IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 20, 2021, accepted June 8, 2021, date of publication June 21, 2021, date of current version June 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3090776

Path Planning of UAV Based on Improved Adaptive Grey Wolf Optimization Algorithm

WEI ZHANG, SAI ZHANG, FENGYAN WU, AND YAGANG WANG

Shanghai Key Laboratory of Modern Optical System, University of Shanghai for Science and Technology, Shanghai 200093, China
Engineering Research Center of Optical Instrument and System, Ministry of Education, University of Shanghai for Science and Technology, Shanghai 200093, China
School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Yagang Wang (ygwang@usst.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 11502145, Grant 61703277, and Grant 61074087.

ABSTRACT Aiming at the three-dimensional path planning of unmanned aerial vehicle (UAV) in the complex environment of material delivery in earthquake-stricken areas, this paper proposes an improved adaptive grey wolf optimization algorithm (AGWO) based on the grey wolf optimization algorithm (GWO). There are two main contributions of the proposed method. Firstly, we propose an adaptive convergence factor adjustment strategy and an adaptive weight factor to update the individual's position. The effectiveness of the improved algorithm is verified by the convergence analysis and the test function simulation experiment. Secondly, the improved algorithm is applied to UAV path planning, the environmental map model is established by integrating digital elevation map and equivalent mountain threat model, and the performance evaluation function is established by fitting the calculated track length. The simulation results show that the improved AGWO is superior to the traditional intelligent algorithm in convergence precision, speed and stability performance, and it is effective for 3D trajectory optimization in complex environment.

INDEX TERMS Adaptive grey wolf optimization algorithm (AGWO), path planning, unmanned aerial vehicle (UAV).

I. INTRODUCTION

With the development of the unmanned aerial vehicle (UAV) technology, UAVs have been used in complex and dangerous environments such as searching and rescuing in earthquake-stricken areas. In the earthquake-stricken area, a large number of relief materials need to be transported to the designated places. However, most of the earthquake areas are mountainous areas with complex terrain. Moreover, earthquakes are often accompanied by debris flows and aftershocks in these areas. As a result, the risk of transporting relief materials on land is very high. To ensure timely, accurate and safe delivery of relief materials, UAV technology has become the best choice. As an important part of the UAV mission planning system, the development of path planning technology directly affects whether UAVs can complete the task or not. Therefore, 3D flight path planning technology becomes the key to the development of UAV technology.

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

Traditional path planning algorithms including artificial potential field method [1], A* algorithm [2], [3], Simulated Annealing algorithm (SA) [4], and so on. These traditional path planning methods are not suitable for complex environments with various constraints. Therefore, research on path planning methods based on metaheuristic algorithms has become a topic of general interest.

In recent decades, metaheuristic algorithms have been developed rapidly such as the Particle Swarm Optimization (PSO) [5] algorithm, the Genetic Algorithm (GA) [6] algorithm, the Differential Evolution (DE)[7] algorithm and the Gradient-Based Optimizer (GBO)[8], Ant Colony Optimization algorithm (ACO) [9], Cuckoo Search algorithm (CS) [10], Glowworm Swarm Optimization algorithm (GSO) [11], Whale Optimization Algorithm (WOA) [12], Ant Lion Optimizer (ALO) [13], Virus Colony Search (VCS) [14], Slime Mould Algorithm (SMA) [15], and Harris Hawks Optimization (HHO) [16], etc. These metaheuristic algorithms have more advantages in solving path planning problems for complex environments. More and more researchers have applied

these metaheuristic algorithms to solve the path planning problems for complex environments. Dewang *et al.* [17] proposed the APSO algorithm with adaptive weight factor based on the PSO algorithm for solving a two-dimensional path planning problem. The proposed method can overcome the shortcoming of falling into the local optimal solution of PSO algorithm. Ref [18] put forward the idea of combining the PSO algorithm with optimal control, which compensates for the defects of a single algorithm and has a good effect on the path planning of surgical robots. Ref [19] proposed a new ACO-DE algorithm for UAV 3D path planning. The proposed algorithm improved the updating process of ant pheromone and applied DE to optimize the pheromone trail of the improved ACO model during the process of ant pheromone updating. The Firefly Algorithm [20] was used to solve the UAV path planning problems, which has a good performance in the 3D environment as well. Ref [21] proposed an improved cuckoo search algorithm based on compact and parallel techniques, which saves the memory space of the robot and improves the accuracy and speed of the algorithm. Except for these algorithms which have mentioned above, a large number of algorithms have been applied to path planning problems in different scenes [22]–[24].

Grey Wolf Optimization algorithm (GWO) [25], a swarm intelligent optimization algorithm proposed by Mirjalili *et al.* in 2014, is a simple and flexible algorithm. It is very effective for solving highly nonlinear, multi-variable, and multimodal function optimization problems. Many researchers have studied the GWO algorithm and confirmed that the GWO algorithm is superior to the PSO and ACO algorithm in solving different types of optimization problems [26]–[28]. Moreover, GWO algorithm has also been applied to UAV path planning. Zhang *et al.* [29] firstly applied the GWO algorithm for UAV path planning. The proposed method has good performances in terms speed, accuracy, and stability and has been verified from the simulation results in two-dimensional space. Dewangan *et al.* [30] realized 3D path planning for multiple UAVs by using the GWO algorithm. However, it should be pointed out that GWO algorithm also has premature convergence and local optimal problems in large-scale optimization problems. To overcome these weaknesses of the GWO algorithm [31], Ge *et al.* [32] proposed a new method based on GWO algorithm and FOA algorithm. The path is initialized by the GWO algorithm firstly. Then the FOA algorithm is adopted for optimization. The accuracy of the GWO algorithm has been improved. Qu *et al.* [33] proposed a novel hybrid HSGWO-MSOS algorithm which is a combination of the GWO algorithm and the modified symbiotic organisms search (MSOS) algorithm. To ensure the operation of solar UAVs in an urban environment, Wu *et al.* [34] proposed an improved IGWO algorithm to overcome the local optimal defect.

To improve the fluency of UAV in complex environment such as the earthquake disaster area, this paper analyzes the convergence strategy of the GWO algorithm and improves two aspects. Firstly, an adaptive convergence factor strategy is proposed. The convergence rate is adjusted dynamically by introducing the rate of variation of centrifugal distance. Secondly, an adaptive weight factor is designed for multi-dimensional complex optimization problems and continuous optimization problems. The individual position can be dynamically updated with the overall convergence degree of the population. Finally, a UAV environment map model is established to generate an effective path for the UAV flight by integrating a digital elevation map and equivalent mountain threat model. The effectiveness of the proposed improving strategy is verified by convergence analysis and test function simulation experiments. And the UAV running trajectory is simulated to verify the reliability of the improved GWO algorithm. The source codes of this algorithm can be found in https://github.com/ZS-Lib/AGWO.

II. PROBLEM DESCRIPTION

A. ENVIRONMENTAL MODEL

The environment of UAV is complex when it performs tasks in the actual scene. The environment model is the foundation for the UAV path planning. In this paper, except for elevation information on the basis of a two-dimensional map, the threat of complex terrain is also considered. An equivalent environment model combining digital elevation information and threat information is generated to simulate the real environment.

The method of building digital elevation information is to abstract the planning space into a 3D digital map with elevation information. It assumes that the elevation value of each point in the space is a function of the coordinates of the point on the horizontal plane, and the height of the horizontal plane is set as $H_{\min}$. This means that when the UAV's altitude is lower than the horizontal plane, the UAV is forbidden to fly. Then, we assume that the horizontal coordinate of a point in space is (x, y), the elevation information of this point is

$$Z_1(x, y) = f(x, y) + H_{\min} \tag{1}$$

where $Z_1(x, y)$ is the elevation value of point (x, y), $f(x, y)$ represents the corresponding relationship between the elevation value of point (x, y) and the horizontal coordinates, $H_{\min}$ is the minimum altitude limit for UAV flight.

The specific equivalent transformation formula of the threat model can be expressed as follows:

$$Z_2(x, y) = \sum_{i=1}^{N} h_i \exp\left[-\left(\frac{x - x_{i0}}{x_{si}}\right)^2 - \left(\frac{y - y_{i0}}{y_{si}}\right)^2\right] \tag{2}$$

where $Z_2(x, y)$ is the elevation value of the current threat point, N is the total number of threat points, h_i represents the scope of action of the threat point, (x_{i0}, y_{i0}) is the coordinate of the central position of the threat point, x_{si} represents the slope of the threat point along the X-axis, and y_{si} represents the slope along the Y-axis.

According to the above model, information fusion technology is used to fuse 3D elevation digital map and threat

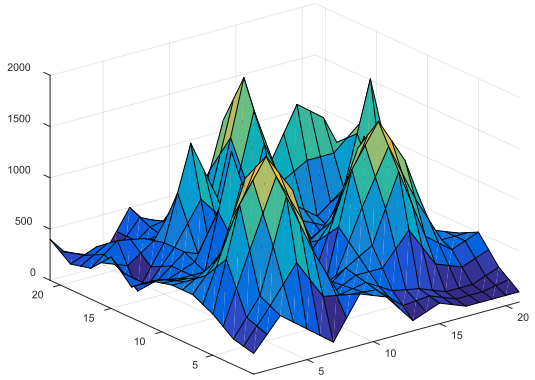


FIGURE 1. Environmental model for three-dimensional planning.

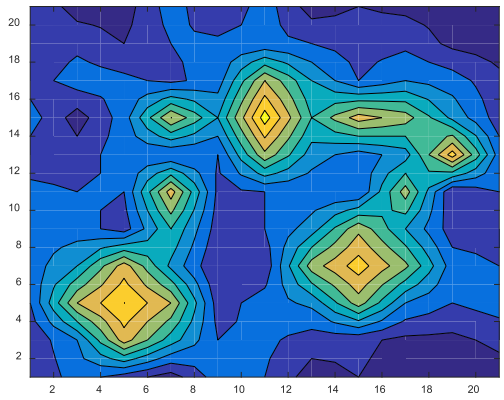


FIGURE 2. Top view of environmental model for three-dimensional planning.

model to generate equivalent environment model. The elevation numerical information of the environment model can be expressed as follows:

$$Z(x, y) = \max\left[Z_1(x, y), Z_2(x, y)\right] \tag{3}$$

where $Z_1(x, y)$ is the elevation value of the point (x, y) in the original elevation map, $Z_2(x, y)$ is the elevation value of the threat point in the threat model.

The environment model is shown in Fig. 1 and the vertical view is shown in Fig. 2.

B. TRACK CONSTRAINTS AND COST FUNCTION

The flight trajectory of the UAVs can be evaluated using a cost function. The optimal trajectory is chosen according to the cost value. In the context of the application of disaster relief materials transportation, we mainly consider the fuel consumption cost and threat cost in the planning process. The track cost assessment function is established as follows:

$$\min J = \min \int_0^L \left[\delta_f J_f(s) + \delta_t J_t(s)\right] ds \tag{4}$$

where J is the objective function of track optimization, L is the flight path length of the UAV, $J_f(s)$ represents the cost of fuel consumption, $J_t(s)$ represents the terrain threat cost, δ_f, δ_t represent the weight value of the cost.

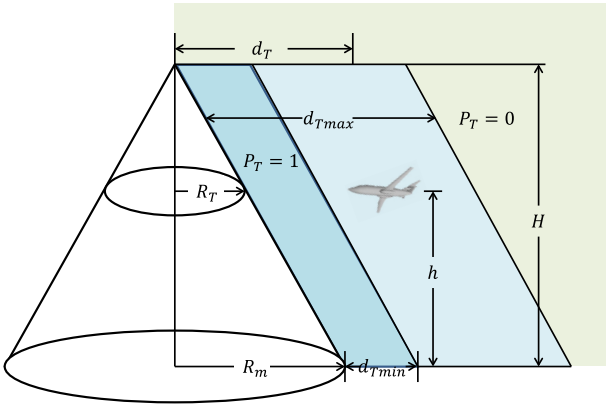


FIGURE 3. The mountain threat model is represented by a cone.

The fuel consumption during UAV flight is mainly related to the length of flight path, and the fuel consumption cost can be expressed as follows:

$$J_f = c_1 * L \tag{5}$$

where c_1 represents the proportional relationship between track length and fuel consumption.

In Fig. 1, it can be seen that the threat information is mainly composed of mountain model. Therefore, we can use cone to simulate the mountain threat of UAV in flight. As shown in Fig. 3, we suppose that the peak's maximum height above the ground is H, the maximum radius of the terrain is R_m, and the slope of the peak is θ. It is assumed that the flying height of the UAV at a certain time is h, and $h \le H$, the distance from the UAV to the central axis of the mountain is d_T, and the extending radius of the mountain at this height is $R_T(h)$. The crash probability of the UAV under the mountain terrain threat can be expressed as follows.

$$\theta = \arcsin\left(\frac{H}{\sqrt{R_m^2 + H^2}}\right), R_T(h) = \frac{H - h}{\tan\theta} \tag{6}$$

$$p_T(d_T, h) = \begin{cases} 0 \quad (h > H, or, d_T > R_T + d_{T\max}) \\ \dfrac{d_{T\max} + R_T(h) - d_T}{d_{T\max} - d_{T\min}} \\ \quad (h \le H, R_T + d_{T\min} \le d_T \le R_T + d_{T\max}) \\ 1 \quad (h \le H, d_T < R_T + d_{T\min}) \end{cases} \tag{7}$$

where $d_{T\min}$ represents the minimum distance between the UAV and the mountainous terrain, $d_{T\max}$ represents the maximum distance that the mountain terrain has an impact on the UAV.

During the flight, the terrain threat cost of a mountain k_T to the UAV can be approximately expressed as:

$$J_{T_{kT}} = \begin{pmatrix} p_T(d_{kT,1}) + p_T(d_{kT,2}) + p_T(d_{kT,3}) \\ + p_T(d_{kT,4}) + p_T(d_{kT,5}) \end{pmatrix} / 5 \tag{8}$$

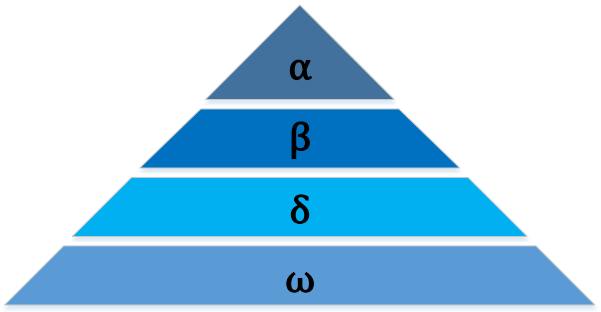


FIGURE 4. The social hierarchy of the grey wolf.

Assuming that there are n_T mountain terrain threats in the whole track, the terrain threat cost can be expressed as:

$$J_t = \sum_{k_T=1}^{n_T} J_{T_{kT}} \tag{9}$$

III. GREY WOLF OPTIMIZATION ALGORITHM

The grey wolf optimization algorithm constructs a strict hierarchical task system of grey wolf population by simulating the nature, internal mechanism and hunting behavior of grey wolf population. The grey wolf population is divided into four categories according to the class: α, β, δ and ω. The social hierarchy of the grey wolf is shown in Fig. 4.

In the grey wolf optimization algorithm, the results are corresponding to the four categories of individuals. The position of α wolf is defined as the history of the optimal solution. The position of β wolf is defined as the suboptimal solution. The position of δ wolf is the third optimal solution, and the other candidate solutions are the remaining individuals of ω wolves.

When grey wolves hunt in groups, their main behaviors contain surrounding, hunting and attacking their prey. The behavior of grey wolf populations surrounding their prey can be described by the following formula:

$$D = \left|C * X_p(t) - X_i(t)\right| \tag{10}$$

$$X_i(t+1) = X_p(t) - A * D \tag{11}$$

where D represents the distance between the individual and the target; t is the current iteration number; $X_p(t)$ is the prey's current location coordinate and can be expressed as $X_p = \left(X_p^1, X_p^2, \cdots, X_p^D\right)$;$X_i(t+1)$ is the ith wolf in iteration after the position vector and can be expressed as $X_i = \left(X_i^1, X_i^2, \cdots, X_i^D\right), i = 1, 2, \cdots, N, N$ is the population size; A and C are synergies, which can be expressed as follows:

$$A = 2a^* rand_1 - a \tag{12}$$

$$C = 2^* rand_2 \tag{13}$$

$$a = 2 - \frac{2t}{t_{\max}} \tag{14}$$

where $rand_1$ and $rand_2$ are random numbers in the range of [0,1], a is called the convergence factor and $t_{\max}$ is the maximum number of iterations.

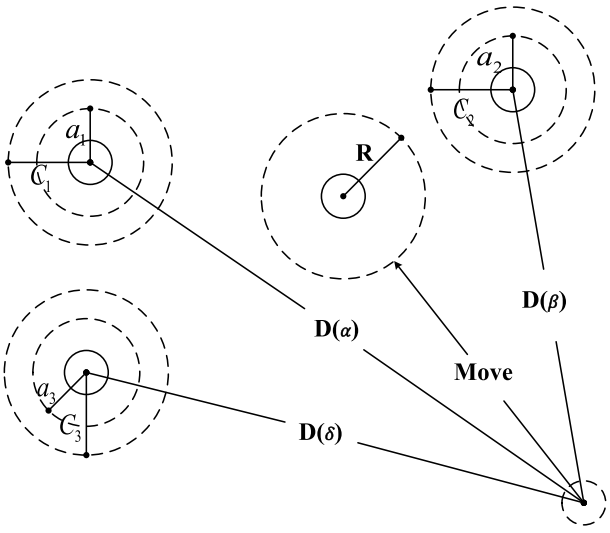


FIGURE 5. Grey wolf population location update process.

In the actual hunting process, the location of the prey is generally known to the grey wolf population. But in the abstract search space, the location of the prey is often unknown. In the process of random search for the prey, individual grey wolves in the population will adjust their positions in real time according to the position of the closest individual to the prey, and then gradually move closer to the prey position. In the grey wolf optimization algorithm, the nearest individuals to the prey are α, β, δ. The location updating of the grey wolf optimization algorithm is based on the location of the α, β, and δ. Fig. 5 shows the position updating process of the grey wolf population according to the positions of α, β, δ.

The renewal process of the grey wolf population location can be described as follows:

$$\begin{cases} D_\alpha = |C_1 * X_\alpha(t) - X_i(t)| \\ D_\beta = \left|C_2 * X_\beta(t) - X_i(t)\right| \\ D_\delta = |C_3 * X_\delta(t) - X_i(t)| \end{cases} \tag{15}$$

$$\begin{cases} X_{i,\alpha}(t+1) = X_\alpha(t) - A_1 * D_\alpha \\ X_{i,\beta}(t+1) = X_\beta(t) - A_2 * D_\beta \\ X_{i,\delta}(t+1) = X_\delta(t) - A_3 * D_\delta \end{cases} \tag{16}$$

$$X_i(t+1) = \frac{X_{i,\alpha} + X_{i,\beta} + X_{i,\delta}}{3}. \tag{17}$$

where C_1, C_2 and C_3 are random vectors, $X_i(t)$ is the ith wolf in the current position vector, A_1, A_2 and A_3 are adaptive vectors,$X_\alpha(t)$ represents the current location of the α wolf, $X_\beta(t)$ represents the current location of the β wolf, $X_\delta(t)$ is the current location of δ wolf. Equation (16) describes the distances between individual grey wolf and α β, δ wolves. Equation (17) defines the final position of the grey wolf individual.

IV. IMPROVED GREY WOLF OPTIMIZATION ALGORITHM

In the face of multi-dimensional complex optimization problems, grey wolf optimization algorithm is prone to premature

and fall into local optimization. To overcome these problems, the convergence factor and position updating formula of the grey wolf optimization algorithm are improved.

A. DYNAMIC ADJUSTMENT OF NONLINEAR CONVERGENCE FACTOR

In the grey wolf optimization algorithm, parameter A plays a role in balancing the global and local capabilities of the algorithm. It is necessary to assign an appropriate value to parameter A. It is known that when $|A| > 1$, the grey wolf population tends to expand the search range in order to find more suitable prey, which corresponds to the global search capability of the algorithm. When $|A| < 1$, the grey wolf population tends to narrow the search range, encircling the prey from all directions and launching attacks, which corresponds to the local search capability of the algorithm. Equation (14) shows that the parameter a decreases linearly with the number of iterations, while the parameter A is greatly affected by a. The grey wolf optimization algorithm is a nonlinear optimization process, and linear decreasing a cannot fully represent this process. Therefore, the parameter a needs to be redesigned.

Inspired by the adaptive weight optimization in PSO algorithm, this paper proposes an adaptive convergence factor strategy. In this strategy, we propose the concept of centrifugal distance change rate. The change rate of the current centrifugal distance of each individual can be calculated from the average centrifugal distance and the maximum centrifugal distance. The centrifugal distance is actually the spatial distance between an individual in a population and the historical optimal position. The distribution of candidate solutions is judged by the change of centrifugal distance, and the parameter a is adjusted to dynamic change and nonlinear attenuation. The introduction of centrifugal distance variation rate makes the algorithm realize both global searching and local searching appropriately according to the distribution of solutions.

When the population searches for the optimal solution in a D-dimensional space, the average and maximum centrifugal distances of each individual in the space can be expressed by equations (18) and (19), respectively.

$$dist_{ave} = \frac{\sum_{i=1}^{N} \sqrt{\sum_{d=1}^{D} \left(X_i^d - X_\alpha^d\right)^2}}{N} \tag{18}$$

$$dist_{\max} = \max_{i=1,2,\cdots,N} \left(\sqrt{\sum_{d=1}^{D} \left(X_i^d - X_\alpha^d\right)^2} \right) \tag{19}$$

where N is the size of the population, X_α^d is the best position vector in history, namely, the position vector of α wolf, X_i^d is the d-dimensional position vector of grey wolf individual i.

The change rate of centrifugal distance can be expressed as follows:

$$\lambda = \frac{dist_{\max} - dist_{ave}}{dist_{\max}} \tag{20}$$

At the beginning of the iteration, due to large λ value, the divergence of population distribution should make a decrease rapidly, and the local search will be strengthened appropriately to improve the convergence speed. In later period, λ is a smaller value, population distribution is relatively dense, this should make a at a slower rate decreases, and disperse the wolves appropriately, avoid algorithm falls into local optimum. Accordingly, adding λ to the update formula of convergence factor a can dynamically adjust its convergence speed and effectively coordinate global searching and local searching.

The improved convergence factor updating formula is given as follows:

$$a = 2 - \lg\left(1 + 6 * \lambda * \frac{t}{t_{\max}}\right) \tag{21}$$

where t is the current iteration number, $t_{\max}$ is the maximum number of iterations, λ is the centrifugal distance change rate.

B. ADAPTIVE WEIGHTING FACTOR

According to equation (17), the final position of the wolf ω is determined by its average step length and direction towards the α, β and δ wolves. However, in the actual environment, due to the strict hierarchical mechanism of grey wolf population, the effects of α, β and δ wolves on grey wolf population are different. Therefore, the adaptive weight factor is introduced into the position update formula of the algorithm. Based on the centrifugal distance change rate in the previous section, the dynamic distribution of candidate solutions can be effectively reflected and the adaptive weight factor is designed by using the centrifugal distance change rate. In this way, the position of individual grey wolf can be dynamically updated with the overall convergence degree of the population in the iteration process, so as to effectively improve the optimization performance of the algorithm.

The change rate of the centrifugal distance λ in the upper segment mainly reflects the closeness between each individual and the α wolf. However, the complexity of the actual environment leads to the fact that the α wolf is not necessarily the global optimal solution. Therefore, we also need to know how close each individual is to β wolf and δ wolf. The formula for calculating the centrifugal distance of each individual to α, β and δ are given as follows.

$$dist_{xave} = \frac{\sum_{i=1}^{N} \sqrt{\sum_{d=1}^{D} \left(X_i^d - X_x^d\right)^2}}{N}, \quad (x = \alpha, \beta, \delta) \tag{22}$$

$$dist_{x\max} = \max_{i=1,2,\cdots,N} \left(\sqrt{\sum_{d=1}^{D} \left(X_i^d - X_x^d\right)^2} \right), \quad (x = \alpha, \beta, \delta) \tag{23}$$

where X_i^d is the D-dimensional position vector of individual i, X_x^d is the D-dimensional position vector of individual x, which is the historical optimal position vector, where x may be α, β or δ. Then the change rate of the current centrifugal

distance of each individual with respect to α, β and δ can be expressed as follows:

$$\lambda_\alpha = \frac{dist_{\alpha\,\max} - dist_{\alpha ave}}{dist_{\alpha\,\max}} \tag{24}$$

$$\lambda_\beta = \frac{dist_{\beta\,\max} - dist_{\beta ave}}{dist_{\beta\,\max}} \tag{25}$$

$$\lambda_\delta = \frac{dist_{\delta\,\max} - dist_{\delta ave}}{dist_{\delta\,\max}} \tag{26}$$

where λ_α is described as the variation of the centrifugal distance of the individual grey wolf with respect to the α wolf; λ_β is described as the variation of the centrifugal distance of β wolf; λ_δ is described as the variation of the centrifugal distance of the δ wolf. Then the following adaptive weight factors can be designed according to the current centrifugal distance change rate of each individual with respect to α, β, δ.

$$\omega_1 = \frac{\lambda_\alpha}{\lambda_\alpha + \lambda_\beta + \lambda_\delta} \tag{27}$$

$$\omega_2 = \frac{\lambda_\beta}{\lambda_\alpha + \lambda_\beta + \lambda_\delta} \tag{28}$$

$$\omega_3 = \frac{\lambda_\delta}{\lambda_\alpha + \lambda_\beta + \lambda_\delta} \tag{29}$$

where ω_1 represents the influence degree of wolf α on other wolves; ω_2 represents the influence degree of wolf β on other wolves; and ω_3 represents the influence degree of wolf δ on other wolves. Combined with the designed adaptive weight factor, the new position updating formula is obtained as follows.

$$X_i\,(t+1) = \frac{\omega_1 X_{i,\alpha} + \omega_2 X_{i,\beta} + \omega_3 X_{i,\delta}}{3} \tag{30}$$

where $X_{i,\alpha}$ denotes the distance between individual i and wolf α, $X_{i,\beta}$ denotes the distance between individual i and wolf β, $X_{i,\delta}$ denotes the distance between i and δ wolf.

C. COMPUTAYIONAL COMPLEXITY AND PSEUDO CODE

This part analyzes the computational complexity of the proposed AGWO algorithm and gives pseudocode.

The big O method is used to express computational complexity. As can be seen from the pseudo code below, the size of the initial population X_i is n (denoted as N), and the maximum number of iterations is $t_{\max}$ (denoted as T). Therefore, the time complexity of initializing the population is O(N), and the time complexity of individual iteration is O(T). It can be verified that the complexity of the entire algorithm can be denoted as O($N + T$). Big O notation only cares about the scale of the input data, so the time complexity of most swarm intelligence algorithms can be recorded as O($N + T$), such as GWO, PSO, WOA algorithms. This means that the improved AGWO algorithm has the same computational complexity as the GWO algorithm, and the big O notation method cannot well reflect their differences.

Thus we give a new calculation method of time complexity below. Firstly, it is assumed that the population size is N, the maximum number of iterations is m. Then, we assume that the time for each individual to calculate once is t, then the total running time T of the algorithm can be expressed as follows.

$$T = N * m * t \tag{31}$$

The value of t depends on the number of mathematical operations included in the algorithm. We classify addition and subtraction as one type of operation, the running time is denoted as t_a, multiplication and division are a type of operation, and the running time is denoted as t_m, the time of logarithmic operation is recorded as t_l, t_e represents the time of exponential operation, and t_t represents the time of trigonometric function operation. Then (31) can be expressed as

$$T = \sum_{i=1}^{m} N * (x * t_a + y * t_m + z * t_l + u * t_e + v * t_t) \tag{32}$$

where x, y, z, u, v represent the number of the corresponding mathematical operations.

For GWO algorithm, it can be calculated that $x = 12$, $y = 16$, $z = 0$, $u = 0$, $v = 0$. For the proposed AGWO, the values are $x = 22, y = 37, z = 1, u = 0, v = 0$. Therefore, in terms of computational complexity, AGWO has increased its computational complexity due to the introduction of two new strategies.

The pseudo code is as follows:

V. NUMERICAL EXPERIMENT

Three kinds of benchmark functions including unimodal, multimodal and fixed dimension multimodal are used to test the effectiveness of the proposed AGWO algorithm. These three kinds of functions have different characteristics, which can be used to test the performance of the algorithm. Table 1 gives the specific function description of the test functions.

To verify the effectiveness of the proposed method, simulation results of the proposed method are compared with GWO algorithm [25], GWO_1 algorithm [35], NGWO algorithm [36], LFGWO algorithm [37] and IGWO algorithm [38]. In simulation, AGWO_1, AGWO_2 and AGWO_3 are improved GWO using the strategy proposed in this paper. AGWO_1 improves the convergence factor, AGWO_2 improves the weight factor, and AGWO_3 improves both the convergence factor and the weight factor. All the comparative experiments set the experimental parameters uniformly. The population size is set as 30, and the number of iterations is set as 500. In order to avoid the randomness of the experiment, we run all the experimental algorithms independently for 20 times, and take the average results. The convergence accuracy of every algorithm is expressed by the average value of the results of 20 runs, and the stability of the algorithm is expressed by the standard deviation of the results of 20 runs. The experimental results are shown in Table 2, Table 3 and Table 4.

TABLE 1. Benchmark function description.

Type	Functions	Dim	Range	$f_{\min}$
Unimodal function	$f_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
	$f_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	30	[-100,100]	0
	$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	[-100,100]	0
	$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30,30]	0
	$f_6(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
Multimodal function	$f_7(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.982×5
	$f_8(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12]	0
	$f_9(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[-32,32]	0
	$f_{10}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0
Fixed dimension multimodal function	$f_{11}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[-65,65]	1
	$f_{12}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	4	[-5,5]	0.00030
	$f_{13}(x)=4x_1^2-2.1x^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
	$f_{14}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,5]	0.398
	$f_{15}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
	$f_{16}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[0,1]	-3.86

TABLE 2. Results of unimodal benchmark functions.

function	performance	GWO	GWO_1	NGWO	LFGWO	IGWO	AGWO_1	AGWO_2	AGWO_3
f_1	mean	2.2152e-27	2.4560e-33	3.8253e-36	1.9346e-49	3.6488e-39	1.9498e-54	**0**	**0**
	std	5.4499e-27	3.4835e-33	6.6844e-36	2.5013e-49	5.8011e-39	7.6319e-54	**0**	**0**
f_2	mean	9.4099e-17	5.1393e-20	1.0711e-21	2.0309e-30	1.9576e-23	1.2220e-32	**1.6446e-162**	3.0365e-162
	std	5.8378e-17	3.8554e-20	5.5545e-22	1.6380e-30	2.0705e-23	1.6933e-32	**0**	**0**
f_3	mean	6.4263e-06	1.1458e-05	2.4585e-06	4.3187e-10	2.6826e-07	4.1311e-10	**0**	**0**
	std	1.6713e-05	4.7643e-05	1.0045e-05	7.8485e-10	9.3692e-07	1.7986e-09	**0**	**0**
f_4	mean	9.5841e-07	2.2674e-08	3.3149e-09	1.1659e-12	8.1292e-10	1.4275e-14	**2.3814e-163**	4.2324e-163
	std	1.1974e-06	2.1576e-08	2.5114e-09	1.1035e-12	1.2056e-09	3.9709e-14	**0**	**0**
f_5	mean	27.1676	27.1431	28.1384	27.0903	27.1111	**26.9433**	28.9438	28.9398
	std	0.7056	0.7702	0.7575	0.6748	0.7253	0.7133	**0.0226**	0.0327
f_6	mean	0.0021	0.0017	0.0056	0.0011	0.0015	0.0012	**5.2201e-05**	9.8747e-05
	std	7.8319e-04	8.3551e-04	0.0030	4.9607e-04	8.7762e-04	6.6929e-04	**5.0553e-05**	1.2017e-04

A. CONVERGENCE ACCURACY ANALYSIS OF THE ALGORITHM

As shown in Table 2, Table 3 and Table 4, it can be seen clearly that the proposed three algorithms have better performances than other methods. Besides, the AGWO_2 algorithm performs best for unimodal benchmark functions (See Table 2). This shows that the adaptive weight factor strategy has a positive impact on the convergence accuracy and local search ability of the algorithm. From Table 3, it can be seen that AGWO_3 algorithm achieves the best performance. This shows that the AGWO_3 algorithm using the two improved strategies is optimal in terms of global exploration capability. Compare to the results in Table 2 and Table 3, it can be concluded that AGWO_3 algorithm takes both local search ability and global search capability into account and achieves a certain balance between them. From Table 4, it can be verified that AGWO_1 algorithm reach the best performance. This implies that the adaptive convergence

Algorithm 1 AGWO Algorithm

Begin

Step1: Initialize the grey wolf population X_i $(i = 1, 2, \ldots, n)$, the maximum number of iterations $t_{\max}$, the parameters a, A, and C, the position of the X_α, X_β, and X_δ wolf. Initialize the distance matrix of each individual grey wolf to the X_α, X_β, and X_δ wolf

Step2: **For** all X_i **do**
Evaluate the fitness value of each individual grey wolf by F(X_i)
End for
Get the first three best wolves as X_α, X_β, and X_δ

Step3: **While**(t $< t_{\max}$)
Calculate the change rate of centrifugal distance λ_α, λ_β, and λ_δ by Eq.(22) $\sim$ Eq.(26)
Update a, A, and C
Calculate the distance between the individual wolf and the X_α, X_β, and X_δ wolf by Eq.(16)
Calculate adaptive weighting factor ω_1 ω_2 ω_3 by Eq.(27) $\sim$ Eq.(29)
Update X_i by Eq. (30)
For all X_i **do**
Evaluate the fitness value of each individual grey wolf by F(X_i)
End for
Get the first three best wolves as X_α, X_β, and X_δ
Update X_α, X_β, and X_δ
t = t + 1
End while
Return X_α

End

factor strategy is effective for fixed-dimension multimodal benchmark functions.

Note: For the results in Table 4, it can be seen that the performance of AGWO_1 is better than AGWO_3. From formula (30), it can be seen that the updating position should be calculated according to three adaptive weighting factors. This means that the new location is not actually dominated by the location of the α wolf. When the location of the α wolf is locally optimal, the next location will be determined by the β wolf and δ wolf, and the new position can jump out of the local optimum. But when the α wolf's position is already at the global optimum, the α wolf's position cannot dominate the next position updating, which will cause the algorithm to miss the global optimum. This is the reason that the proposed AGWO_3 algorithm cannot achieve the best performance for unimodal benchmark functions and fixed-dimension multimodal benchmark functions.

To sum up, it is recommended that readers use AGWO_1 or AGWO_3 algorithm when solving unimodal optimization problems. When solving multi-modal optimization problems, AGWO_3 algorithm is the best choice. For solving fixed dimension optimization problems, AGWO_1 algorithm is the best choice.

B. CONVERGENCE CURVE ANALYSIS OF THE ALGORITHM

In order to further test the performance of AGWO algorithm, some representative test functions are selected to analyze the convergence rate of fitness value of all algorithms. The convergence curve is shown in Fig. 6.

From Fig. 6 (a) and (b), it can be seen that the convergence curves of AGWO_2 algorithm and AGWO_3 algorithm decrease faster than other algorithms and convergence speed of the AGWO_3 algorithm is the best. Comparing the data in Table 2, it can be seen that although the convergence accuracy of AGWO_3 is the same as that of AGWO_2, AGWO_3 is superior in speed.

From Fig. 6 (c), it can be seen that the convergence speed of AGWO_3 algorithm and AGWO_2 algorithm is consistent in multimodal benchmark functions. Moreover, it can be verified from Fig. 6(d) that AGWO_3 has the highest convergence accuracy and can effectively avoid local optimization. This also shows that the AGWO_3 algorithm with two improved strategies has excellent performance in optimizing unimodal benchmark functions and multimodal benchmark functions.

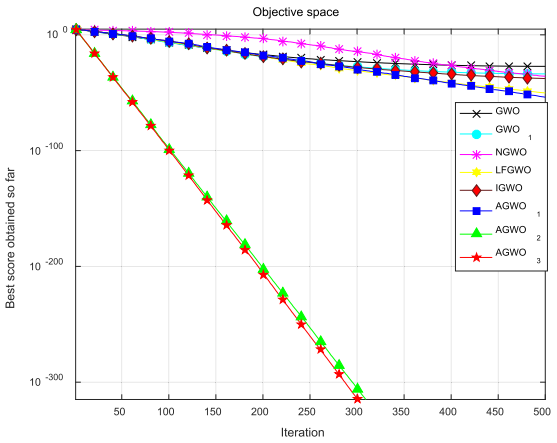
From Fig. 6 (e) and (f), it can be seen that AGWO_1 algorithm can converge to the theoretical optimal value in fixed-dimension multimodal benchmark functions F12 and F14, which is the same as the results of the above analysis of convergence accuracy. Moreover, AGWO_1 algorithm is slightly faster than other algorithms in convergence speed, which shows that the strategy of improving the convergence factor is effective.

The performance of AGWO_3 on specific problems is analyzed in the following simulation experiment of path planning problem.

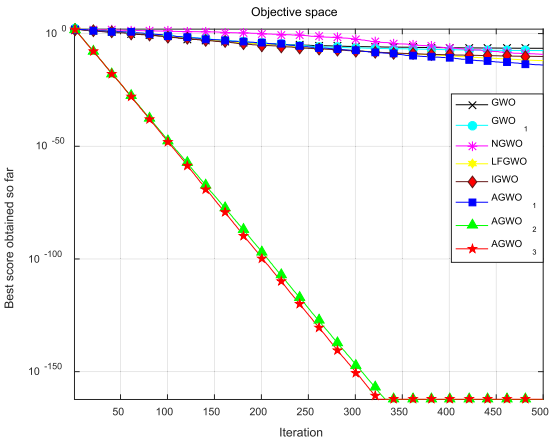
VI. SIMULATION OF ROUTE PLANNING BASED ON ADAPTIVE GREY WOLF OPTIMIZATION ALGORITHM

In order to further test the performance of AGWO_3 algorithm on specific problems, we use AGWO_3 algorithm to simulate UAV route planning on the equivalent 3D digital map. The proposed method is compared with GWO algorithm [25], PSO [5], Moth-flame optimization (MFO) [39] and Equilibrium optimizer (EO) algorithm [40]. The following AGWO_3 algorithm is referred to as AGWO algorithm.

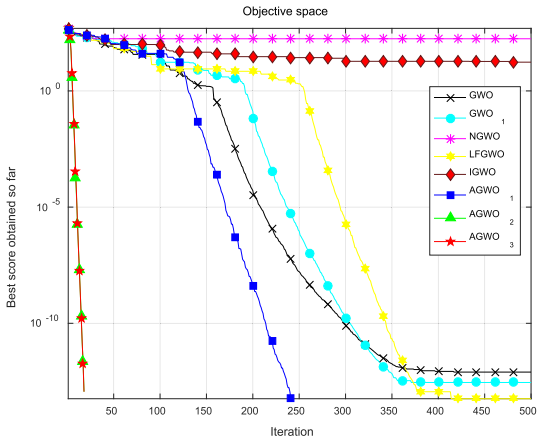
It is known that the minimum flight altitude has been incorporated into the elevation value for processing when the original 3D map is established, so it is not necessary to set the minimum flight altitude in the experiment. The safety of UAVs in the process of delivering relief materials is the first priority, the length of flight route should be considered on the premise of ensuring flight safety. When the fitness function is calculated, the weight of threat cost is set to 10, and the weight of fuel consumption cost is set to 1. The threat cost is mainly calculated by the mountain threat cost. The



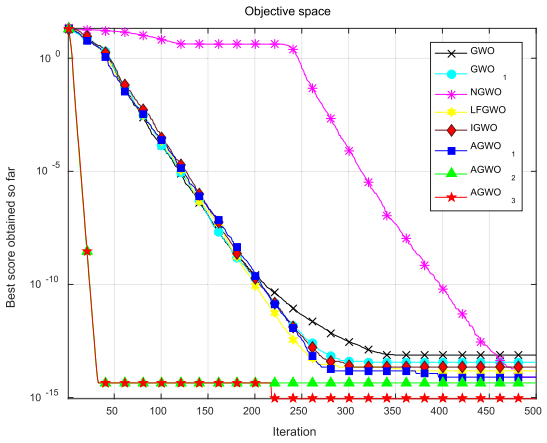
(a) Convergence curve of F1



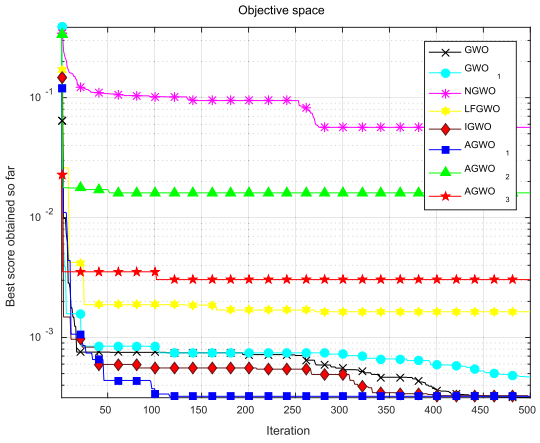
(b) Convergence curve of F4



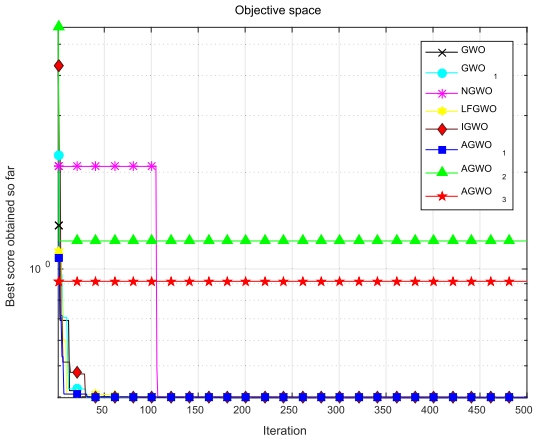
(c) Convergence curve of F8



(d) Convergence curve of F9



(e) Convergence curve of F12



(f) Convergence curve of F14

FIGURE 6. **Comparison of fitness values and convergence curves of some test functions.**

minimum distance $d_{T\min}$ between UAV and mountain terrain is set as 10 meters, and the maximum distance $d_{T\max}$ that mountain terrain will affect UAV is set as 300 meters. The fuel consumption cost is directly proportional to the planned track length, and the proportional relationship c_1 is set to 1.

In addition to the above parameters, we set the population size of each algorithm as 30, and every algorithm iterates 500 times. In order to eliminate the influence of different initial paths on the algorithm results, we set the initial path of each algorithm as the same, and set the same UAV starting

TABLE 3. Results of multimodal benchmark functions.

function	performance	GWO	GWO_1	NGWO	LFGWO	IGWO	AGWO_1	AGWO_2	AGWO_3
f_7	mean	-5735.9030	-6059.0571	-4157.0585	-3566.1671	-4440.6894	-3531.5070	-3911.7127	**-2267.3783**
	std	890.9620	884.3290	561.8324	**311.1941**	599.5731	319.9770	427.7702	464.0397
f_8	mean	2.7839	2.3646	162.3203	0.9567	1.4251	0.9045	**0**	**0**
	std	5.6442	4.3236	40.3449	3.1688	2.3773	4.0449	**0**	**0**
f_9	mean	1.0036e-13	3.6060e-14	0.8114	1.3856e-14	2.4336e-14	1.0658e-14	2.4869e-15	**2.3093e-15**
	std	1.4117e-14	4.7358e-15	1.6469	2.8874e-15	3.5340e-15	2.7938e-15	1.8134e-15	**1.7857e-15**
f_{10}	mean	0.0040	0.0054	0.0022	0.0040	**0**	9.2010e-04	**0**	**0**
	std	0.0083	0.0089	0.0069	0.0104	**0**	0.0041	**0**	**0**

TABLE 4. Results of fixed-dimension multimodal benchmark functions.

function	performance	GWO	GWO_1	NGWO	LFGWO	IGWO	AGWO_1	AGWO_2	AGWO_3
f_{11}	mean	5.3545	4.5726	7.04419	**3.6417**	5.0072	3.8439	11.3396	12.0530
	std	4.3607	4.2267	5.2734	4.0142	4.6261	3.6626	2.9344	**2.1880**
f_{12}	mean	0.0054	0.0044	0.0192	0.0055	0.0035	**0.0025**	0.0058	0.0087
	std	0.0089	0.0082	0.0379	0.0088	0.0061	0.0073	**0.0057**	0.0074
f_{13}	mean	**-1.0316**	**-1.0316**	-1.0194	**-1.0316**	**-1.0316**	**-1.0316**	-1.0088	-0.9314
	std	**2.5802e-08**	7.6827e-08	0.0155	5.6450e-06	3.3135e-06	2.7609e-06	0.0112	0.1136
f_{14}	mean	0.3979	0.3979	0.3987	0.3982	0.3981	**0.3975**	1.0080	1.3316
	std	1.2408e-04	**1.0709e-04**	0.0029	3.1053e-04	2.8102e-04	7.0012e-04	0.5178	1.1902
f_{15}	mean	3.0001	**3.0000**	3.0001	**3.0000**	**3.0000**	**3.0000**	20.9447	30.3190
	std	18.1121	3.3196e-05	8.9533e-05	2.7660e-05	3.7496e-05	**1.2779e-05**	27.1061	51.7645
f_{16}	mean	-3.8622	-3.8618	-3.8619	-3.8613	-3.8613	**-3.8621**	-3.2908	-3.2424
	std	0.0015	0.0017	**6.7326e-04**	0.0021	0.0025	0.0029	0.3254	0.3974

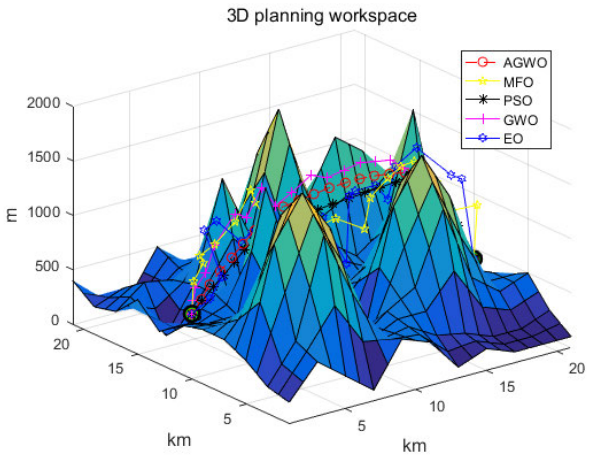


FIGURE 7. Optimal flight path of adaptive grey wolf optimization algorithm in 3D equivalent digital map.

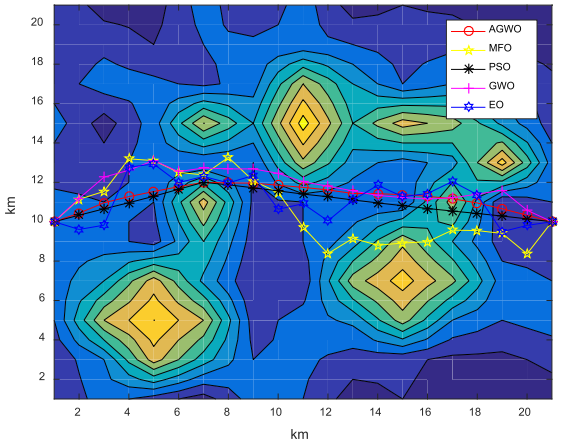


FIGURE 8. Transformation from 3D path planning map to 2D aerial view.

point coordinates S(0, 10, 3) and target point G(21, 10, 2), with the unit of kilometer. Finally, the optimal flight paths of UAV planned on 3D equivalent digital map by using AGWO, GWO, PSO, MFO, EO algorithms are shown in Fig. 7.

In order to observe the path planned by each algorithm more clearly, we convert the 3D path planning map into a 2D aerial view, as shown in Fig. 8.

From Fig. 8, it can be seen that the path planned by AGWO algorithm can effectively avoid threats when flying in the mountain area. In addition, the flight curve of the UAV is relatively smooth, and it flies approximately in a straight line in the flat and open area. Although the result of PSO algorithm is close to that of AGWO algorithm, it can be seen from Fig. 7 that the path planned by PSO algorithm is higher than that planned by AGWO algorithm in height. From Fig. 7 and Fig. 8, it can be verified that the proposed AGWO algorithm is better than PSO, GWO, MFO, and EO algorithm.

In order to see the effect of each algorithm more clearly, convergence curves of fitness value for different algorithms in the process of path planning are given in Fig. 9. Table 5 concludes the fitness value and the running time of different algorithms. From Fig. 9, it can be seen that the fitness value of AGWO algorithm converges faster than other methods at the beginning of iteration. When the iteration is about 150 times, the convergence speed is significantly reduced, and the fitness value has reached a relatively small level. When the AGWO algorithm is iterated to 300 times, it basically converges to the optimal value of path cost. As a comparison, MFO, GWO, and EO algorithms are difficult to find the optimal flight

TABLE 5. Comparison details between AGWO, MFO, PSO, GWO and EO.

Experiment Number	Runtime(s)					Fitness				
	AGWO	MFO	PSO	GWO	EO	AGWO	MFO	PSO	GWO	EO
1	**0.3673**	0.4910	1.3675	0.3721	0.5367	**21.5892**	31.0497	21.6947	27.2407	34.1866
2	**0.3880**	0.4936	1.3845	0.3876	0.4784	**22.0686**	33.406	22.3419	22.68	35.2094
3	**0.3638**	0.4584	1.4674	0.3767	0.4588	**22.9351**	33.432	43.1512	30.6362	35.9582
4	**0.3691**	0.4817	1.3265	0.3807	0.5003	**22.4119**	31.4895	22.5679	26.1331	31.8818
5	0.4277	0.6749	1.4624	**0.3679**	0.4703	**28.8944**	33.3305	39.9086	33.3486	35.3306

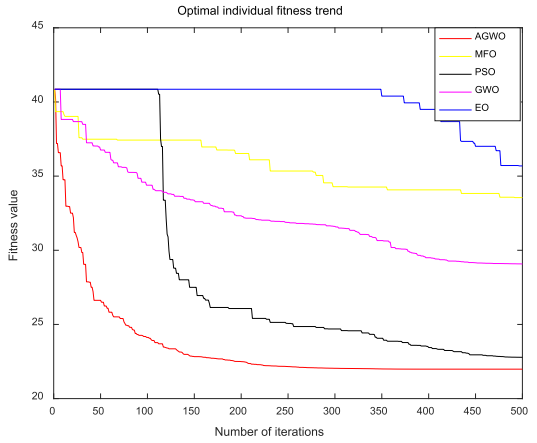


FIGURE 9. The convergence curve of fitness value with the increase of iteration times in path planning.

path from the perspective of convergence accuracy. Although the results of PSO algorithm are close to that of AGWO algorithm, its running time is too long.

Table 5 shows that the running time of AGWO is less than that of GWO. This is because that the path planning problem is calculated based on the global map. Since the search step length and direction of the intelligent algorithm are random, it is necessary to determine whether the path generated by the current iteration number is within the map range during the calculation process. If it is not in the map, we need to search for a feasible path again under the current iteration number, otherwise the program will eventually fail. Therefore, the complexity calculation formula (32) can be rewritten as the formula (33).

$$T = \sum_{i=1}^{m} N * S * (x * t_a + y * t_m + z * t_l + u * t_e + v * t_t) \quad (33)$$

where S represents the number of re-searches.

VII. CONCLUSION AND FUTURE WORK

The present work reports a novel improved AGWO algorithm, which is used to solve the path planning problem of the UAV in the earthquake disaster area. The main contributions are as follows. Based on the GWO algorithm, a dynamic adjustment strategy for nonlinear convergence factors is proposed. The distance change rate according to the distance between each individual and the current optimal individual is calculated. The convergence factor is dynamically adjusted according to the distance change rate. This strategy can overcome the shortcoming of the GWO algorithm that is easy to fall into the local optimum in the later stage. In addition, the location updating strategy is improved by introducing an adaptive weight factor, which improves the global optimization capability of the algorithm. Finally, the improved AGWO algorithm is applied to the path planning problem of UAV for complex environment to illustrate the effectiveness of the proposed method.

Though the proposed method has some merits and can be used for the path planning problem of UAV for complex environment, there are still some problems needed to research. Firstly, the proposed AGWO_3 algorithm can be further analyzed for achieving better performance. Secondly, the testing of the proposed algorithms is limited to simulation, and there is a lack of testing for path planning problems in the actual environment. The realization of the proposed method for real path planning on the quadrotor is our future work.

REFERENCES

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Mar. 1985, pp. 396–404.
[2] S. Bayili and F. Polat, "Limited-damage A*: A path search algorithm that considers damage as a feasibility criterion," *Knowl-Based Syst.*, vol. 24, no. 4, pp. 501–512, 2011.
[3] T. Zheng, Y. Xu, and D. Zheng, "AGV path planning based on improved A-star algorithm," in *Proc. IEEE 3rd Adv. Inf. Manage., Communicates, Electron. Automat. Control Conf. (IMCEC)*, Oct. 2019, pp. 1534–1538.
[4] S. Hayat and Z. Kausar, "Mobile robot path planning for circular shaped obstacles using simulated annealing," in *Proc. Int. Conf. Control, Automat. Robot.*, May 2015, pp. 69–73.
[5] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromach. Hum. Sci.*, 1995, pp. 39–43.
[6] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artif. Intell.*, vol. 40, nos. 1–3, pp. 235–282, Sep. 1989.
[7] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[8] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new Metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020.
[9] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[10] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
[11] K. N. Krishnanand and D. Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Jun. 2005, pp. 84–91.
[12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[13] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
[14] M. D. Li, H. Zhao, X. W. Weng, and T. Han, "A novel nature-inspired algorithm for optimization: Virus colony search," *Adv. Eng. Softw.*, vol. 92, pp. 65–88, Feb. 2016.

[15] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
[16] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[17] H. S. Dewang, P. K. Mohanty, and S. Kundu, "A robust path planning for mobile robot using smart particle swarm optimization," *Procedia Comput. Sci.*, vol. 133, pp. 290–297, Jan. 2018.
[18] M. T. Ramezanlou, V. Azimirad, and M. Zakeri, "Hybrid path planning of robots through optimal control and PSO algorithm," in *Proc. 7th Int. Conf. Robot. Mechatronics (ICRoM)*, Nov. 2019, pp. 259–264.
[19] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm," *Simul. Model. Pract. Theory*, vol. 18, no. 8, pp. 1104–1115, Sep. 2010.
[20] U. Goel, S. Varshney, A. Jain, S. Maheshwari, and A. Shukla, "Three dimensional path planning for UAVs in dynamic environment using glow-worm swarm optimization," *Procedia Comput. Sci.*, vol. 133, pp. 230–239, Jan. 2018.
[21] P.-C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106443.
[22] G.-G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerosp. Sci. Technol.*, vol. 49, pp. 231–238, Feb. 2016.
[23] H. Shorakaei, M. Vahdani, B. Imani, and A. Gholami, "Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm," *Robotica*, vol. 34, no. 4, pp. 823–836, Apr. 2016.
[24] E. Tuba, I. Strumberger, D. Zivkovic, N. Bacanin, and M. Tuba, "Mobile robot path planning by improved brain storm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.
[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[26] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the grey wolf optimizer for solving optimal reactive power dispatch problem," *Appl. Soft Comput.*, vol. 32, pp. 286–292, Jul. 2015.
[27] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai, "Grey wolf optimizer for parameter estimation in surface waves," *Soil Dyn. Earthq. Eng.*, vol. 75, pp. 147–157, Aug. 2015.
[28] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *J. Comput. Sci.*, vol. 8, pp. 109–120, May 2015.
[29] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.
[30] R. K. Dewangan, A. Shukla, and W. W. Godfrey, "Three dimensional path planning using grey wolf optimizer for UAVs," *Appl. Intell.*, vol. 49, no. 6, pp. 2201–2217, Jun. 2019.
[31] W. Gai, C. Qu, J. Liu, and J. Zhang, "An improved grey wolf algorithm for global optimization," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 2494–2498.
[32] F. Ge, K. Li, W. Xu, and Y. Wang, "Path planning of UAV for oilfield inspection based on improved grey wolf optimization algorithm," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 3666–3671.
[33] C. Qu, W. Gai, J. Zhang, and M. Zhong, "A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105530.
[34] 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, Jan. 2018.
[35] Z. Z. Guo, R. Liu, and C. Q. Gong, "Study on improvement of grey wolf algorithm," (in Chinese), *Appl. Res. Comput.*, vol. 34, no. 12, pp. 3603–3606, 2017.
[36] M. Wang and M. Z. Tang, "A new grey wolf optimization algorithm with nonlinear convergence factor," (in Chinese), *Appl. Res. Comput.*, vol. 33, no. 12, pp. 3648–3653, 2016.
[37] T. B. Wu, W. H. Gui, and C. H. Yang, "Improved grey wolf optimization algorithm with logarithmic function to describe convergence factor and its application," (in Chinese), *J. Central South Univ.*, vol. 49, no. 4, pp. 93–100, 2018.
[38] W. Long and T. B. Wu, "Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation," (in Chinese), *Control Decis.*, vol. 32, no. 10, pp. 1749–1758, 2017.
[39] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[40] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.

WEI ZHANG received the B.S. degree from the Department of Mathematics, Shaanxi Normal University, Xi'an, China, in 2003, the M.S. degree in operation research and cybernetics from Northeastern University, Shenyang, China, in 2008, and the Ph.D. degree in control theory and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. He is currently an Associate Professor with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai. His research interests include robust control, optimal control, decoupling control, and their applications in air vehicle and industry.

SAI ZHANG received the B.S. degree in electronic information engineering from Harbin University of Science and Technology, China, in 2018. He is currently pursuing the M.S. degree in control science and engineering with the University of Shanghai for Science and Technology, China. He focuses on the research of path planning for air vehicle using intelligent algorithm.

FENGYAN WU received the B.S. degree in light source and lighting from the Taiyuan University of Technology, in 2016, and the M.S. degree in control science and engineering from the University of Shanghai for Science and Technology, in 2020. She focuses on the research of intelligent control and its application in air vehicle.

YAGANG WANG received the B.E. degree in electrical engineering from the China University of Mining and Technology, China, in 1988, the M.E. degree in control theory and application from the Taiyuan University of Technology, China, in 1991, and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, China, in 2000. From 2000 to 2002, he worked as a Research Fellow with Nanyang Technological University, Singapore. From 2002 to 2007, he worked as a Senior Engineer at Honeywell (China) Company Ltd. He is currently a Professor with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include process automation, system identification, and wireless sensor networks.