An Improved Hybrid Grey Wolf Optimizer for Multi-agent Trajectory Planning in Complex Environment

Yutong Zhu, Ye Zhang, Jingyu Wang, Ke Zhang

***Abstract*—This paper considers the problem that Grey Wolf Optimizer (GWO) has some defects in solving trajectory optimization problems. To solve this problem, this paper proposes the improved GWO algorithm based on GWO by the idea of linear differential decrement and dynamic exponential weighted average. Compared with other algorithms, this algorithm has more flexibility in position updating and finds the global optimal solution effectively. Finally, simulation results demonstrate the superiority of the improved GWO algorithm in terms of search accuracy and running time.**

***Index Terms*—Grey wolf optimizer, trajectory optimization, linear differential decrement, dynamic exponentially weighted average**

I. Introduction

When confronted with various optimization challenges, researchers frequently opt for optimization algorithms to seek the best possible solution for a given problem. In contrast, bionic algorithms rely on intuitive or empirical principles, offering feasible solutions to problems at an acceptable cost (such as computational time or memory usage). However, these algorithms don't ensure an optimal solution; instead, they provide practical outcomes that meet the criteria within acceptable computational constraints. GWO is a bionic intelligent optimization algorithm [1]. This algorithm simulates the predation strategy and hierarchy of grey wolves in nature. GWO has been successfully applied to many engineering optimization problems [2]. Several variations of GWO have emerged, aiming to circumvent local optima and expedite convergence by altering the fundamental mechanism of the original GWO algorithm. [3] introduced mGWO, incorporating a nonlinear control parameter strategy to achieve a balanced exploration-exploitation approach within GWO. While inheriting the movement strategy from GWO, mGWO aims to mitigate the risk of getting stuck in local optima and experiencing premature convergence. In a separate study by [4], the Quasi-Oppositional based Learning (Q-OBL) theory was integrated into the conventional GWO, resulting in the QOGWO variant, aiming to enhance its performance. [5] proposed EEGWO, modifying the position updating mechanism of the algorithm to address its limitations. Nevertheless, concerns regarding trapping in local optima and premature convergence persisted despite these adjustments. Addressing the algorithm's persisting issues, [6] introduced ROL-GWO, where a modified parameter C was employed to augment the exploration capabilities of the algorithm. However, due to the structure of GWO and the characteristics of the bionic algorithm [7], GWO can ultimately only generate near-optimal solutions. As a result, it is easy to fall into a local optimum during its iterative process and has the disadvantages of poor exploitation capability and inflexible update strategies. To address these drawbacks, this paper proposes a combined improved GWO, called **L**inear **D**ifferential **D**ynamical **W**eighted **GWO** (LDDWGWO).

Linear differential decrement strategy (LDD) is a mathematical method used in recent years to explore new algorithms [8]. LDD enables the variable selection of parameters in GWO, with curvature changes in the initial and terminal phases leading to large changes in future update positions. Exponentially weighted averaging accelerates the efficiency of algorithm iteration [9]. Inspired by [10], [11], this paper proposes Dynamic Exponentially Weighted Averaging (DEWA) on this basis, which improves the inflexibility of the original GWO update to increase the diversity of solutions, while the convergence speed is improved.

The aim of this paper is to improve GWO by combining LDD and DEWA to produce solutions with flexible location updates and to reduce the likelihood of solutions falling into local optima.

(i) LDD serves the purpose of dynamically updating parameters within GWO, while DEWA operates on updating individual positions. DEWA's role is crucial in augmenting solution diversity, aiding in escaping local optima and steering the optimization process towards the global optimum.

(ii) When employing LDDWGWO in path planning problems susceptible to getting trapped in local optima, comparative evaluations against other optimization algorithms reveal distinctions in mean value, best value, p-value, convergence curve, and runtime performance.

*This work was supported by the Guangdong Basic and Applied Basic Research Foundation under the Grant No. 2021A1515110569, the National Natural Science Foundation of China under the Grant No. 52202502 and Fundamental Research Funds for the Central Universities under the Grant No. G2021KY05109. (*Corresponding author: Ye Zhang.*)

Yutong Zhu is with the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: yutong.zhu@mail.nwpu.edu.cn).

Ye Zhang is with the Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen, 518063, China; the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhang_ye@nwpu.edu.cn).

Jingyu Wang is with the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: jywang@nwpu.edu.cn).

Ke Zhang is with the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhangke@nwpu.edu.cn).

II. Preliminaries and problem description

A. Problem description

The problem of global optimal of the optimization algorithm is studied. This paper introduces and establishes the following definitions:

Finding X which optimizes $f(X)$

subject to:

$$\begin{aligned} &g_i(X) \leq 0, \quad i = 1, \cdots, n \\ &h_j(X) = 0, \quad j = 1, \cdots, q \\ &lb_i \leq X_i \leq ub_i, \quad i = 1, \cdots, d \end{aligned} \tag{1}$$

where n represents the count of inequality constraints while q stands for the number of equation constraints. The variable X denotes the solution vector, with d representing the dimensionality of the variables. Additionally, lb_i and ub_i signify the lower and upper bounds, respectively, of the variables.

However, due to the nature of GWO's update mechanism and meta-heuristic algorithms, GWO updates are inflexible and ultimately only near-optimal solutions are created. So its process easily converges to a local optimum, thereby reducing the diversity of understanding and halting its exploration [12].

Despite its strong optimization capabilities compared to other optimization algorithms, GWO still has a small probability of global non-convergence. We now describe the problem to be solved in this paper. GWO is improved by combining LDD and DEWA to produce solutions with flexible location updates and to reduce the likelihood of solutions falling into local optima.

B. Model of GWO

Within GWO, the algorithm mimics the predatory behavior and hierarchical structure observed in grey wolves in nature. In this context, α spearheads the predatory actions, β assists in decision-making, δ handles specific action arrangements, while ω follows the directives of the first three, encircling and ultimately executing the predatory action to secure the prey.

In GWO, when addressing an optimization problem, the population comprises n grey wolf individuals navigating a **D**-dimensional search space. Each grey wolf's position, denoted as $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$, represents their location in this multi-dimensional space. Within this population, α signifies the current best individual, β represents the second-best, δ denotes the third-best, while the rest are designated as ω. The prey's position aligns with the optimal solution of the optimization problem. GWO process unfolds as follows: an initial group of grey wolves is randomly generated across the search space. The fitness evaluation of the top three individuals, α, β, and δ, serves as the benchmark for identifying the prey's location (optimal solution). Subsequently, the positions of the next generation of grey wolves are computed based on these top individuals' locations, driving the optimization process forward. Refer to Fig. 1 for an illustrative depiction of this enclosure.

GWO has observation mechanism and hunting mechanism. In observation mechanism the grey wolf population must first take encirclement action on its prey. The distance between individuals and prey must be determined in the optimal search process of GWO, and we can get

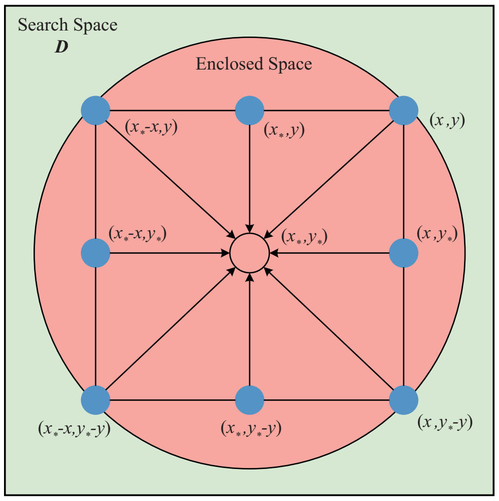


Fig. 1. Diagram of GWO enclosing strategy. Suppose the prey position is (x_*, y_*) and the grey wolf position is (x, y). At $A = (1, 0)$ and $C = (1, 1)$, the grey wolf will move to $(x_* - x, y_*)$ according to the formula based on the prey position, so the enclosing effect will be produced when the vectors A and C take different values.

$$D_i(t) = |C \cdot X_*(t) - X_i(t)| \tag{2}$$

$$X_i(t+1) = X_*(t) - A \cdot D_i(t) \tag{3}$$

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

$$C = 2 \cdot r \tag{5}$$

where $D_i(t)$ is the individual's distance from i and $X_*(t)$ at time $t(1 < t < \hat{t})$. $\hat{t}$ represents the number of iterations undergone. The notation $X_*(t)$ represents the position vector of the prey at time t, while $X_i(t)$ denotes the position vector of an individual grey wolf at the same time point. The coefficients A and C are involved in the calculations. a decreases linearly from 2 to 0 with the number of iterations and r is a random number between $[0, 1]$.

$$a = 2 - \frac{2}{t_{\max}} \cdot t \tag{6}$$

where $t_{\max}$ is the max number of iterations. This is a typical linear decrement strategy.

In the hunting mechanism, for the other wolves in the population, the position of prey based on the position of the top three grey wolves α, β and δ individuals, therefore

$$\begin{cases} D_\alpha = |C_1 \cdot X_\alpha - X_i| \\ D_\beta = |C_2 \cdot X_\beta - X_i| \\ D_\delta = |C_3 \cdot X_\delta - X_i| \end{cases} \tag{7}$$

$$\begin{cases} X_{i1} = X_\alpha - A_1 D_\alpha \\ X_{i2} = X_\beta - A_2 D_\beta \\ X_{i3} = X_\delta - A_3 D_\delta \end{cases} \tag{8}$$

then the position that grey wolf i will move to can be calculated from Eq. (9)

$$X_i(t+1) = \frac{X_{i1}(t) + X_{i2}(t) + X_{i3}(t)}{3} \tag{9}$$

Although GWO has the advantages of fewer control parameters and a certain degree of avoidance of falling into local optimal, GWO also has certain drawbacks in solving trajectory optimization problems. When GWO randomly initializes individuals, the path points generated are haphazard and disorderly, resulting in a relatively low fitness of individuals. Because the distance control parameter a is linearly decrement, the exploitation capability is insufficient. When the position is updated, GWO uses an average calculation based on the top three individuals in terms of fitness value, which leads to an inflexible position update strategy and makes it difficult to find the global optimal solution by stepping out of the local optimum 100% of the time in some optimization search processes.

III. A combined improved GWO algorithm

The original GWO converges slowly and readily yields a locally optimal solution. We propose LDDWGWO algorithm combining a linear differential decrement strategy with a dynamic exponentially weighted average. LDDWGWO increases the unpredictability of updating search agent positions and the speed of iterative convergence, allowing it to reach every corner of the search space in as short a time as possible.

A. Linear differential decrement strategy

As the GWO progresses through iterations, the value of parameter a gradually diminishes, resulting in a gradual slowdown in the search velocity. Consequently, GWO exhibits improved local search prowess while its capacity for global exploration weakens. From Eq. (6), the slope is constant, so the change in search speed always remains at the same level. When the initial iterations fail to yield superior points, the accumulation of subsequent iterations coupled with the swift decline in search velocity often culminate in converging towards a local optimum towards the algorithm's conclusion. Based on the classical linear decrement strategy for a, we introduce differential equations to construct a new decrement strategy, then we have

$$\frac{da}{dt} = \frac{2 \cdot 2}{t_{\max}^2} \cdot t \tag{10}$$

$$\int_a^2 da = \frac{2 \cdot 2}{t_{\max}^2} \int_0^t \tau \, d\tau \tag{11}$$

$$a = 2 - \frac{2}{t_{\max}^2} \cdot t^2 \tag{12}$$

It can be seen from Eq. (12) that a and t are still negatively correlated and that a is a quadratic function of t. Fig. 2 shows that a changes slowly during the initial iterations, which is conducive to finding a local optimum that satisfies the conditions during the initial iterations. a changes faster as

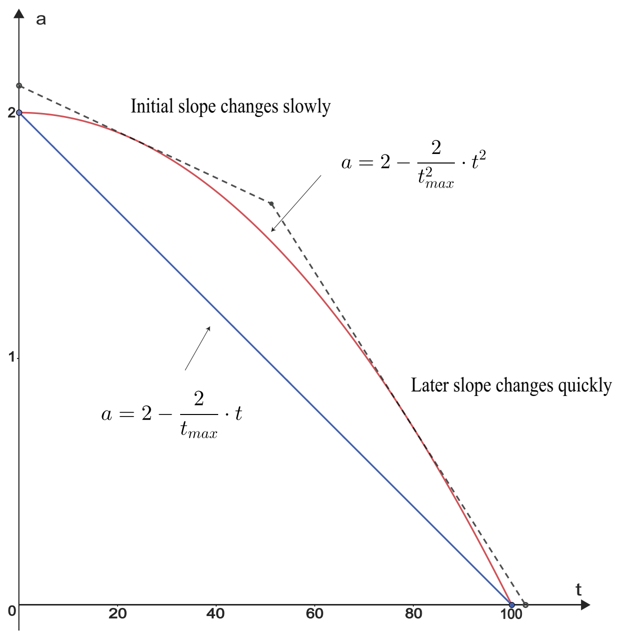


Fig. 2. Plot of iteration curves under classical linear decrement and linear differential decrement strategies. In the case of 100 iterations, a initially changes slowly, searching for a local optimum that satisfies the conditions. Later changes are faster, and after finding the local optimum is able to converge quickly to approximate the global optimum.

it approaches the maximum number of iterations, and can converge quickly to approximate the global optimum after finding the local optimum, improving the efficiency of the operation.

B. Dynamic exponentially weighted average

In order to overcome the shortcomings of inflexible location update and slow convergence in GWO, inspired by [10], [11], we construct a dynamic exponentially weighted average (DEWA) location update strategy. DEWA intervention threshold with the following update method

$$Y_i(t+1) = \mu Y_i(t) + (1-\mu) X_i(t+1) \tag{13}$$

where $Y_i(t+1)$ represents the mean value at time $t+1$. $Y_i(0)$ is equal to 0. μ is the adaptively adjustable weight, which generally ranges from $(0,1)$. To improve the accuracy of the exponentially weighted average, a correction can be made by the correction formula

$$Z_i(t+1) = \frac{Y_i(t+1)}{1-\mu^{t+1}} \tag{14}$$

where $Z_i(t+1)$ is the deviation-corrected value. The next moment individual position is dynamically adjusted according to the fitness value of the optimal three individuals, with the following update formula

$$\sigma = \varepsilon \cdot a \tag{15}$$

$$\varphi_\gamma = \varphi_\alpha + \varphi_\beta + \varphi_\delta \tag{16}$$

$$X_i(t+1) = \begin{cases} \dfrac{\varphi_\alpha X_{i1}(t) + \varphi_\beta X_{i2}(t) + \varphi_\delta X_{i3}(t)}{\varphi_\gamma}, \\ |X_{i1}(t) - X_{i3}(t)| > \sigma \\ \dfrac{X_{i1}(t) + X_{i2}(t) + X_{i3}(t)}{3}, \\ |X_{i1}(t) - X_{i3}(t)| \leq \sigma \end{cases} \tag{17}$$

where σ is the threshold value. ε is the dynamic weighted intervention scale factor. φ_α, φ_β, φ_δ is the fitness value of α, β, δ individuals. The algorithm's merit-seeking capability is further enhanced as the position is continuously updated. Algorithm 1 presents the overall process for LDDWGWO.

Algorithm 1 LDDWGWO algorithm

```
1: Initialize the population of whales $X_i (i = 1, 2, \cdots, n)$
2: Initialize the dynamic weighted intervention scale factor
   $\varepsilon$ and a random number $r$
3: Calculate the fitness of each search agent
4: while $t \leq \hat{t}$ do
5:     for each search agent do
6:         Update $a$ with Eq. (12)
7:         Update $A$, $C$, $\sigma$
8:         Compare the fitness of each search agent and filter
           out the top three best individuals $X_\alpha$, $X_\beta$, $X_\delta$
9:         if $|X_{i1}(t) - X_{i3}(t)| \leq \sigma$ then
10:            For each search agent in the population, update its
               position with Eq. (9)
11:        else
12:            Update its position with Eq. (17)
13:        end if
14:        Adaptive adjustment weight $\mu$ and modify individual
           position with Eq. (14)
15:        Check if any search agent goes beyond the search
           space and amend it
16:        Calculate the fitness value of each search agent
17:    end for
18:    $t \leftarrow t + 1$
19: end while
20: return
```

IV. Simulation

A. Basic setting

To verify the effectiveness and superiority of LDDWGWO, we combine it with the classical artificial potential field (APF) algorithm and apply it to the path planning problem. The local optimum problem is the most typical problem for the APF algorithm in Refs. [13], [14]. In addition, APF can provide a 3D simulation scenario. This therefore provides a good validation for reducing the possibility of the solution falling into a local optimum. We have carried out extensive simulation tests and report the results of this part.

The configuration of the simulation experiments is described below: in a 100km $\times$ 100km $\times$ 20km $x-y-z$ space, the basic setup of the UAV and obstacle positions is shown in Fig. 3, where the obstacle influence radius $X \in [4, 18]$ and $Y \in [5, 14]$. A quadrotor UAV model is used in this paper. The UAV has a minimum speed of 50 meters per second, a minimum turning radius of 2 kilometers and a minimum flight altitude of 1 kilometer. In the multi-UAV path planning problem, the minimum distance between UAVs is $[50, 50, 20]^T$ meters. Based on the simulation scenario, the initial position of UAVs, the location and range of influence of the obstacles, and the location and orientation of the targets are configured. The following assumptions are imposed in order to address the above problem.

Assumption 1. Do not consider mutual collisions between UAVs.

Assumption 2. The continuous solution space of infinite states maps to a finite discrete set.

Assumption 3. All UAV models are considered as particles.

To evaluate the performance of LDDWGWO comprehensively, a mixed static and dynamic obstacle scenario is created with different types of fitness functions, including unimodal functions $(f_1 - f_6)$ and multimodal functions $(f_7 - f_9)$ [15], [16] in Table I. The performance of the proposed LDDWGWO is tested on 10 fitness benchmark functions and compares with the original GWO [1] and 2 other optimization algorithms. This incorporates both particle swarm optimization (PSO) [17] and genetic algorithms (GA) [18].

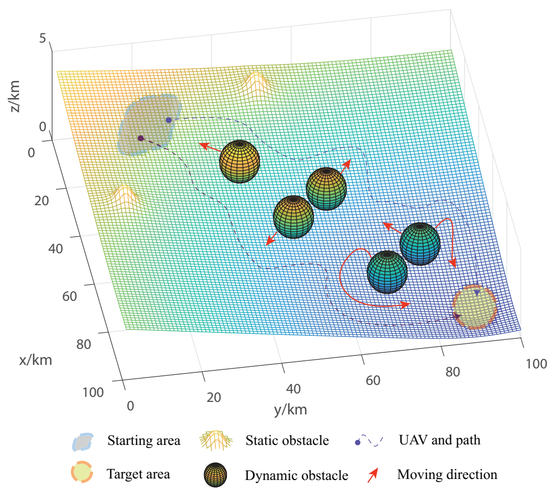


Fig. 3. Schematic diagram of the basic setup of UAVs and obstacles positions

The number of iterations for each algorithm is 100 and the dimension is 30. Four evaluation metrics such as the mean value, the best one, the std and the p-value are calculated as shown in Table II.

B. Convergence analysis

As can be seen from Table II, most of the results of LDDWGWO outperformed those of other similar products. The results of the convergence curves for the four algorithms for the nine benchmark functions are shown in Fig. 4. The

TABLE I
DESCRIPTION OF BENCHMARK FUNCTIONS

Function	Dim	Iteration	Equation
Sphere	30	100	$f_1(X) = \sum_{i=1}^{n} X_i^2$
Sumsquares	30	100	$f_2(X) = \sum_{i=1}^{n} iX_i^2$
Step	30	100	$f_3(X) = \sum_{i=1}^{n} [X_i + 0.5]^2$
Quartic	30	100	$f_4(X) = \sum_{i=1}^{n} iX_i^4 + \text{random}[0,1)$
Rosenbrock	30	100	$f_5(X) = \sum_{i=1}^{n-1} [100(X_{i+1} - X_i^2) + (X_i - 1)^2]$
Schwefel 2.22	30	100	$f_6(X) = \sum_{i=1}^{n} \lvert X_i \rvert + \prod_{i=1}^{n} \lvert X_i \rvert$
Rastrigin	30	100	$f_7(X) = \sum_{i=1}^{n} [X_i^2 - 10\cos(2\pi X_i) + 10]$
Griewank	30	100	$f_8(X) = \sum_{i=1}^{n} \frac{X_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{X_i}{\sqrt{i}}) + 1$
Ackley	30	100	$f_9(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} X_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi X_i)\right) + 20 + e$

TABLE II
COMPARISON OF RESULTS (MEAN, BEST, STD, p-VALUE) FOR UNIMODAL AND MULTIMODAL BENCHMARK FUNCTIONS

Function	Metrics	PSO	GA	GWO	LDDWGWO
f_1	Mean	9.98E−01	1.67E+01	9.53E−03	**3.04E−04**
	Best	1.51E−01	9.18E+00	2.63E−03	**3.39E−05**
	Std	1.16E+00	5.27E+00	2.55E−02	**5.28E−04**
	p-value	3.02E−11	7.27E−08	9.33E−05	
f_2	Mean	5.28E−03	3.28E+00	4.38E−02	**5.26E−05**
	Best	3.53E−04	2.57E−01	3.69E−04	**4.92E−08**
	Std	7.29E−03	9.43E+00	6.53E−02	**7.46E−05**
	p-value	3.78E−08	5.03E−05	3.97E−09	
f_3	Mean	8.27E−01	2.07E+00	6.52E−03	**1.42E−03**
	Best	7.63E−02	9.13E−01	5.42E−04	**2.31E−04**
	Std	3.28E+00	8.31E+00	9.75E−03	**3.96E−03**
	p-value	3.55E−11	5.68E−08	7.31E−04	
f_4	Mean	3.07E−02	9.21E−01	2.98E−03	**2.53E−03**
	Best	7.73E−03	2.37E−01	**3.82E−05**	5.23E−05
	Std	6.63E−02	3.78E+00	6.43E−03	**4.05E−03**
	p-value	9.42E−14	3.57E−07	6.24E−11	
f_5	Mean	5.24E−01	3.15E+00	2.13E−02	**2.31E−03**
	Best	6.73E−03	2.21E−01	7.98E−04	**9.25E−05**
	Std	7.84E−01	7.25E+00	5.65E−02	**4.73E−03**
	p-value	9.20E−12	5.32E−07	2.97E−18	
f_6	Mean	3.24E−03	7.21E−01	4.45E−02	**1.56E−03**
	Best	5.76E−04	9.74E−02	8.35E−04	**9.96E−05**
	Std	2.21E−02	3.33E+00	6.29E−02	**3.97E−03**
	p-value	3.08E−04	5.62E−06	6.48E−11	
f_7	Mean	3.57E−02	6.81E−01	4.28E−02	**1.93E−04**
	Best	8.73E−04	7.36E−02	2.04E−04	**2.09E−07**
	Std	5.42E−02	9.17E−01	5.15E−02	**5.18E−04**
	p-value	4.83E−09	8.37E−11	1.32E−05	
f_8	Mean	4.23E−02	4.43E−01	2.91E−03	**7.26E−05**
	Best	8.76E−04	6.92E−02	7.85E−05	**2.21E−07**
	Std	7.73E−02	8.37E−01	8.51E−03	**9.97E−05**
	p-value	4.82E−04	8.36E−11	9.27E−21	
f_9	Mean	8.13E−02	3.58E−01	3.48E−03	**9.73E−04**
	Best	4.47E−03	8.65E−02	9.89E−05	**7.56E−05**
	Std	2.32E−01	6.68E−01	8.31E−03	**2.21E−03**
	p-value	3.29E−14	8.86E−08	6.94E−24	

results show that the LDDWGWO algorithm outperforms the other optimization algorithms.

C. Running time comparison

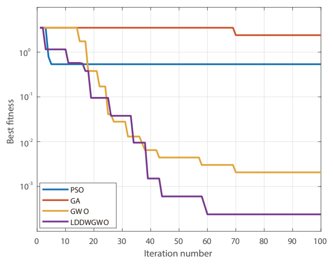
Furthermore, in evaluating the algorithms' performance, we conducted comparative analyses of the running times between LDDWGWO and other optimization algorithms. Table III displays the average duration needed to execute each iteration for every algorithm under assessment. Each algorithm was still iterated 100 times in dimensions 20, 30, 50 and 100. each experiment was repeated 10 times to obtain the average running time for each iteration as shown in Table III. The results show that the average running time of LDDWGWO is comparable to the average running time of the original GWO when the dimension is 20. Furthermore, as the dimension increases, it is clear that LDDWGWO outperforms the other optimization algorithms in terms of the average running time per iteration.

TABLE III
COMPARISON OF AVERAGE RUNNING TIME OF EACH ITERATION FOR DIFFERENT ALGORITHMS

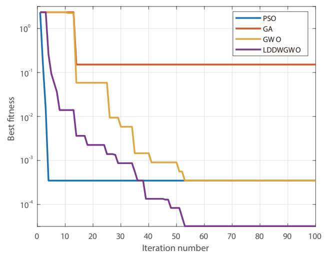
Dimension	PSO	GA	GWO	LDDWGWO
20	0.0378	0.0321	**0.0215**	0.0223
30	0.0392	0.0306	0.0284	**0.0258**
50	0.0839	0.1278	0.0735	**0.0678**
100	0.3196	0.4425	0.2981	**0.2832**

V. CONCLUSION

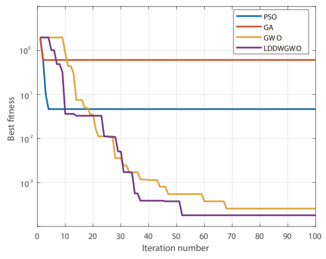
GWO has certain drawbacks in solving trajectory optimization problems. When randomly initializing individuals, the path points generated are haphazard and disorderly, resulting in a relatively low degree of individual adaptation. In addition, the position update strategy is inflexible, so it is difficult to go beyond the local optimum to find the global optimum solution in some of the optimization search processes. To reduce the above problems, this paper combines GWO with a linear differential decrement strategy and dynamic exponentially weighted averaging. Specifically, a LDDWGWO algorithm is proposed. It uses linear differentiation to update the parameters of GWO, thus improving the average fitness value of the initial population. In addition, a dynamic exponentially weighted averaging method is proposed in order to increase the flexibility of the location update, in the hope of jumping out of the local optimum. In this paper, LDDWGWO is applied to APF that are prone to local optima, and LDDWGWO is evaluated by using nine benchmark fitness functions with different shapes. In contrast to other standard optimization algorithms and the original GWO, LDDWGWO exhibits superior performance concerning both search accuracy and runtime efficiency, as revealed by comparative analyses.



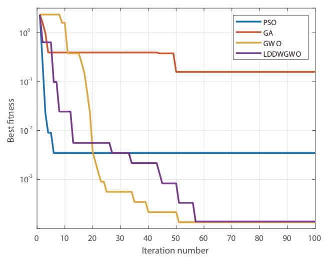
(a) Sphere (f_1)



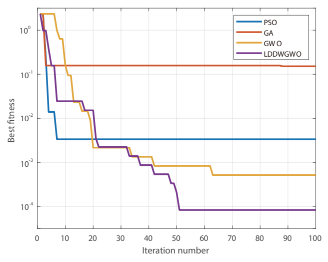
(b) Sumsquares (f_2)



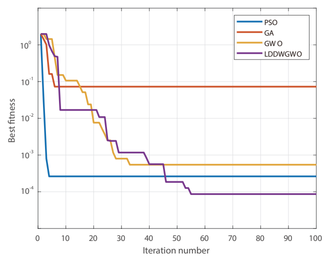
(c) Step (f_3)



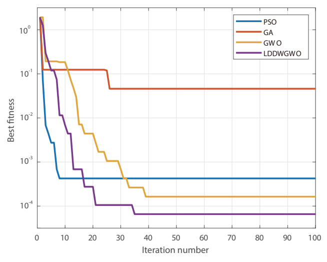
(d) Quartic (f_4)



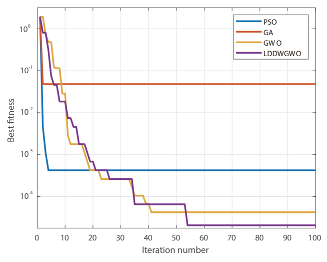
(e) Rosenbrock (f_5)



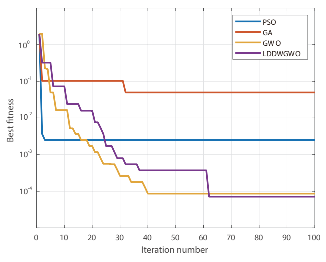
(f) Schwefel 2.22 (f_6)



(g) Rastrigin (f_7)



(h) Griewank (f_8)



(i) Ackley (f_9)

Fig. 4. Comparison of convergence curves of LDDWGWO and other four algorithms obtained in nine benchmark functions

References

[1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

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

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

[4] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of large scale power system using quasi-oppositional grey wolf optimization algorithm," *Engineering Science and Technology, an International Journal*, vol. 19, no. 4, pp. 1693–1713, 2016.

[5] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80, 2018.

[6] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE access*, vol. 7, pp. 113 810–113 825, 2019.

[7] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Advances in Engineering Software*, vol. 99, pp. 121–136, 2016.

[8] J. Li, X. Dong, S. Ruan, and L. Shi, "A parallel integrated learning technique of improved particle swarm optimization and bp neural network and its application," *Scientific Reports*, vol. 12, no. 1, p. 19325, 2022.

[9] J. S. Hunter, "The exponentially weighted moving average," *Journal of quality technology*, vol. 18, no. 4, pp. 203–210, 1986.

[10] N. A. Saleh, M. A. Mahmoud, and A.-S. G. Abdel-Salam, "The performance of the adaptive exponentially weighted moving average control chart with estimated parameters," *Quality and Reliability Engineering International*, vol. 29, no. 4, pp. 595–606, 2013.

[11] A. Yeganeh, A. Shadman, and A. Amiri, "A novel run rules based mewma scheme for monitoring general linear profiles," *Computers & Industrial Engineering*, vol. 152, p. 107031, 2021.

[12] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of computational design and engineering*, vol. 5, no. 4, pp. 458–472, 2018.

[13] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, and H. Yang, "Path planning for solar-powered uav in urban environment," *Neurocomputing*, vol. 275, pp. 2055–2065, 2018.

[14] C. YongBo, M. YueSong, Y. JianQiao, S. XiaoLong, and X. Nuo, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 266, pp. 445–457, 2017.

[15] J. Luo and B. Shi, "A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems," *Applied Intelligence*, vol. 49, pp. 1982–2000, 2019.

[16] J. Bi, W. Gu, and H. Yuan, "Hybrid whale optimization algorithm with differential evolution and chaotic map operations," in *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, vol. 1. IEEE, 2021, pp. 1–6.

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[18] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.