight

WEIMING LIN
Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K.
e-mail: minggg2018@163.com

ABSTRACT The Grey Wolf Optimization Algorithm (GWO) replicates the leadership and foraging mechanisms of the natural grey wolf and excels in solving problems in a variety of domains. However, the algorithm tends to converge to a local optimal and has a slow convergence rate. This paper proposes an enhanced Grey Wolf optimization algorithm (HTGWO) based on hyperbolic tangent inertia weights to solve this problem. HTGWO employs inertia weights based on hyperbolic tangent functions to balance GWO's global and local search capabilities of the GWO. HTGWO has a faster convergence rate and more accurate solutions than GWO. Five classical test functions were used to construct comparative experiments between the HWGWO and five classical intelligent optimization algorithms. The comparison results indicate that HWGWO has superior convergence speed, solution precision, and stability compared with the other five classical intelligent optimization algorithms. Moreover, experimental evidence suggests that the HWGWO is more effective at solving multimodal functions than unimodal functions. In addition, HTGWO can balance exploration and exploitation by adjusting the parameters according to the characteristics of different problems.

INDEX TERMS Hyperbolic tangent, inertia weight, convergence, accuracy, unimodal, multi-model.

I. INTRODUCTION

This article proposes a method to prevent GWO from falling into local optima, accelerate convergence speed, and improve GWO's ability to solve different engineering problems. Because the hyperbolic tangent function is an S-type function, it can effectively balance the global and local searches. Therefore, in this study, a hyperbolic tangent inertia weight is developed to solve this problem. The main contributions of this work can be summarized as follows. This paper proposes a GWO algorithm based on hyperbolic tangent inertia weight (HTGWO) to reconcile the global search and local search abilities and find the global optimum. The method improves the performance of the algorithm as a whole and the ability of the algorithm to cope with different problems.

With the advancement of science and technology in recent decades, optimization problems have become prevalent in numerous disciplines. Researchers have used algorithms based on swarm intelligence to identify the optimal solution among numerous schemes and parameter values. Standard intelligent optimization algorithms consist of the particle swarm optimization (PSO) algorithm [1], antlion optimization (ALO) algorithm [2], firefly algorithm (FA) [3], Fruit Fly optimization algorithm (FOA) [4], and Grey Wolf optimization (GWO) algorithm [5].

Simulating the predation behavior of grey wolves optimizes the GWO. This algorithm has a basic structure, fewer parameters, and capacity for rapid implementation. Its convergence factor adjustment and information feedback mechanism allow for a significant balancing of local and global searches. GWO has been utilized in numerous research projects because of its high accuracy and convergence speed. Qiao et al. [6] proposed a GWO that increases population diversity by employing lost wolves and mating strategies to prevent populations from reaching local optima. Miao et al. [7] upgraded the GWO by adding crossover and mutation operators to the optimization process to solve the problem of precise planning and construction of a fuel cell system model to obtain the optimal parameters of the Proton Exchange Membrane Fuel Cell. Fang et al. [8] designed an adaptive grey Wolf optimizer to address the optimization problem in 5G systems with ample design space, resulting in improved electromagnetic interference shielding performance and increased system stability. Lei and Ouyang [9]

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

utilized GWO to compensate for the lack of robustness of the algorithm against noise in image segmentation. This essential technology combines enhanced GWO and KIFCM algorithms, which can substantially enhance the robustness of the algorithm.

Furthermore, GWO has been utilized in industrial production. For instance, biodiesel output is directly influenced by the process parameters. Kumar et al. [10] used a novel hybrid genetic program, GWO, to optimize the production process and augment its precision and resilience. GWO can also be applied to path planning models for multiple UAVs. Huang et al. [11] developed a hybrid discrete intelligence algorithm based on GWO, which enhanced the global convergence capability of the original algorithm and reduced the energy consumption of UAVs in various states.

It is clear from these studies that GWO is characterized by easy implementation, simple concepts, simple formulas, and few parameters and is more potent than other algorithms for exploring complex problem spaces. They are also widely used in various practical applications. For this reason, we chose to improve the GWO algorithm, which is more beneficial for engineering applications in this field.

The remainder of this paper is organized as follows. Section II describes the current research progress in GWO and related work on other improvements to GWO, discusses the strengths and limitations of existing modifications to GWO, and presents the need for the current work. Section III introduces the concepts and formulas of the GWO. Section IV details the principles of HTGWO and the need for improvement. Section V focuses on the experimental tests and results. Section VI presents an application and case study of HTGWO. Finally, Section VII summarizes the study.

II. RELATED WORK

To address some of the shortcomings mentioned earlier in GWO. The sigmoid function was introduced to GWO to reduce GWO computation and accelerate convergence when solving the product backpack problem [12]. However, the step function leads to a fast convergence in the mid-term of the iteration. Liu et al. [13] added a lion optimization algorithm and dynamic weight algorithm to GWO to resolve the issues of delayed convergence and local optima. In a typical GWO, a waiting period exists. The updated algorithm optimizes more effectively than the original algorithm owing to its dynamic weight and diversity. Nevertheless, the introduced dynamic weight function adjusts the weight with a fixed threshold as the boundary point along with the number of iterations, which may result in a single weight change and have a weak effect on optimizing the algorithm itself.

Zhang et al. [14] devised a dynamic algorithm to grant each wolf algorithm independence without waiting for other units to complete the comparison to eliminate the waiting period and increase the speed. This method verifies that the enhancements based on the dynamic GWO are superior to those based on static GWO. In addition, the GWO has undergone numerous enhancements. Although the convergence speed was fast under the same number of iterations, the running time of the dynamic GWO was longer than that of the GWO. Guo et al. [15] proposed a GWO based on the tracking mode, which addresses the constraint problem in function optimization more effectively. Singh [16] improved the accuracy of the solution by modifying the circling behavior of the algorithm and revising the position equation. These methods need to help find global optima in some unimodal and multimodal functions. In some complex function problems, although the convergence speed is fast in the early stage, it easily falls into a local optimum in the early stage of iteration.

Qin et al. [17] employed a fuzzy GWO strategy to strike a balance between convergence and diversity, thereby accelerating the location of the optimal solution. In addition, the chaotic variant of GWO [18] and a GWO based on oppositional learning [19] can effectively address the issue of early convergence and enhance the exploration capability of the algorithm. However, more experiments are needed during the design phase, and there are no higher-dimensional experiments. The performance of the algorithms in specific problems deteriorates with increasing dimensions. To address this issue, Sun et al. [20] proposed an equalized GWO with refraction reverse learning, which surmounted the low swarm population diversity of the GWO in the final iteration. Nevertheless, it only accelerates convergence and considers that the direction of the individual solutions is limited, which has the possibility of falling into a local optimum in the early stages of iteration.

Ou et al. [21] proposed a method that combines the clone selection algorithm and GWO and uses a nonlinear function to adjust the convergence factor, thus overcoming the problems of slow convergence, low accuracy of the single-peak function, and local optimum in the standard GWO. Nadimi-Shahraki et al. [22] published a discrete improved GWO that uses a local and dimension-based learning amount hunting search to enhance the performance of the algorithm. In addition, the AGWO [23] algorithm uses an adaptive GWO to solve the problem by automatically adjusting the exploration and exploitation parameters based on fitness history. Ma et al. [24] increased the search range of GWO by combining the exploitation capabilities of GWO with the exploration capabilities of the Aquila Optimizer to improve the global search capability. To further solve the local optimum, Liang et al. [25] used a reverse learning strategy to increase the population diversity and then used a simulated annealing algorithm to jump out of the local optimum. In addition, Li et al. [26] used nonlinear factors and adaptive position updating to balance exploration and exploitation and solve the local optimum. However, all of these algorithms suffer from slow convergence in the early iterations, and the algorithm performance is related to the problem chosen and cannot be adapted to a specific problem.

In addition, BGWO [27], which uses an adaptation function as a local search strategy, performs better than IGWO [28], which does not use a balancing strategy, although both algorithms outperform GWO. This demonstrates the importance of balancing the local search and searching in the algorithm. For this reason, El-Ela et al. [29] proposed an update mechanism by modifying the parameters of the algorithm, thereby balancing exploration and exploitation. Shu et al. [30] used nonlinear convergent silvers and weighting factors as a balancing strategy. However, these algorithms require more experiments to address complex functional problems in higher dimensions, even though they accelerate the convergence speed and accuracy of the GWO in low dimensions. Hu et al. [31] employed an improved Wolf bootstrapping of GWO to achieve better performance in an unknown space, which both improved the convergence speed and prevented the algorithm from falling into a local optimum. However, the mechanism of the three wolves' positions interacting with each other at the beginning of the iterations limits the exploration ability of the algorithm.

Existing algorithms for GWO improvement have some limitations, such as slow convergence in the early stage and inability to balance global and local search capabilities well, leading the algorithms to fall into local optima. In addition, these algorithms cannot be adapted to specific problems that are reflected in different function problems and dimensions. Therefore, this paper proposes a method that balances exploration and exploitation, which enhances the ability of GWO to escape from local optima by introducing a Hyperbolic Tangent Inertia Weight into the position-updating formula and improves the convergence speed and accuracy. In addition, the method can adjust the Hyperbolic Tangent Inertia Weight formula for different problems to improve the ability of the algorithm to handle different problems.

III. GREY WOLF OPTIMIZATION ALGORITHM

The global search and local search for GWO algorithm steps are as follows:

1) Initialization parameters: maximal number of iterations M, population size N, spatial population dimension dim, convergence factor a, and coefficient vectors A and C.
2) According to the test function, the fitness of the initial grey wolf individual was obtained, and the three wolves with the best fitness were preserved: α, β, and δ.
3) X is set as the spatial coordinate component, the next position of grey wolf i under the guidance of grey wolf α is X_1, the distance between grey wolf α and prey is D_α, and the positions of the individual populations are updated according to Equations (1)–(3). The position formula of the grey wolf is as follows:

$$\begin{cases} D_\alpha = |C_1 \cdot X_\alpha - X| \\ D_\beta = |C_2 \cdot X_\beta - X| \\ D_\delta = |C_3 \cdot X_\delta - X| \end{cases} \tag{1}$$

$$\begin{cases} X_1 = X_\alpha - A \cdot D_\alpha \\ X_2 = X_\beta - A \cdot D_\beta \\ X_3 = X_\delta - A \cdot D_\delta \end{cases} \tag{2}$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{3}$$

4) Update a, A, and C according to Formula (4)-(6):

$$A = 2a \cdot r_1 - a \tag{4}$$

$$C = 2r_2 \tag{5}$$

$$a = 2 - 2t/M \tag{6}$$

5) Calculate the individual fitness of each grey wolf and update the positions of α, β, and δ.
6) Determine whether the stop condition is met: If yes, output the α-wolf optimization result. Otherwise, re-perform steps 3-6.

IV. IMPROVED GREY WOLF OPTIMIZATION ALGORITHM

A. HYPERBOLIC TANGENT INERTIA WEIGHT

The hyperbolic tangent inertia weight updating formula is computed as follows:

$$\omega = \frac{1}{\dfrac{e^{t/r} - e^{-t/r}}{e^{t/r} + e^{-t/r}} + b} + c \tag{7}$$

cArgument b is used to maintain the monotonicity of the function and define the range of values of the function. Parameter c was used to adjust the upper and lower bounds of the weights. r is related to the maximum iteration and takes a value in the range (0, M]. The parameter t increases with the number of iterations, with values in the range $[l, r]$, and $|l + r| \leq M$.

When the program needs to increase the global search capability of the algorithm, the function can be made to decrease more slowly in the early stages by decreasing the maximum and minimum values of t so that the inertia weights can be maintained for a more extended period at the beginning of the iteration. The concave shape of the inertia weight in the later iteration stage can increase the local search ability of the algorithm. The range of inertia weights can also be reduced by increasing the value of b: in the value of b gets larger, the maximum and minimum values of the function become closer to 0. The value of b cannot be less than one.

where $b = 1.8$, $c = -0.3$, $r = M/5$, t takes values in the range $[-M/2, M/2]$ starting from the minimum and increasing with iterations, as illustrated in Fig. 1(a). where $b = 1.2$, $c = -0.3$, $r = M$, t is the present number of iterations, as illustrated in Fig. 1(b).

As shown in Fig. 1(a), the curve with hyperbolic tangent weights is more prominent at the beginning of the iteration, which is conducive to expanding the search scope of the algorithm and preventing it from collapsing into the local optimum. As the number of iterations increases, the curve decreases until the iteration is complete, which is conducive to the continuous reduction in the inertia weight of the algorithm in the late iteration period and helps the algorithm approach the optimal solution more precisely in the later period.

Therefore, the optimal inertia weight-decreasing curve can be obtained by adjusting the parameters of the inertia weight

TABLE 1. Benchmark functions.

Name	Function	Dim	Interval	Global optimum
F1	$f_1(x)=\sum_{i=1}^{n} x_i^2$	30, 100, 500	[-100, 100]	0
F2	$f_2(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	30, 100, 500	[-100, 100]	0
F3	$f_3(x)=\max_i\{\|x_i\|,1\le i\le n\}$	30, 100, 500	[-10, 10]	0
F4	$f_4(x)=\sum_{i=1}^{n}\left[x_1^2-10\cos(2\pi x_i)+10\right]$	30, 100, 500	[-5.12, 5.12]	0
F5	$f_5(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\Pi_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	30, 100, 500	[-600, 600]	0

formula to balance the exploration and exploitation of different problems.

B. IMPROVED GREY WOLF OPTIMIZATION ALGORITHM

The steps of the HTGWO algorithm are as follows:

1) Initialization parameters: maximum number of iterations M, population size N, spatial population dimension dim, convergence factor a, and coefficient vectors A and C.
2) The fitness of the initial grey wolf individual was obtained according to the test function, and the three wolves with the best fitness were preserved:, α, β, and δ.
3) Update inertia weight ω according to Formula (7).
4) The updated position formula is shown in Formula 8:

$$\begin{cases} X_1 = \omega \cdot X_\alpha - A \cdot D_\alpha \\ X_2 = \omega \cdot X_\beta - A \cdot D_\beta \\ X_3 = \omega \cdot X_\delta - A \cdot D_\delta \end{cases} \tag{8}$$

5) Update parameters a, A, and C according to Formula (4)-(6).
6) Calculate the individual fitness of each grey wolf and update the positions of α, β, and δ.
7) Determine whether the stop condition is met: If yes, output the α Wolf optimization result. Otherwise, repeat Steps 3 to 6.

The HTGWO method enhances the inertia weight of the GWO algorithm using the hyperbolic tangent inertia weight.

V. EXPERIMENT AND RESULT

A. EXPERIMENTAL DESIGN

Equipment and software: Windows 10, 16 GB memory, main computer frequency of 2.3 GHz, MATLAB R2016a. Five classical test functions were selected to compare the optimization performance of the HTGWO with the other five classical swarm intelligent optimization algorithms. The five classic test functions The evaluation functions are listed in Table 1. To facilitate the experiment, the parameters used in Equation (7) are $b = 1.2$, $c = -0.3$, $M = 200$, $r = M$, and t is the present number of iterations.

The six algorithms were run 30 times in three different dimensional environments, with a population of 10 and a maximum number of iterations of 200 and boundary of ± 600. This study introduces the logarithmic mean value (MBFL) to calculate the current optimal adaptation value:

$$MBFL(t) = \frac{1}{G}\sum_{i=1}^{G}\log_{10}(gbest_{(i)}(t)) \tag{9}$$

where G is the maximum number of runs of the algorithm and is the current optimal value of running the t iteration for the i time.

B. EXPERIMENTAL RESULTS AND ANALYSIS

1) COMPARISON OF AVERAGE CONVERGENCE CURVES OF 6 ALGORITHMS

In Fig. 3–7, the vertical coordinate is the MBFL value, and the horizontal coordinate is the number of iterations. As shown in Fig. 3 to Fig. 7, HTGWO does not fall into the local optimum with the increase in iterations in low dimensions ($dim = 30$) and high dimensions ($dim = \{100, 500\}$). Additionally, the convergence speed and solving accuracy were superior to those of the other five algorithms. As shown in Fig. 3 to Fig. 5, PSO, FA, FOA, and GWO, all fall into local optima rapidly, whereas HTGWO approach the globally optimal solution faster. It can be seen from Fig. 6 and 7 that PSO, FA, and FOA rapidly fall into the global optima. Compared with PSO, FA, and FOA, GWO still falls into the local optimum with a higher convergence accuracy, and HTGWO finds the global optimal 0 at a faster speed.

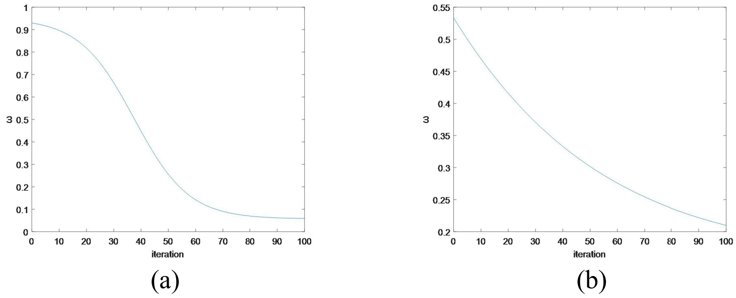

FIGURE 1. **Inertia weight decline curves.**

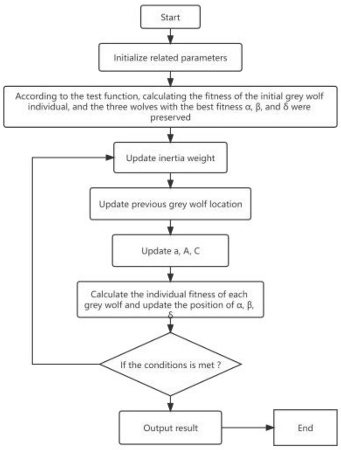

FIGURE 2. **Flow chart of HTGWO algorithm.**

Moreover, it can be observed from Fig. 3 Fig. 4 and Fig. 6 Fig. 7 that HTGWO performs well in handling unimodal function problems and effectively handles multimodal function problems. Moreover, the downward trend of the curve shown in Fig. 6 to Fig. 7 demonstrates that HTGWO can find the optimal global value of 0 when dealing with the optimal solution of multimodal functions. The HTGWO algorithm has a faster convergence speed and greater solving accuracy than the other five intelligent optimization algorithms.

In Fig. 7, the convergence curve of HTGWO converges faster in the early stage compared with the other algorithms. With the convergence of inertia weights, after 20 iterations, the focus from the global search to the local search starts to decrease. At approximately 30 iterations, the convergence curve starts to drop sharply, which means that the balancing strategy of HTGWO can locate the global optimum faster, and this phenomenon is almost the same in the three dimensions. The other algorithms are significantly less capable of exploring multimodal functions than the HTGWO.

2) COMPARISON BASED ON DESCRIPTIVE STATISTICS

The solution scales were 30, 100, and 500. The six algorithms were independently run 30 times, and the maximum, minimumi, average, and standard deviation of the obtained optimal adaptation values were compared, as shown in Table 2.

Table 2 illustrates the mean value and standard deviation of the optimal value independently run 30 times by six algorithms in the low dimension (dim = 30) and high dimension (dim = {100, 500}). As seen from Table 2, even though the value of HTGWO on the three unimodal test functions of F1, F2, and F3 does not get the global optimal value of 0, it can still obtain a result near the global optimal value. It is also an optimal solution with a higher precision than other algorithms. Furthermore, the average value of the optimal solution under the test of multimodal test functions F4 and F5 can converge to the optimal solution in three different dimensions. Compared with the other five algorithms, HTGWO has a minor standard deviation of the optimal solution in three dimensions, indicating that it has higher stability than the other five algorithms. Therefore, HTGWO has a faster convergence speed, higher solution accuracy, and higher stability than the other five intelligent optimization algorithms.

Fig. 8 shows a box plot of each function in three different dimensions for each box, with the center mark indicating the median, the top and bottom of the box indicating the 75th and 25th percentiles, respectively, and the dashed lines extending to the largest and smallest non-outliers indicated by dots along the way. As shown in Fig. 8, HTGWO outperforms the other four algorithms in all worst performance cases and converges smoothly without outliers, implying that the algorithm does not fall into a local optimum prematurely. In addition, HTGWO on the test function can provide superior mean, quartile, and minimum results compared with the other algorithms. The above results show that the HWGWO has an excellent exploration capability. Compared to other algorithms, HTGWO has almost no outliers, which means that it is more stable and competitive.

From the positions of the values and the maximum and minimum values in Fig. 8, it can be concluded that the HTGWO does not converge as fast as the algorithm on the multi-modal functions F4-F5 in the later stages of the unimodal functions F1-F3. Moreover, by comparing the location of the maximum value and the quartiles, the early convergence speed of the multimodal function is also faster than

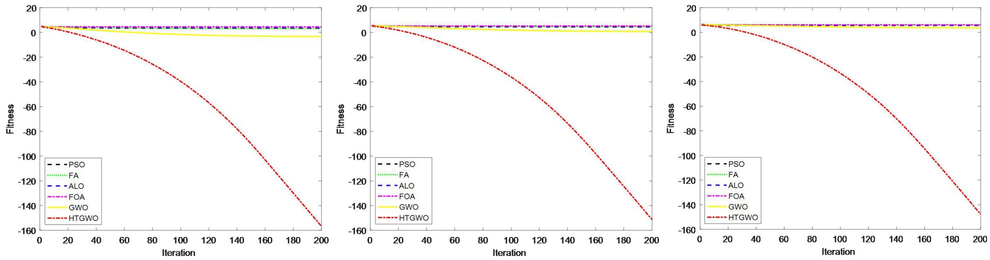

FIGURE 3. Average convergence curves of F1 (dim = {30, 100, 500}).

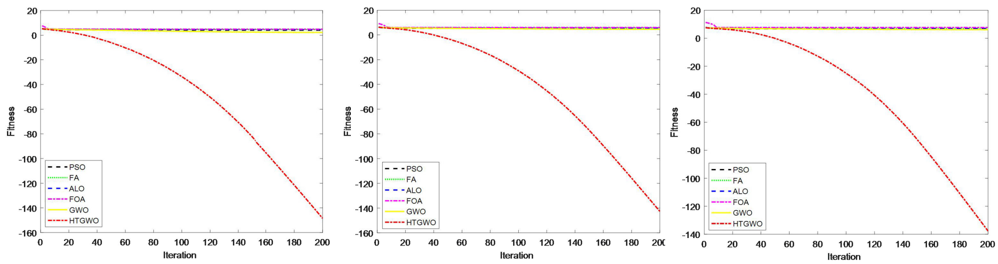

FIGURE 4. Average convergence curves of F2 (dim = {30, 100, 500}).

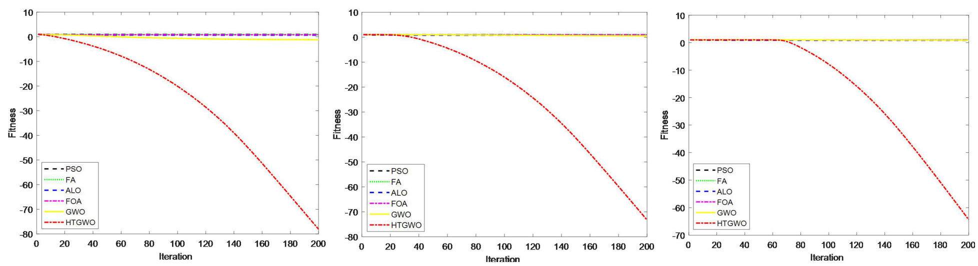

FIGURE 5. Average convergence curves of F3 (dim = {30, 100, 500}).

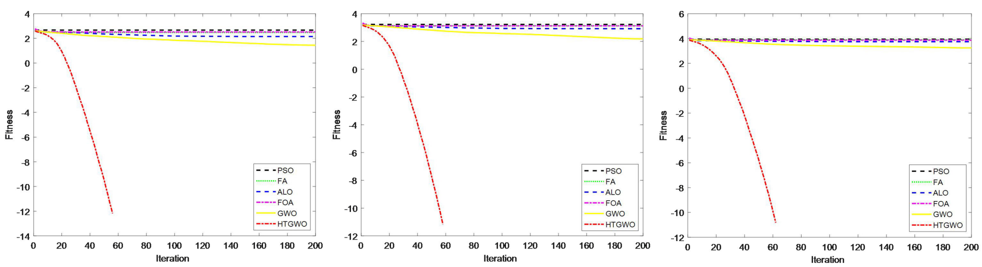

FIGURE 6. Average convergence curves of F4 (dim = {30, 100, 500}).

that of the unimodal function. In addition, the speed of convergence in the early part of the HTGWO iteration is less affected by the change in dimension, as can be seen by the amount of data before the median line in the box plots of the three different dimensions. Only on the unimodal function, F3 exhibits a higher number of values produced before the median. The change after the median indicates that the convergence speed in the middle and late iterations of the algorithm is less affected by the dimension.

In summary, the hyperbolic tangent inertia weight strategy adopted in this study has a positive impact on enhancing the optimization performance of the GWO algorithm. To a certain extent, it solves problems such as low convergence speed, inadequate solving accuracy, and insufficient stability of the GWO.

VI. CASE STUDY

A. HTGWO FOR OPTIMAL POWER FLOW PROBLEM

The optimal power flow (OPF) is a classical optimization problem in power systems, where the objective is to find an equilibrium point that minimizes the cost of power generation while satisfying all conditions and demands. It can be formulated as a nonlinear functional problem with the possibility of a local optima because the row domain is non-convex [32].

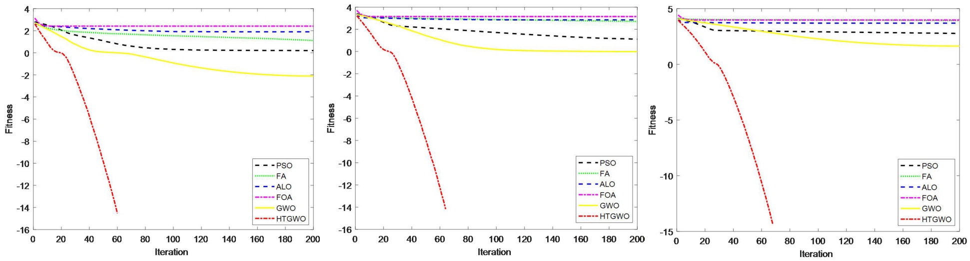

FIGURE 7. Average convergence curves of F5 (dim = {30, 100, 500}).

TABLE 2. Comparison of the mean value and standard deviation of the solution functions of the 6 algorithms.

Function	Algorithm	dim=30		dim=100		dim=500	
		mean	std	mean	std	mean	std
F1	ALO	1.02E+04	4.27E+03	8.18E+04	1.48E+04	5.70E+05	1.08E+05
	FA	1.96E+03	1.32E+03	4.96E+04	1.59E+04	9.26E+05	6.08E+04
	FOA	2.93E+04	4.52E+03	1.60E+05	1.23E+04	1.07E+06	2.55E+04
	GWO	7.19E-04	1.56E-03	5.66E+00	2.34E+00	4.52E+03	9.24E+02
	PSO	2.25E+03	3.01E+02	2.59E+04	3.37E+03	3.33E+05	3.32E+04
	HTGWO	**4.33E-156**	**2.02E-155**	**1.11E-151**	**2.48E-151**	**2.57E-147**	**8.30E-147**
F2	ALO	3.31E+04	1.10E+04	3.95E+05	1.41E+05	1.03E+07	3.79E+06
	FA	3.33E+04	1.48E+04	3.70E+05	1.06E+05	9.74E+06	3.91E+06
	FOA	6.51E+04	1.28E+04	9.36E+05	2.24E+05	5.96E+07	4.19E+07
	GWO	2.50E+02	2.71E+02	3.40E+04	1.18E+04	1.40E+06	1.88E+05
	PSO	1.15E+04	4.49E+03	1.51E+05	3.75E+04	3.98E+06	8.01E+05
	HTGWO	**1.92E-148**	**7.24E-148**	**1.33E-142**	**2.99E-142**	**6.05E-137**	**2.83E-136**
F3	ALO	3.87E+01	5.98E+00	5.62E+01	5.55E+00	6.91E+01	5.88E+00
	FA	7.60E+01	1.18E+01	9.61E+01	1.86E+00	9.93E+01	2.68E-01
	FOA	6.28E+01	4.32E+00	8.06E+01	2.07E+00	9.24E+01	2.32E+00
	GWO	9.18E-01	5.40E-01	3.57E+01	7.75E+00	7.99E+01	3.37E+00
	PSO	2.16E+01	2.96E+00	5.11E+01	5.84E+00	7.61E+01	4.12E+00
	HTGWO	**1.22E-77**	**1.94E-77**	**9.17E-72**	**4.66E-71**	**1.87E-63**	**2.85E-63**
F4	ALO	1.04E+04	3.38E+03	7.86E+04	1.37E+04	5.75E+05	1.13E+05
	FA	4.33E+03	2.85E+03	5.89E+04	1.67E+04	9.55E+05	8.03E+04
	FOA	3.13E+04	4.86E+03	1.61E+05	1.12E+04	1.06E+06	2.68E+04
	GWO	4.49E+01	1.51E+01	4.51E+02	6.56E+01	9.33E+03	1.03E+03
	PSO	2.54E+03	2.62E+02	2.53E+04	2.66E+03	3.39E+05	3.32E+04
	HTGWO	**0**	**0**	**0**	**0**	**0**	**0**
F5	ALO	3.12E+00	7.33E-01	2.12E+01	4.69E+00	1.40E+02	2.53E+01
	FA	1.60E+00	6.12E-01	1.43E+01	4.50E+00	2.37E+02	1.46E+01
	FOA	8.60E+00	1.16E+00	4.12E+01	2.65E+00	2.67E+02	7.96E+00
	GWO	3.06E-02	3.49E-02	1.52E-01	1.10E-01	2.16E+00	1.93E-01
	PSO	1.53E+00	7.94E-02	7.51E+00	5.58E-01	8.40E+01	8.17E+00
	HTGWO	**0**	**0**	**0**	**0**	**0**	**0**

The objective function of the OPF can be written as

$$\min \sum_{g\in G} f\left(p_g^G\right) \tag{10}$$

$$\text{subject to: } \sum_{g\in G} p_g^G = \sum_{d\in D_b} p_d^D + \sum_{b'\in B_b} p_{bb'}^L + G_b^B v_b^2 \tag{11}$$

G is the set of generators in the power-system network. Let D^b be the demands of bus b; let B_b be the set of b connected by a line (bb'); variable $p^L_{bb'}$ is the real power flowing into bb'; and parameter v_b is the voltage at bus band p^D_d is the real power consumed by load d. Adjust the parameters of the most appropriate inertia weight function for the problem

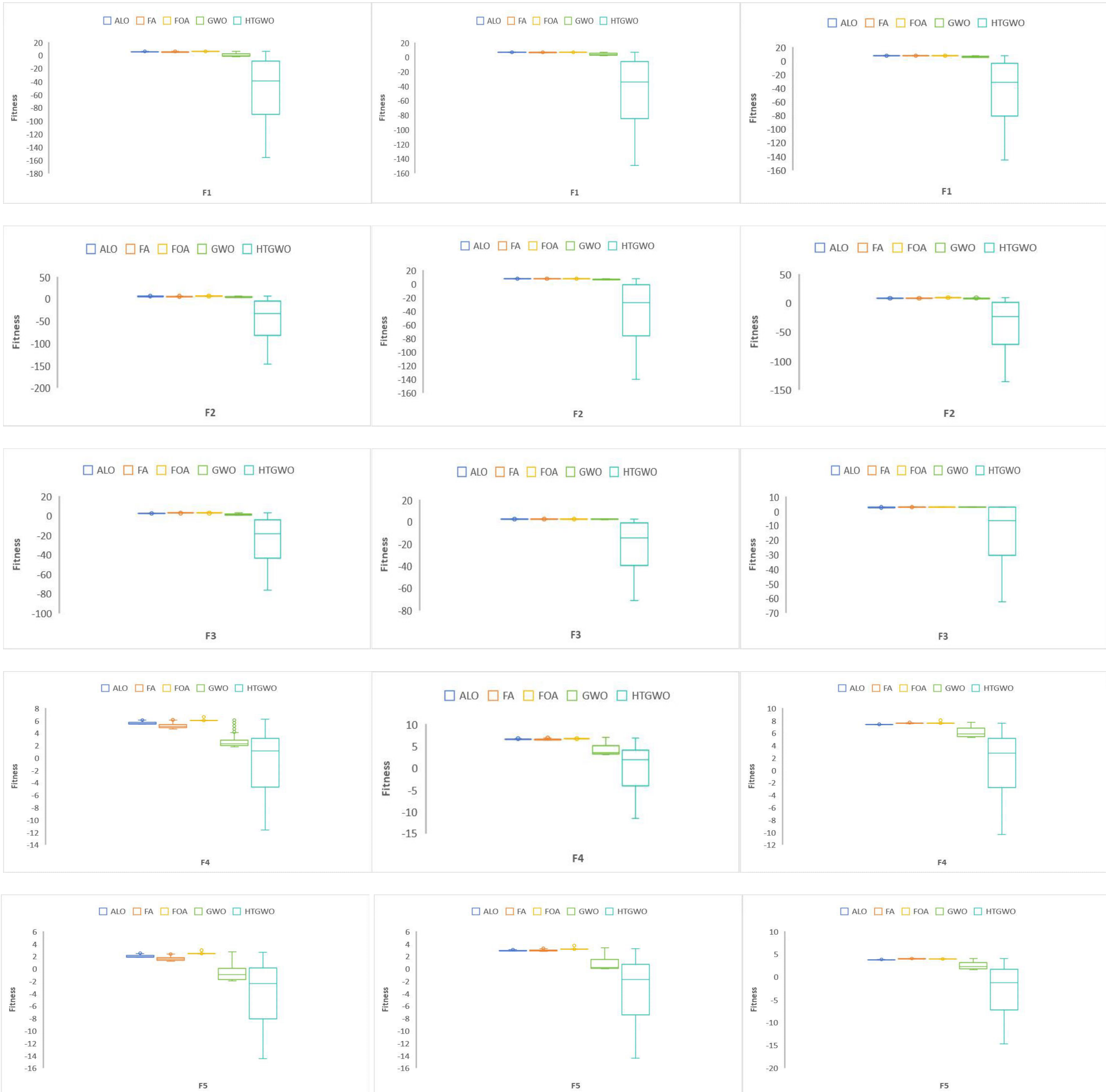

FIGURE 8. Box plots of benchmark functions (dim = {30, 100, 500}).

where $b = 2.2$, $c = -0.1$, $M = 1000$, $r = M/5$, and t is in the range $[-M/4,\ M/2 + M/4]$. Each test function was iterated 1000 times in a fixed dimension, and the results of the HTGWO and GWO are shown in Fig. 9. This shows that HTGWO outperforms GWO in terms of accuracy and convergence speed, and avoids local optima. In addition, its performance in this problem was better than GWO, which proves the accuracy and effectiveness of HTGWO in OPF problems for engineering applications. Regarding the OPF problem, designers can adjust the inertia weight parameters according to a specific problem, and this scheme can be used as a reference.

B. HTGWO FOR FREQUENCY OFFSET ESTIMATION

Orthogonal frequency division multiplexing (OFDM) has been widely used in wireless communication systems because of its effectiveness in enhancing the spectral robustness. However, OFDM is susceptible to carrier frequency offsets (CFOs), which affect the performance by affecting the signal-to-noise ratio (SNR). Therefore, CFO estimation has become the main technique for minimizing the frequency impact. The CFO estimation associated with GWO can minimize the necessity of center wavelength accuracy during laser transmission, thus enhancing and minimizing performance. The necessity of center wavelength accuracy during

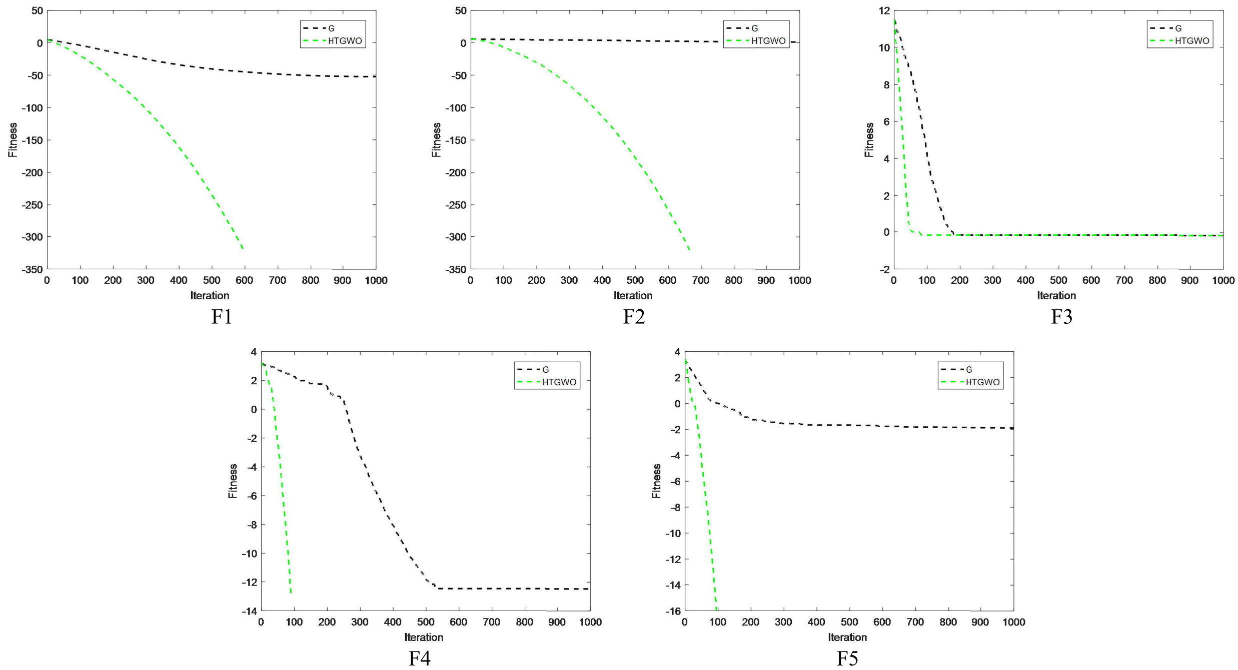

FIGURE 9. Convergence of HTGWO and GWO for optimal power flow problem.

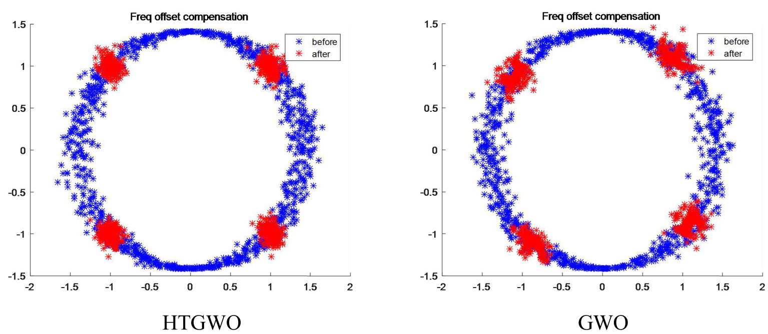

FIGURE 10. Frequency offset compensation by HTGWO and GWO.

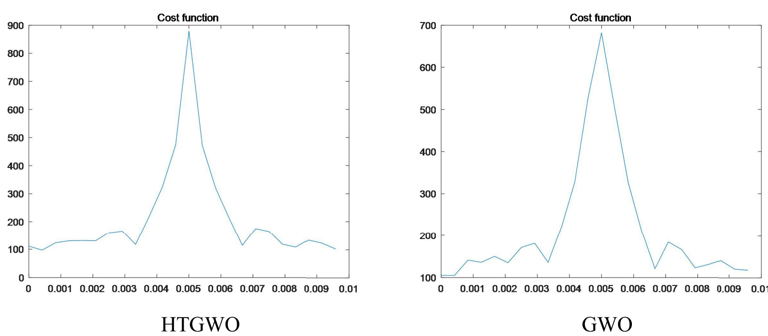

FIGURE 11. Best score of HTGWO and GWO for frequency offset compensation.

laser transmission enhances the performance of OFDM systems [33]. The signal was assumed to have uncompensated frequency detuning. Equation (11) is the cost function that generates a histogram from the complex phase values of the signal multiplied by a complex sine with frequency W0. This metric is the difference between the sum of the four largest histogram cells and the four smallest cells. This case aims to optimize the value of w_0. j is the population, t is the number of iterations, and p is the parameter associated with SNR.

$$C\text{data} = C * \exp(j * w_0 * t) + p \tag{12}$$

$$SNR(f_e) \geq \frac{SNR}{1 + 0.5947 SNR \sin^2 \pi f_e} \left(\frac{\sin \pi f_e}{\pi f_e} \right)^2 \tag{13}$$

$$p = 10^{(-SNR/20)} \tag{14}$$

For this problem, the parameters of the inertia weights are set as $b = 1.2$, $c = -0.3$, $M = 1000$ $r = M/5$, and t takes values in the range $[-M/2, M/2]$. HTGWO and GWO are used to compensate for the frequency dispersion, and the results, as shown in Fig. 10, indicate that HTGWO is significantly better than GWO in compensating for the frequency dispersion and has fewer dispersion points. The best score for HTGWO was higher than that for GWO (Fig. 11). From the above two points, it is clear that HTGWO is more effective than GWO in compensating for the frequency of the CFO problems. An algorithm can be used to solve this problem, and the exact parameter design depends on the problem requirements.

VII. CONCLUSION

This paper proposes HTGWO to address the slow convergence problem and local optimization of GWO. The HTGWO utilizes the characteristics of hyperbolic tangent inertia weights to improve the inertia weights of the GWO. HTGWO has significantly improved its global search ability and local search ability, mainly because its search efficiency is better than that of the other algorithms. Experiments demonstrate that HTGWO has the characteristics of faster convergence, higher accuracy, and higher stability than other algorithms. In the application, the algorithm is simple to implement in programming, which is convenient for subsequent research. Moreover, the characteristic of HTGWO is that the inertia weight strategy can be changed according to specific problems.

Different problems require manual adjustment of the test parameters suitable for the problem of the parameter strategy, which is the main limitation of the method. Therefore, in future work, developing methods that can adaptively adjust parameters is the main direction, given that the method also has relatively good performance in engineering applications. Therefore, the adaptive adjustment of parameters should be combined with practical engineering problems to increase the ability of the method to solve different engineering problems.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[2] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[3] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Frome, U.K.: Luniver Press, 2010.

[4] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[6] Y. Qiao, H. Y. Hsu, and J. S. Pan, "Behaviour-based grey wolf optimiser for a wireless sensor network deployment problem," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 39, nos. 1–2, p. 70, 2022, doi: 10.1504/IJAHUC.2022.120947.

[7] D. Miao, W. Chen, W. Zhao, and T. Demsas, "Parameter estimation of PEM fuel cells employing the hybrid grey wolf optimization method," *Energy*, vol. 193, Feb. 2020, Art. no. 116616, doi: 10.1016/j.energy.2019.116616.

[8] W. Fang, W. Zhang, C. Zhang, Z. Xie, and T. Yu, "Separation method of main and foreign water for the measuring weirs of Danjiangkou Earth-rock dam," *Water*, vol. 14, no. 22, p. 3620, Nov. 2022, doi: 10.3390/w14223620.

[9] X. Lei and H. Ouyang, "Kernel-based intuitionistic fuzzy clustering image segmentation based on grey wolf optimizer with differential mutation," *IEEE Access*, vol. 9, pp. 85455–85463, 2021.

[10] V. Kumar, K. Kalita, S. Madhu, U. Ragavendran, and X.-Z. Gao, "A hybrid genetic programming–gray wolf optimizer approach for process optimization of biodiesel production," *Processes*, vol. 9, no. 3, p. 442, Mar. 2021, doi: 10.3390/pr9030442.

[11] G. Huang, Y. Cai, J. Liu, Y. Qi, and X. Liu, "A novel hybrid discrete grey wolf optimizer algorithm for multi-UAV path planning," *J. Intell. Robot. Syst.*, vol. 103, no. 3, pp. 1–18, Nov. 2021, doi: 10.1007/s10846-021-01490-3.

[12] Z. Li, Y. He, Y. Li, and X. Guo, "A hybrid grey wolf optimizer for solving the product knapsack problem," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 1, pp. 201–222, Jan. 2021.

[13] J. Liu, X. Wei, and H. Huang, "An improved grey wolf optimization algorithm and its application in path planning," *IEEE Access*, vol. 9, pp. 121944–121956, 2021, doi: 10.1109/ACCESS.2021.3108973.

[14] X. Zhang, Y. Zhang, and Z. Ming, "Improved dynamic grey wolf optimizer," *Frontiers Inf. Technol. Electron. Eng.*, vol. 22, no. 6, pp. 877–890, Jun. 2021, doi: 10.1631/fitee.2000191.

[15] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo, and W. Xie, "An improved grey wolf optimizer based on tracking and seeking modes to solve function optimization problems," *IEEE Access*, vol. 8, pp. 69861–69893, 2020, doi: 10.1109/ACCESS.2020.2984321.

[16] N. Singh, "A modified variant of grey wolf optimizer," *Scientia Iranica*, vol. 27, no. 3, pp. 1450–1466, 2018.

[17] H. Qin, T. Meng, and Y. Cao, "Fuzzy strategy grey wolf optimizer for complex multimodal optimization problems," *Sensors*, vol. 22, no. 17, p. 6420, Aug. 2022, doi: 10.3390/s22176420.

[18] L. R. Rodrigues, "A chaotic grey wolf optimizer for constrained optimization problems," *Expert Syst.*, vol. 40, no. 4, p. e12719, May 2023, doi: 10.1111/exsy.12719.

[19] J. C. Bansal and S. Singh, "A better exploration strategy in grey wolf optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1099–1118, Jan. 2021, doi: 10.1007/s12652-020-02153-1.

[20] L. Sun, B. Feng, T. Chen, D. Zhao, and Y. Xin, "Equalized grey wolf optimizer with refraction opposite learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–18, May 2022, doi: 10.1155/2022/2721490.

[21] Y. Ou, P. Yin, and L. Mo, "An improved grey wolf optimizer and its application in robot path planning," *Biomimetics*, vol. 8, no. 1, p. 84, Feb. 2023, doi: 10.3390/biomimetics8010084.

[22] M. H. Nadimi-Shahraki, E. Moeini, S. Taghian, and S. Mirjalili, "Discrete improved grey wolf optimizer for community detection," *J. Bionic Eng.*, vol. 20, no. 5, pp. 2331–2358, Sep. 2023, doi: 10.1007/s42235-023-00387-1.

[23] K. Meidani, A. Hemmasian, S. Mirjalili, and A. B. Farimani, "Adaptive grey wolf optimizer," *Neural Comput. Appl.*, vol. 34, no. 10, pp. 7711–7731, May 2022, doi: 10.1007/s00521-021-06885-9.

[24] C. Ma, H. Huang, Q. Fan, J. Wei, Y. Du, and W. Gao, "Grey wolf optimizer based on Aquila exploration method," *Expert Syst. Appl.*, vol. 205, Nov. 2022, Art. no. 117629, doi: 10.1016/j.eswa.2022.117629.

[25] X. Liang, D. Wang, and M. Huang, "Improved grey wolf optimizer and their applications," in *Proc. IEEE 7th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dalian, China, Oct. 2019, pp. 107–110, doi: 10.1109/ICCSNT47585.2019.8962504.

[26] S.-Y. Li, S.-M. Wang, P.-F. Wang, X.-L. Su, X.-S. Zhang, and Z.-H. Dong, "An improved grey wolf optimizer algorithm for the inversion of geoelectrical data," *Acta Geophysica*, vol. 66, no. 4, pp. 607–621, Aug. 2018, doi: 10.1007/s11600-018-0148-8.

[27] Y. Jieran, W. Aimin, G. Yan, and S. Xinyi, "An improved grey wolf optimizer for flexible job-shop scheduling problem," in *Proc. IEEE 11th Int. Conf. Mech. Intell. Manuf. Technol. (ICMIMT)*, Cape town, South Africa, Jan. 2020, pp. 213–217, doi: 10.1109/ICMIMT49010.2020.9041184.

[28] A. A. Musa, S. H. Imam, A. Choudhary, and A. P. Agrawal, "Parameter estimation of software reliability growth models: A comparison between grey wolf optimizer and improved grey wolf optimizer," in *Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Noida, India, Jan. 2021, pp. 611–617, doi: 10.1109/Confluence51648.2021.9377194.

[29] A. A. A. El-Ela, R. A. El-Seheimy, A. M. Shaheen, and I. A. Eissa, "Optimal allocation of DGs and capacitor devices using improved grey wolf optimizer," in *Proc. 21st Int. Middle East Power Syst. Conf. (MEPCON)*, Cairo, Egypt, Dec. 2019, pp. 441–446, doi: 10.1109/MEPCON47431.2019.9008228.

[30] Y. Shu, Z. Shen, L. Xu, J. Duan, L. Ju, and Q. Liu, "Inverse modeling of seepage parameters based on an improved gray wolf optimizer," *Appl. Sci.*, vol. 12, no. 17, p. 8519, Aug. 2022, doi: 10.3390/app12178519.

[31] P. Hu, S. Chen, H. Huang, G. Zhang, and L. Liu, "Improved alpha-guided grey wolf optimizer," *IEEE Access*, vol. 7, pp. 5421–5437, 2019, doi: 10.1109/ACCESS.2018.2889816.

[32] W. A. Bukhsh, A. Grothey, K. I. M. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, Nov. 2013, doi: 10.1109/TPWRS.2013.2274577.

[33] N. Rashmi and M. Sarvagya, "Self-improved grey wolf optimization for estimating carrier frequency offset in SCM-OFDM systems," *Int. J. Pervas. Comput. Commun.*, vol. 16, no. 1, pp. 53–73, Jan. 2020, doi: 10.1108/ijpcc-03-2019-0020.

WEIMING LIN received the M.S. degree from the University of Liverpool. His current research interests include machine learning and computer science.