

Article

Improved Grey Wolf Algorithm: A Method for UAV Path Planning

Xingyu Zhou , Guoqing Shi * and Jiandong Zhang 

School of Electronic Information, Northwestern Polytechnic University, Xi'an 710129, China;
xingyuzhou@mail.nwpu.edu.cn (X.Z.); jdzhang@nwpu.edu.cn (J.Z.)

* Correspondence: shiguqing@nwpu.edu.cn

Abstract: The Grey Wolf Optimizer (GWO) algorithm is recognized for its simplicity and ease of implementation, and has become a preferred method for solving global optimization problems due to its adaptability and search capabilities. Despite these advantages, existing Unmanned Aerial Vehicle (UAV) path planning algorithms are often hindered by slow convergence rates, susceptibility to local optima, and limited robustness. To surpass these limitations, we enhance the application of GWO in UAV path planning by improving its trajectory evaluation function, convergence factor, and position update method. We propose a collaborative UAV path planning model that includes constraint analysis and an evaluation function. Subsequently, an Enhanced Grey Wolf Optimizer model (NI-GWO) is introduced, which optimizes the convergence coefficient using a nonlinear function and integrates the Dynamic Window Approach (DWA) algorithm into the model based on the fitness of individual wolves, enabling it to perform dynamic obstacle avoidance tasks. In the final stage, a UAV path planning simulation platform is employed to evaluate and compare the effectiveness of the original and improved algorithms. Simulation results demonstrate that the proposed NI-GWO algorithm can effectively solve the path planning problem for UAVs in uncertain environments. Compared to Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), GWO, and MP-GWO algorithms, the NI-GWO algorithm can achieve the optimal fitness value and has significant advantages in terms of average path length, time, number of collisions, and obstacle avoidance capabilities.



Citation: Zhou, X.; Shi, G.; Zhang, J. Improved Grey Wolf Algorithm: A Method for UAV Path Planning. *Drones* **2024**, *8*, 675. <https://doi.org/10.3390/drones8110675>

Academic Editors: Yu Wu and Liguang Sun

Received: 15 September 2024
Revised: 26 October 2024
Accepted: 27 October 2024
Published: 14 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: multi-agent systems; UAV navigation; planning; intelligent control; artificial intelligence

1. Introduction

Since the 21st century, Unmanned Aerial Vehicles (UAVs) have garnered widespread attention due to their rapid development across various application fields [1]. UAVs have demonstrated immense potential and benefits in agriculture [2], logistics [3], national defence [4], environmental monitoring [5], aerospace [6], and emergency rescue [7]. Particularly in the military domain, UAVs have become the mainstream direction for developing future advanced weapon systems due to their superior mobility and flexibility [8]. With the rapid advancement of computer and electronic information technology, significant progress has been made in UAV technology [9,10], significantly promoting the evolution of military missions towards diversification and autonomy [11].

In modern warfare, a single UAV may reveal limitations in combat capabilities in the complex and changing battlefield environment, as researchers have gradually shifted their focus from single UAV operations to UAVs collaborative operations [12]. This shift not only signifies a leap at the technological level but also represents innovation at the tactical and strategic levels.

It is worth noting that before UAVs execute autonomous manoeuvring decisions and enter the combat zone, they must fly from the takeoff point to the mission area, avoiding enemy threat areas and devising precise flight plans to reach the predetermined

destination [13] safely. Formulating this flight plan does not require direct intervention by operators. Still, it relies on real-time environmental information and potential threats to autonomously guide the UAV formation to reach the mission target point successfully. UAV path planning is a crucial step before successfully entering the combat area; enhancing the capability of UAV path planning means adapting more efficiently to the changing battlefield environment, achieving higher combat efficiency and adaptability [14]. Therefore, in-depth research and optimisation of UAV path planning methods are particularly critical, as they promote the development of UAV technology and provide new perspectives for the innovation of future aerial combat strategies.

1.1. Related Work

UAV path planning is a pivotal technology determining UAVs' optimal flight paths and trajectories within a preset flying environment. The core objective is to enhance the efficiency and safety of flight missions. A key challenge in this process is efficiently avoiding obstacles and optimising critical flight indicators, including flight distance, time cost, and energy consumption [15]. Through designed path planning, UAVs can reduce the risk of collision, improve energy use efficiency, enhance the success rate of missions, and maintain flight stability in a variable aerial environment [16].

In this field, researchers have developed a variety of algorithms, including but not limited to metaheuristic algorithms such as Artificial Potential Fields (APF) [17] and Simulated Annealing (SA) [18], random sampling-based strategies like Rapidly-exploring Random Tree (RRT) [19] and Rapidly-exploring Random Tree Star (RRT*) [20], and swarm intelligence optimisation algorithms such as Ant Colony Optimization (ACO) [21] and Artificial Bee Colony (ABC) [22].

Regarding metaheuristic algorithms, Yang et al. [23] proposed the AAPF* algorithm, which combines the A* search algorithm with the APF method, aiming to enhance the safety of autonomous vehicles and ensure smoother global path planning, especially for vehicle cornering constraints. Deng et al. [24] proposed a path planning method that combines an improved Beetle Antennae Search (BAS) algorithm with SA to address the issue of poor real-time performance of UAV path planning algorithms in urban low-altitude logistics. Although heuristic optimisation algorithms have the advantages of fast planning speed, good parallelism, and ease of operation, they are prone to falling into local optimal solutions and heavily rely on heuristic information, making the algorithm's complexity, universality, and convergence speed challenging to determine [25].

For random sampling strategies, Noreen et al. [26] provided a comprehensive review, summarising the application and development of path planning methods based on the RRT* algorithm in the field of mobile robots. However, the RRT* algorithm has limitations in convergence speed. To address this issue, Aslan et al. [27] developed a Goal Distance-based RRT* (GDRRT*) approach that performs intelligent sampling while considering the goal distance. Subsequently, the authors utilised Particle Swarm Optimization (PSO) to shorten the paths discovered by GDRRT* (resulting in PSO-GDRRT*). BiLSTM-PSO-GDRRT* offers exceedingly rapid path planning suitable for real-time UAV applications. Nevertheless, the paths generated by this algorithm may not be optimal, and the complexity of the problem increases exponentially with the expansion of the search space [28].

Another type is swarm intelligence optimisation algorithms, and more and more research is committed to achieving a balance between the convergence and search capabilities of the algorithms [29]. Zhang et al. [30] proposed an improved SSA for workshop inspection path planning, demonstrating better convergence ability and optimisation performance. Although the swarm above intelligence optimisation algorithms have performed well in many application scenarios, they often face the challenge of falling into local optimal solutions [31].

Based on the above content, Table 1 has been compiled to provide an overview of the algorithms' inherent limitations.

Table 1. Limitations of the optimization algorithms discussed.

Algorithms	Limitations
Metaheuristic algorithm	Prone to getting trapped in local optima.
Sampling-based methods	High memory consumption and slow convergence rates as the search space expands.
Swarm intelligence optimisation algorithm	The parameters are susceptible, convergence is slow, and additional iterations may be required to ensure convergence.

To address the shortcomings above, Mirjalili, inspired by the hunting strategy of grey wolves, proposed the Grey Wolf Optimizer (GWO) algorithm [32]. The algorithm simulates the social behaviours and hierarchical structures of grey wolves. The bionic structure of the GWO algorithm makes it relatively easy to implement in practical applications and demonstrates good flexibility in dealing with various optimisation problems [33]. The GWO algorithm has been applied in many engineering and control fields, including but not limited to load frequency control [34], feature selection [35], and the resolution of UAV path planning [36]. However, some views point out that the traditional GWO intelligent optimisation algorithm may face challenges in certain situations, such as relatively slow convergence speed [37]. Therefore, it is necessary to improve and optimise the algorithm further.

Many studies on improving the GWO algorithm in the field of UAV path planning mainly focus on optimising path planning algorithms for individual UAVs [38]. These studies concentrate on enhancing the path efficiency and adaptability of a single UAV during mission execution, but do not consider the complexity and dynamics of UAVs collaborative operations.

Rao et al. [39] proposed an enhanced multi-strategy collaborative GWO algorithm, NOGWO. NOGWO bolsters algorithmic performance through random walk strategies, oppositional learning models, and innovative convergence factors. Chen et al. [40] proposed an optimization algorithm for UAV path planning, GWO-APF, which integrates the GWO and the APF method, aiming to enhance the efficiency and safety of path planning.

Presently, the application of the GWO algorithm based on UAVs in path planning methods is relatively limited, indicating that there is still significant room for development and potential opportunities for innovation in this area. Sun et al. [41] explored the application of UAVs across various fields, proposing the Improved MP-GWO algorithm. The research, however, does not delve into the complexities of three-dimensional path planning.

When UAVs are in motion, they inevitably encounter dynamic obstacles, necessitating effective avoidance maneuvers. The Dynamic Window Approach (DWA) [42] is a typical algorithm for local obstacle avoidance. The core principle of DWA is to evaluate potential velocities to determine the most suitable trajectory for the UAV in the short term.

Jorge Bes et al. [43] introduced a method for UAVs to safely and efficiently navigate autonomously and plan paths in complex and confined spaces. The system integrates a Rapidly-exploring RRT* global planner with a new reactive planner, DWA-3D, to accomplish navigation in complex scenarios. Wang et al. [44] improved upon the DWA algorithm by incorporating electrostatic potential energy theory and designs a dynamic adjustment function for velocity parameters. Experiments have demonstrated that the enhanced DWA algorithm can effectively address the path planning issues for UAVs in areas dense with local obstacles. Li et al. [45] proposed an improved A* global planning algorithm, combined with a sliding window local planning method, to design a hybrid robot obstacle avoidance path planning algorithm. The results show that this algorithm enables the planned path to effectively circumvent dynamic obstacles and accurately reach the target point. From the aforementioned literature, it is evident that the DWA algorithm has significant advantages in avoiding dynamic obstacles, path planning efficiency, and model adaptability. Therefore, this paper employs the DWA algorithm to accomplish dynamic obstacle avoidance tasks.

Building on the enhancement proposals for the GWO algorithm presented in the literature, this study endeavors to deeply refine the GWO algorithm. The goal is to develop an algorithm better tailored to the demands of UAV path planning and capable of adapting to more intricate three-dimensional terrain environments. The complexity of UAV path planning is markedly increased, which is primarily reflected in the following aspects: First, the planning space is more extensive, and the environment is more complex, involving a variety of spatial obstacles and potential threats. Second, it is necessary to design a reasonable cost function to meet more constraints, ensuring synchronization in both time and space. Lastly, the UAV path planning strategy must be adaptable to accommodate diverse battlefield environments. The optimization efforts of this study aim to enhance the algorithm's adaptability and robustness, ensuring its effectiveness in devising safe and efficient path planning schemes for UAVs in variable three-dimensional terrains.

1.2. Paper Contribution

This paper focuses on developing an enhanced GWO-based UAV path planning model. The model simulates the process where UAVs plan paths during the strategic phase to reach combat zones. By designing a cost function model, UAVs can effectively accomplish path planning tasks. Nonlinear functions are introduced to adjust convergence parameters, and the fitness values of grey wolves are utilized to update position parameters within the algorithm, enhancing its performance in exploring unknown environments. Integration with the DWA algorithm ensures good safety performance even in the presence of moving obstacles. The study's results show that, compared to other algorithms, this method reduces path length and planning time, significantly lowering the iteration convergence value and the number of iterations, thus improving the efficiency of UAV path planning. This paper introduces a nonlinear convergence factor mechanism, dynamic position updates based on state weights, and a comprehensive trajectory fitness function considering different cost functions, merging the DWA algorithm to guide UAV swarms more effectively to target points, demonstrating superior global path planning and dynamic obstacle avoidance capabilities. The specific research content of this paper is illustrated in Figure 1.

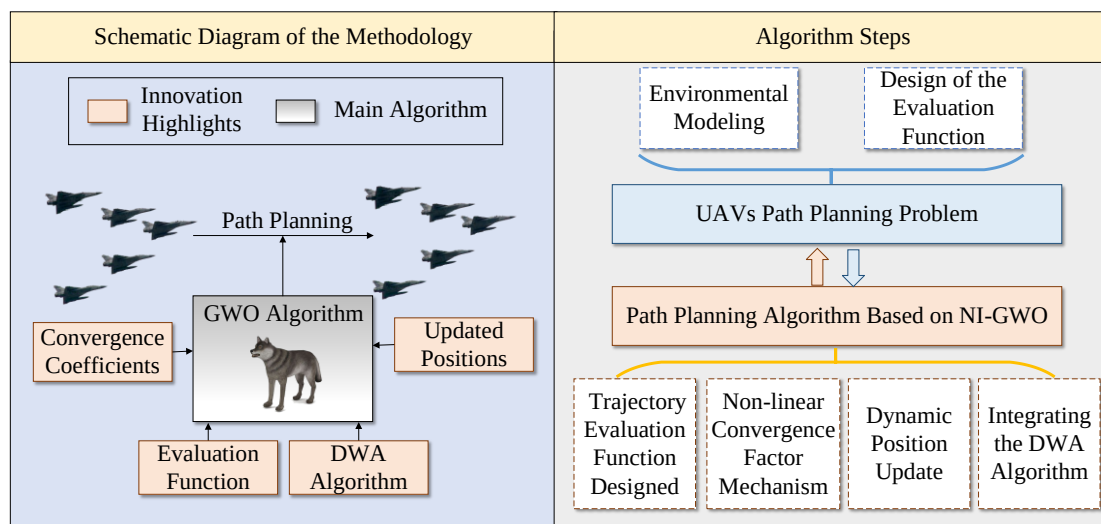


Figure 1. Content block diagram of UAV co-operative path planning model based on improved grey wolf algorithm.

The main innovations of this paper are as follows:

- A co-operative UAV path planning model is designed, which includes an environmental model, a timestamp segmentation model, a cost function model, and a fitness function model.
- Improvements are made to the GWO algorithm to address its slow convergence speed and tendency to get stuck in local optima. The convergence coefficients are optimized

using nonlinear functions, and the position update equation in the improved Grey Wolf Optimizer is modified based on the fitness of different grey wolves to design weight proportionality factors.

(c) Integration with the DWA algorithm allows the algorithm to not only meet the requirements for avoiding static obstacles but also to exhibit good safety performance when facing moving obstacles. It is suitable for the ever-changing real-world environment.

The rest of the chapters in this paper are organized as follows: The second part establishes the UAV path planning model, which includes the environmental model, timestamp segmentation model, cost function model, and fitness function model. The third part introduces NI-GWO, which encompasses the nonlinear optimization algorithm's convergence factor mechanism, position update equation, an enhanced fitness function model for GWO, and a model integrating the DWA algorithm. The fourth part presents experimental simulations that simulate and compare the UAV path planning capabilities of the proposed NI-GWO algorithm with those of PSO, ABC, GWO, and MP-GWO algorithms. Lastly, the fifth part is the conclusion, which summarizes the entire paper.

2. UAV Path planning Model

UAV path planning aims to generate paths for each UAV that meet requirements for safety, short range, and co-operation. Safety describes the need to avoid obstacles and no-fly zones in complex environments. Short range implies less fuel and energy consumption.

The generated target trajectories must satisfy the following constraints:

(a) Maximum yaw angle: UAVs can only turn within the maximum yaw angle range determined by their horizontal maneuverability.

(b) Maximum pitch angle: UAVs can only turn within the maximum pitch angle range determined by their vertical maneuverability.

(c) Maximum and minimum flying altitude: to ensure the safety of UAVs, their flying altitude should be greater than a predetermined minimum height and less than a maximum height.

(d) No collisions: considering the size of the UAVs, they must not collide with terrain, obstacles, or other UAVs in the scene when flying along the planned path.

Only paths that meet the above constraints are feasible. Furthermore, with the premise of path feasibility, the path should be optimized as much as possible. The optimization goals are to make the path shorter, height fluctuations smaller, turns fewer, and to avoid detection by radar.

2.1. Environmental Model

2.1.1. Terrain Model

In this study, the first step is to discretize the three-dimensional geographical space to enable efficient path planning within the planning area. This discretization process is accomplished by dividing the planning area into multiple adjacent cubic cells of uniform size. The specific operation involves searching within the planning area and orderly identifying each waypoint based on the preset number of flight waypoints, then, starting from the starting point, sequentially connecting these waypoints until the endpoint to construct a coherent flight trajectory. The model is shown in Figure 2, where a total of M UAVs fly from the starting point to the destination point, with the path including D intermediate nodes, and the entire path is divided into $D + 1$ segments. The position of each current node is determined by co-ordinates, where $P_{m,n}$ represents the n -th node of the m -th UAV ($m \in M$), and $l_{sm,n}$ represents the length of the line segment connecting node $P_{m,n}$ to node $P_{m,n+1}$, as illustrated by the red line segment in Figure 2. S_m and G_m , respectively, represent the starting point and the target point of the m -th UAV. This method clearly defines and utilizes the planning area for subsequent path planning and analysis tasks. Considering the energy and battery life limitations of the UAVs, this directly affects the flight time and coverage range of the UAVs [46]. Therefore, the flight distance of the UAVs is limited, and the shortest flight path must be chosen under constraints, that is, to

find an optimal or suboptimal solution Ω in the solution space, where Ω is a set of UAV flight paths that avoid various threats. Ω can be represented as:

$$\Omega = (x, y, z), \text{ s.t. } \begin{cases} X_{\min} \leq x \leq X_{\max} \\ Y_{\min} \leq y \leq Y_{\max} \\ Z_{\min} \leq z \leq Z_{\max} \end{cases} \quad (1)$$

where $X_{\min}$, $X_{\max}$, $Y_{\min}$, and $Y_{\max}$ represent the horizontal boundaries, while $Z_{\min}$ and $Z_{\max}$ represent the lower limit and the highest limit at position (x, y) , respectively.

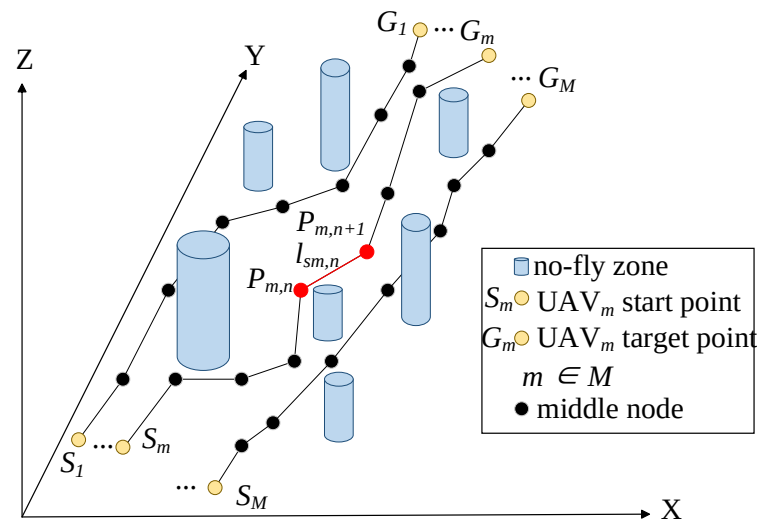


Figure 2. The UAV path planning diagram.

2.1.2. No-Fly Zone Model

In the real-world geographical environment, considering areas where flying is prohibited, such as densely populated areas and military control zones, establishing a corresponding no-fly zone model is crucial for ensuring UAVs' safe flight and compliance with relevant regulatory requirements. This study uses a cylindrical model to define these no-fly zones. In the matrix representation of the planning space, the matrix rows correspond to the horizontal direction, while the columns correspond to the vertical direction. For the i -th no-fly zone ($i \in n$), the co-ordinates of its centre are defined as (x_i, y_i) , and the radius of the no-fly zone is set to r_i . The height h of the no-fly zone is a preset fixed value, representing the maximum flight altitude of the UAV. Figure 2 illustrates this no-fly zone model with a cylindrical shape. According to these definitions, no-fly zones are identified with the letter **B** in the model.

$$\mathbf{B} = \begin{bmatrix} x_1 & y_1 & h_1 & r_1 \\ x_2 & y_2 & h_2 & r_2 \\ \dots & \dots & \dots & \dots \\ x_n & y_n & h_n & r_n \end{bmatrix} \quad (2)$$

2.1.3. Formation Model

In the military domain, aircraft, much like migratory birds, adopt various popular flight formations to conserve energy and reduce air resistance. Reference [47] tested how different formations affect the flight status of UAVs, providing us with common UAV formations seen in the military field. In this paper's UAV path planning experiments that alter UAV formations, we studied three different UAV formations. These formations are: V-shaped, Echelon, and Diamond-shaped, assuming a group consists of five UAVs, as shown in Figure 3.

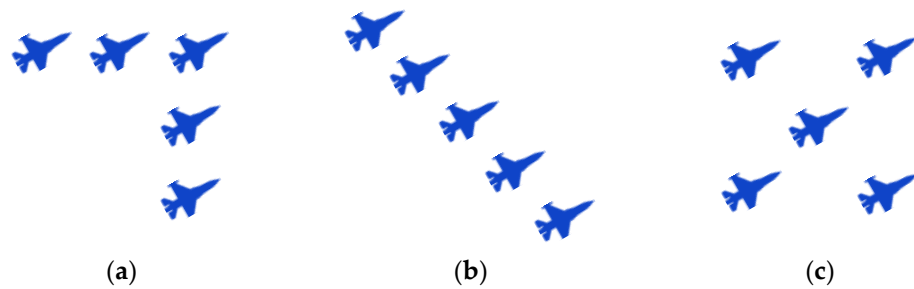


Figure 3. Different flight formations for a swarm of UAVs: (a) v-shaped; (b) echelon; (c) diamond-shaped.

2.2. Timestamp Segmentation Model

In UAV co-operative path planning, the computational burden of the algorithm increases sharply with the number of UAVs, which may lead to slower real-time response. Complex environments may cause the algorithm to process more data and information, further increasing computational complexity. To overcome these challenges, we adopt a timestamp segmentation model to optimize the UAV path planning model, which provides a common time reference, thereby simplifying the calculation of path co-ordination costs.

Assume that all UAVs share the same takeoff time T_t and arrival time T_a . Given the number of path points D , define the timestamp t_s as $t_s = (T_a - T_t)/D$, where t_s is calculated by Formula (4). The path length of the m -th UAV at the n -th timestamp is represented by the red line segment in the figure. The motion of the UAV is composed of two parts: the velocity component $V_{m,n} = [V_{m,n,x}, V_{m,n,y}]$ and the position component $P_{m,n} = [P_{m,n,x}, P_{m,n,y}]$, where $V_{m,n,x}$ and $V_{m,n,y}$ represent the velocities in the X-axis and Y-axis directions, respectively. $P_{m,n}$ is updated based on $V_{m,n}$ as follows:

$$P_{m,n} = P_{m,n-1} + V_{m,n} \cdot t_s \quad (3)$$

where $V_{m,n}$ and $P_{m,n}$ are the horizontal velocity and horizontal position of the m -th UAV at the n -th timestamp, respectively.

The range of the velocity component and the timestamp is determined by the UAV's flight constraints, such as the airspeed range $[V_{\min}, V_{\max}]$. Assuming the straight-line distance between the takeoff point and the target point of the m -th UAV is l_m , the range of the timestamp t_s is:

$$t_s \in \bigcap_{m=1}^M [t_{m,\min}, t_{m,\max}] \quad (4)$$

$$t_{m,\min} = \frac{l_m}{V_{\max} D}, \quad t_{m,\max} = \frac{l_m}{V_{\min} D}$$

In the given context, M represents the number of UAVs, and $\cap$ denotes the intersection. After selecting the timestamp t_s , we can easily determine the common arrival time $T_a = Dt_s + T_t$.

Assuming the flying height z of the UAVs is initially not considered, the starting point O_m pointing to the destination point E_m serves as the X-axis of the UAV's local co-ordinate system $X_m O_m Y_m$. The UAV's local co-ordinate system is shown in Figure 4. In the local co-ordinate system, the UAV's velocity in the X-axis direction is constant, i.e., $\bar{V}_{m,n,x} = l_m / (Dt_s)$.

Then, an optimization algorithm is used to search for the optimal configuration of the height component and the Y-axis velocity component $\bar{V}_{m,n,y}$. Afterward, based on the determined timestamp t_s , the co-ordinates of the path point $\bar{P}_{m,n} = [\bar{P}_{m,n,x}, \bar{P}_{m,n,y}]$ in the local co-ordinate system can be determined. Subsequently, according to $\bar{P}_{m,n}$ and the corresponding height z , the three-dimensional co-ordinates of the path point in the global co-ordinate system can be obtained, thus obtaining the required path. Considering the maximum yaw angle $\psi_{\max}$ and the X-axis velocity component $\bar{V}_{m,n,x}$, we can determine the range of the Y-axis velocity component, i.e., $\bar{V}_{m,n,y} \in [-\bar{V}_{m,n,x} \tan \psi_{\max}, \bar{V}_{m,n,x} \tan \psi_{\max}]$,

length, the point at 0.75 of the path length, and the ending point. If the path segment falls into a no-fly zone, then the threat cost is calculated as follows:

$$f_m^{threat}(L) = \sum_{n=1}^D \frac{l_{sn,m}}{5} \cdot \sum_{i=1}^N \frac{r_i}{10} \cdot \left(\frac{1}{d_{0,i}^n} + \frac{1}{d_{0.25,i}^n} + \frac{1}{d_{0.5,i}^n} + \frac{1}{d_{0.75,i}^n} + \frac{1}{d_{1,i}^n} \right) \quad (8)$$

where N represents the number of no-fly zones, $d_{0.25,i}^k$ is the distance from the i -th no-fly zones center to the point 0.25 along the segment, and r_i is the radius of the i -th no-fly zones.

2.3.3. Flight Characteristic Cost

An effective path should not only avoid various threats but also meet the physical constraints of the UAV, such as the yaw angle $\psi_{m,n}$ and pitch angle $\phi_{m,n}$. Excessive yaw and pitch angles can bring unnecessary operational risks to the UAV. We hope the UAV's trajectory meets the basic conditions with as small yaw and pitch angles as possible [48]. The cost function is shown as follows:

$$f_m^{yaw}(L) = \begin{cases} \sum_{n=1}^D (\psi_{m,n} - \psi_{\max}) & \psi_{m,n} > \psi_{\max} \\ 0, & \psi_{m,n} < \psi_{\max} \end{cases} \quad (9)$$

$$f_m^{pit}(L) = \begin{cases} \sum_{n=1}^D (\phi_{m,n} - \phi_{\max}), & \phi_{m,n} > \phi_{\max} \\ 0, & \phi_{m,n} < \phi_{\max} \end{cases} \quad (10)$$

$$\psi_{m,n} = \left| \arctan\left(\frac{y_{m,n+1} - y_{m,n}}{x_{m,n+1} - x_{m,n}}\right) \right|, \quad n = 1, 2, 3, \dots, D-1 \quad (11)$$

$$\phi_{m,n} = \left| \arctan\left(\frac{z_{m,n+1} - z_{m,n}}{x_{m,n+1} - x_{m,n}}\right) \right|, \quad n = 1, 2, 3, \dots, D-1 \quad (12)$$

where $\psi_{m,n}$ and $\phi_{m,n}$ represent the yaw and pitch angles of the n -th node of the m -th UAV, respectively, as shown in Figure 5; $\psi_{\max}$ and $\phi_{\max}$ represent the maximum yaw and pitch angles, respectively.

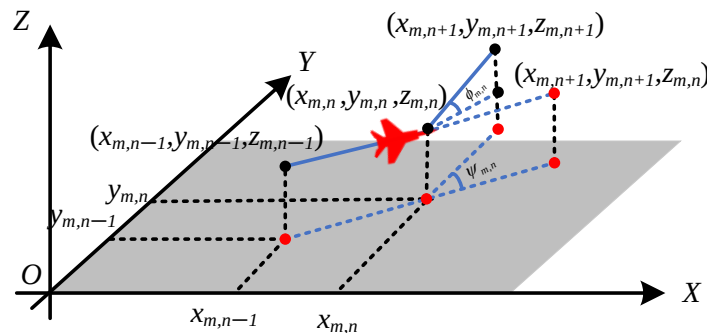


Figure 5. Schematic diagram of pitch and heading angles for the m -th UAV.

When the UAV's flying height is close to the ground height, the risk of the UAV hitting the ground increases. When the flying height is too high, the UAV can be easily detected by enemy radar in special missions, thus failing to achieve the original intention of minimizing the flight path. The UAV's flying path is expected to be at an appropriate height. The cost function related to the flying height $f_m^{height}(L)$ is shown as follows:

$$z_{\min}^{m,n}(x, y) + \Delta h \leq z_{m,n}(x, y) \leq z_{\max} \quad (13)$$

$$f_m^{height}(L) = \begin{cases} \sum_{n=1}^D (z_{m,n}(x, y) - z_{\min}^{m,n}(x, y) - \Delta h), & z_{\min}^{m,n}(x, y) \leq z_m(x, y) \leq z_{\max} \\ \inf, & \text{otherwise} \end{cases} \quad (14)$$

where $z_{m,n}(x, y)$ is the height of the m -th UAV at the n -th waypoint; $z_{\min}^{m,n}(x, y)$ is the ground height at the n -th waypoint for the m -th UAV; $z_{\max}$ is the maximum height allowed for the UAV to fly; and Δh represents the safe distance from the ground.

Therefore, the flight characteristic cost for the UAV is calculated as follows:

$$f_m^{chara}(L) = f_m^{yaw}(L) + f_m^{pit}(L) + f_m^{height}(L) \quad (15)$$

2.3.4. Co-Ordination Cost

When the UAVs fly along their paths, they must not collide with each other, otherwise the co-ordination cost will be extremely high. The co-ordination cost for the m -th UAV and the i -th UAV ($i \in M$) at the n -th waypoint is represented as:

$$f_m^{coor}(L) = \sum_{m=1}^M \sum_{i=m+1}^M \sum_{n=1}^D f_{m,i,n}^{coor}(L) \quad (16)$$

$$f_{m,i,n}^{coor}(L) = \begin{cases} Q & , \|P_{m,n} - P_{i,n}\| \leq \delta_{uav} \\ 0 & , \text{otherwise} \end{cases} \quad (17)$$

where $P_{m,n}$ is the position of the m -th UAV at the n -th waypoint; δ_{uav} is the safe distance between UAVs; M is the number of UAVs.

2.4. Fitness Function Modeling

The quality of the path is usually evaluated by the fitness function. Considering the above cost functions, a fitness function to evaluate the path quality can be defined as:

$$\text{Fit}(L) = w_1 \sum_{m=1}^M f_m^{length}(L) + w_2 \sum_{m=1}^M f_m^{threat}(L) + w_3 \sum_{m=1}^M f_m^{chara}(L) + w_4 \sum_{m=1}^M f_m^{coor}(L) \quad (18)$$

where w_1 , w_2 , w_3 , and w_4 are relative weight factors, and they satisfy $w_1 + w_2 + w_3 + w_4 = 1$; the numerical values of the relative weight factors should change according to different scenarios because each loss function has different importance for the overall path evaluation [48].

3. UAV Path Planning Model Based on Improved GWO Algorithm

3.1. Grey Wolf Optimizer Algorithm

The GWO is a significant method in swarm intelligence optimization strategies, inspired by the social hierarchy and collective hunting behavior of grey wolves. At the top of the social structure of a wolf pack is the Alpha wolf, which holds the highest leadership position, followed by the Beta wolf as the second tier, and the Delta wolf as the third tier. At the bottom of the social hierarchy are the Omega wolves. In the Grey Wolf Optimizer, these three higher tiers correspond to the best, second best, and third best solutions, each demonstrating to the Omega wolves how to explore towards the target. During the optimization process, the pack of wolves continuously adjusts the positions of the Alpha, Beta, Delta, and Omega wolves with the aim of achieving the optimization goal.

The hunting process of a wolf pack consists of four basic steps: encircling, hunting, attacking, and scouting. The hunting behavior begins with tracking and accurately locating the prey, and then, under the guidance of the three leaders, the Alpha, Beta, and Delta wolves, the rest of the pack members adjust their positions based on these lead wolves.

This strategy allows the wolf pack to gradually close the distance with the prey until the final capture is made.

(a) Encircling

During hunting, grey wolves encircle their prey. To quantitatively simulate the encircling behaviour of grey wolves, the following equations are proposed:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(i) - \vec{X}(i) \right| \quad (19)$$

$$\vec{X}(i+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (20)$$

Here, i represents the current iteration number, $\vec{X}_p$ is the position vector of the prey, and $\vec{X}$ is the position vector of the grey wolf. And $\vec{C}$ are coefficient vectors, calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (21)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (22)$$

Here, the convergence factor $\vec{a}$ decreases linearly from 2 to 0 during the iteration process for each component, while r_1 and r_2 are random vectors within the range $[0, 1]$. The equation for updating the parameter a is as follows:

$$\vec{a} = 2 - i \left(\frac{2}{I} \right) \quad (23)$$

Here, i represents the latest iteration number, and I is the total number of iterations.

Grey wolves can update their positions based on the location of the prey. By changing the values of vectors $\vec{A}$ and $\vec{C}$, other positions around the best agent can be identified near the current position. It is worth noting that, due to the random vectors $\vec{r}_1$ and $\vec{r}_2$, wolves can reach any position between the points. Therefore, grey wolves can use Equations (19) and (20) to update their positions at any location surrounding the prey.

(b) Hunting

In order to mathematically simulate the hunting strategy of grey wolves, this model assumes that Alpha, Beta, and Delta wolves have relatively more information about the possible location of the prey, therefore, always using the search results of the top three wolves. All other search agents, including Omega wolves, are instructed to rearrange. In the hunt, the formulas of the Grey Wolf Optimizer are as follows:

$$\begin{cases} \vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \end{cases} \quad (24)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha \right) \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \left(\vec{D}_\beta \right) \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \left(\vec{D}_\delta \right) \end{cases} \quad (25)$$

$$\vec{X}(i+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (26)$$

The final positioning of the agents is a random process that calculates based on the current positions of the Alpha, Beta, and Delta wolves.

(c) Attack

Grey wolves initiate an attack when the prey stops moving, concluding the hunt. To simulate this analytical approach towards the prey, the value of a is reduced, which also decreases the fluctuation range of $\vec{A}$. Specifically, a decreases from 2 to 0 during the iteration process, randomly reducing within the range of $[-2a, 2a]$. When the random value of $\vec{A}$ is between $[-1, 1]$, the wolf's future position can be anywhere between its current position and the prey (target solution). When $|A| < 1$, the pack is forced into the process of charging at the prey.

Grey wolves in the GWO can update their positions based on the positions of Alpha, Beta, and Delta wolves. Therefore, they can use the proposed operators to attack the prey. However, the GWO method is prone to falling into local optimal solutions when using these operators.

(d) Search

To simulate the diffusive characteristics of exploratory behavior, the GWO algorithm employs a method that assigns random values greater than 1 or less than -1 to parameters, thereby prompting the search pack to move in a direction away from the prey.

Another aspect of GWO worth exploring is the $\vec{C}$ vector, which has random values of $[0, 2]$. This part achieves a random enhancement or weakening of the prey's influence distance by assigning random weights to the prey. Unlike $\vec{A}$, the decrease of $\vec{C}$ is not linear.

Finally, a random population of grey wolves is produced through the GWO algorithm. Alpha, Beta, and Delta wolves calculate the potential locations of the prey through iteration, with each possible response altering the distance between it and the prey, with the value of a decreasing from 2 to 0. The GWO algorithm completes the entire process when the termination criteria are met.

3.2. Improved Grey Wolf Optimizer Algorithm

The GWO algorithm uses the position update equation to find the optimal solution. However, it has the following drawbacks: the GWO algorithm divides the wolves into four ranks, and these wolves adjust their positions to hunt prey based on Equation (26). Solving multimodal problems may cause the algorithm to stagnate at local optima [49]. Additionally, during the iteration process, because the solution of the Delta wolf is poor, when the Alpha wolf adjusts its position based on the information from all three wolves, it may lead the Alpha wolf in the wrong direction. Similarly, the Beta wolf may also be misguided. Therefore, Equation (26) assigns the exact weight of $1/3$ to the Alpha, Beta, and Delta wolves, which is unreasonable [50].

Hybrid methods can effectively combine the strengths of algorithms while discarding their weaknesses. A wealth of research has shown that compared to single algorithms, hybrid algorithms perform better in solving a variety of optimisation problems [51]. This concept has been widely applied in traditional evolutionary algorithms, but its application in swarm intelligence algorithms, especially in GWO algorithms, is relatively rare. To fill this research gap and expand the application range of the algorithm, this study first introduces a nonlinear control mechanism to optimise the convergence factor. Then, it improves the position update equation of (26) according to the different social statuses of wolves in the hierarchy to enhance the algorithm's exploration ability and enable it to escape from local optima, namely by using a nonlinear factor to improve the GWO Algorithm (Nonlinear Improved Grey Wolf Optimizer Algorithm, NI-GWO). The specific improvements are as follows:

(a) Convergence Factor Mechanism for Non-linear Optimization Algorithms

In the GWO algorithm, the parameter $\vec{A}$ is the disturbance factor that guides the algorithm in searching for the optimal solution, and the GWO determines the exploitation

and exploration capabilities of the algorithm based on the transformed value of $\vec{A}$. From Equations (21) and (23), it can be seen that the transformed value of the parameter $\vec{A}$ is determined by the convergence factor $\vec{a}$, which linearly decreases from 2 to 0 as the number of iterations increases. Since the search strategy does not always vary linearly, the linear reduction of the convergence factor does not fully reflect the complex search dynamics of the algorithm in dealing with multimodal environments. Therefore, to enhance the search capability, this paper introduces the following nonlinear control mechanism in GWO:

$$\vec{a} = \vec{a}_{\text{initial}} \cdot \frac{1}{1 + e^{\frac{10i-5I}{I}}} \quad (27)$$

The change curve is shown in Figure 6.

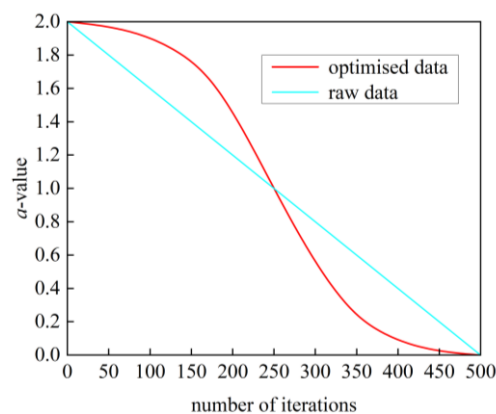


Figure 6. Variation curve of convergence factor.

Here, $\vec{a}_{\text{initial}}$ is the initial value of the convergence factor $\vec{a}$, i represents the current number of iterations, and I represents the maximum number of iterations. From Figure 6, it can be seen that the change curve based on the Sigmoid function correction gradually decreases at the beginning of the iteration, allowing the convergence factor $\vec{a}$ to maintain a higher value for a longer time, thereby improving the global search capability of the algorithm in the early stage. Due to the rapid decline of the value of $\vec{a}$, the disturbance factor A can also maintain a lower value in the later stages of iteration, which increases the local search accuracy of the algorithm and balances its global and local search capabilities.

(b) Improved GWO Position Update Equation

In GWO, the top three wolves have different social statuses. Beta and Delta wolves search around the Alpha wolf to better utilise the information of the leading wolves. In contrast, the Alpha wolf explores independently, calculating weights based on the wolves' fitness values. To enhance the algorithm's development capability, more weight is assigned to the best wolf, as follows:

$$\begin{cases} \vec{X}_\alpha = \vec{X}_\alpha + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_{r_1} - \vec{X}_{r_2}) \\ \vec{X}_\beta = \vec{X}_\beta + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_\alpha - \vec{X}_\beta) \\ \vec{X}_\delta = \vec{X}_\delta + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_\alpha - \vec{X}_\delta) \end{cases} \quad (28)$$

Here, $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ represent the positions of the Alpha, Beta, and Delta wolves, respectively. $\vec{a}$ linearly decreases from 2 to 0, and r_1 and r_2 are different integers ranging from 1 to the number of wolves, where $(r_1 \neq r_2)$, $\text{rand} \in [0, 1]$ is a random number. The rest of the wolves follow the position update equation. However, assigning the same

weight to $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ is unfair. The weights depend on their fitness values. The better the wolf's fitness, the more information it has, and the more weight it should be given. The weights can be calculated based on the wolves' fitness values. Usually, the optimisation problem is to find minimisation. The calculation method of the weights is as follows:

$$(w_\alpha, w_\beta, w_\delta) = \left(f(\vec{X}_\alpha) + f(\vec{X}_\beta) + f(\vec{X}_\delta) \right) \times \left(\frac{1}{f(\vec{X}_\alpha)}, \frac{1}{f(\vec{X}_\beta)}, \frac{1}{f(\vec{X}_\delta)} \right) \quad (29)$$

$$\vec{X}'(t+1) = \frac{w_\alpha \times \vec{X}_1 + w_\beta \times \vec{X}_2 + w_\delta \times \vec{X}_3}{w_\alpha + w_\beta + w_\delta} \quad (30)$$

Here, $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ are the positions of the Alpha, Beta, and Delta wolves, respectively. $f(\vec{X}_\alpha)$, $f(\vec{X}_\beta)$, and $f(\vec{X}_\delta)$ are their fitness values, w_α , w_β , and w_δ are their weights, and the three vectors $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ from Equations (25), (29) and (30) are responsible for calculating the weights assigned to the top three wolves according to their fitness. This improvement enhances the algorithm's development utilisation rate by assigning more weight to the best wolves (instead of the same weight of 1/3 in traditional GWO). Since the Alpha wolf knows the most about the prey's location, it is given the highest weight, the Beta wolf is in second place, and the Delta wolf has the lowest weight.

The algorithm flowchart for solving the model using the proposed NI-GWO algorithm is illustrated in Figure 7.

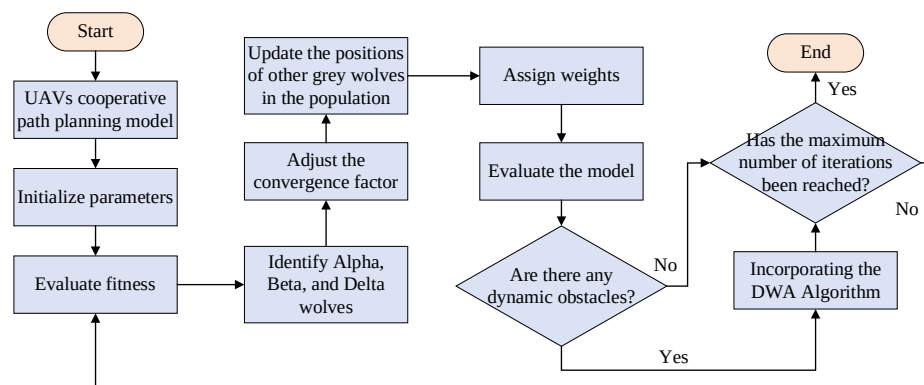


Figure 7. NI-GWO algorithm flowchart.

3.3. Local Path Planning of DWA Algorithm

The improvements proposed earlier only considered static environments, that is, environments without dynamic obstacles. However, in UAV path planning tasks, the presence of some unknown dynamic obstacles is inevitable. Due to the strong real-time performance and low complexity of the Dynamic Window Approach (DWA) algorithm, and considering the control constraints of the UAV motion model, it is necessary to generate a collision-free path in dynamic environments and directly output the corresponding linear and angular velocities [26]. In this paper, we integrate the DWA algorithm to avoid dynamic obstacles [27].

For UAVs, sampled velocities include both linear and angular velocities. However, there are some restrictions in the sampling process:

(a) The UAV is subject to its own velocity limitations. V_r is the set of UAV linear and angular velocities adapted to the maximum dynamic window range:

$$V_r = \{(v, \omega); v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (31)$$

where $v_{\max}$ and $v_{\min}$ represent the maximum and minimum linear velocities of the UAV, respectively, and $\omega_{\max}$ and $\omega_{\min}$ represent the maximum and minimum angular velocities of the UAV, respectively.

(b) The UAV has maximum acceleration and deceleration limits. Within the sampling time, there exists a dynamic window. The velocities within the dynamic window are the actual speeds that the UAV can achieve:

$$V_s = \left\{ (v, \omega); \begin{array}{l} v \in [v_c - \dot{v}_b \Delta t_s, v_c + \dot{v}_b \Delta t_s] \\ \omega \leq \sqrt{2 \text{dist}(v, \omega) \dot{\omega}_b} \end{array} \right\} \quad (32)$$

where v_c and ω_c are the current linear and angular velocities, respectively. $\dot{v}_b$ and $\dot{\omega}_b$ are the maximum linear and angular velocities within the dynamic window for the UAV, respectively. Δt_s is the sampling time.

(c) We combine different linear and angular velocities to obtain different sampling results. Thus, different sampling trajectories are generated. When the UAV detects obstacles, to ensure the safety of the UAV under maximum deceleration conditions, there is a range of UAV velocities to avoid obstacles. V_s is the set of linear and angular velocities of the UAV near the obstacle:

$$V_s = \left\{ (v, \omega); \begin{array}{l} v \leq \sqrt{2 \text{dist}(v, \omega) \dot{v}_b} \\ \omega \leq \sqrt{2 \text{dist}(v, \omega) \dot{\omega}_b} \end{array} \right\} \quad (33)$$

where $\text{dist}(v, \omega)$ is the shortest distance between the path and the obstacle.

The final velocity range is the intersection of the three sets. The dynamic window V_a is defined as follows:

$$V_a = V_r \cap V_d \cap V_s \quad (34)$$

Within the sampled set of velocities, there exists a set of feasible trajectories. Therefore, an evaluation function $G(v, \omega)$ is employed to assess each trajectory. The evaluation process is accomplished through the path length cost (Equation (6)), threat cost (Equation (8)), flight characteristic cost (Equation (15)), and co-ordination cost (Equation (16)) functions outlined in Section 2.3.

$$G(v, \omega) = \sigma \sum_{m=1}^M (a f_m^{\text{length}}(L) + b f_m^{\text{threat}}(L) + c f_m^{\text{chara}}(L) + d f_m^{\text{coord}}(L)) \quad (35)$$

Here, σ is a normalization function, and a , b , c , and d are the weights corresponding to the four cost functions.

Details of the differences between the traditional GWO algorithm and the NI-GWO algorithm designed in this paper are illustrated in the Figure 8, with each innovation point annotated in red to indicate the purpose of its improvement.

The NI-GWO algorithm, with its improved fitness function model, can find more suitable paths for each drone. Furthermore, the dynamic updating strategy for Alpha, Beta, and Delta wolves in NI-GWO significantly enhances the algorithm's potential to avoid local optima, addressing the issues of premature convergence and stagnation during path exploration. Moreover, introducing a new convergence factor in NI-GWO effectively regulates the balance between exploitation and exploration, expanding the search area and preventing premature convergence. By integrating the DWA algorithm, the NI-GWO algorithm endows drones with the capability to dynamically avoid obstacles, enabling it to effectively handle complex battlefield environments.

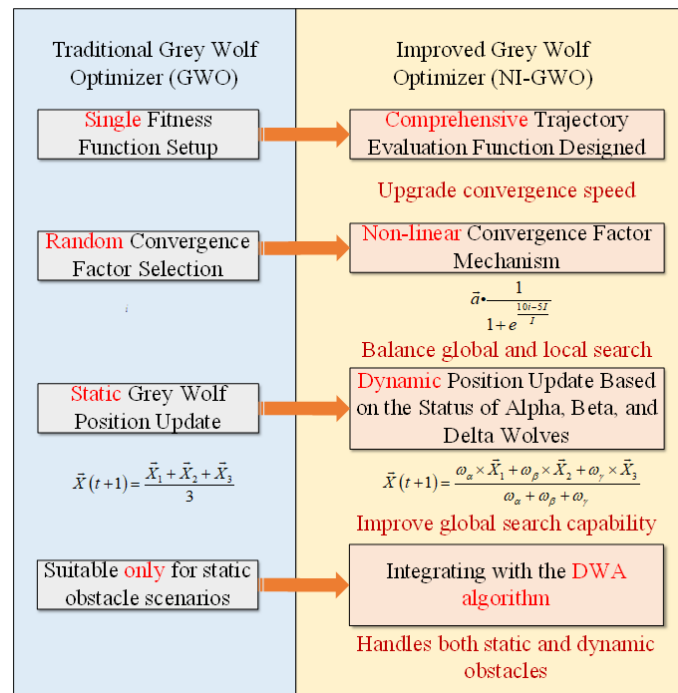


Figure 8. Detail of differences between GWO and NI-GWO algorithms.

3.4. Algorithm Complexity Analysis

In this section, the complexity assessment of the newly proposed NI-GWO algorithm is divided into four stages. The complexity analysis for each stage is detailed below.

First Stage: Initialization Phase. The population initializes the wolf pack for subsequent tasks, with a computational complexity of $O(N \cdot D)$. Here, N represents the size of the wolf pack, and D represents the dimensionality of the problem.

Second Stage: Fitness Calculation Phase. The objective function evaluation for each solution (wolf) requires $O(N)$ time.

Third Stage: Wolf Type Identification Phase. Determining the Alpha, Beta, and Delta wolves within the pack takes $O(N)$ time.

Fourth Stage: Position Update Phase. The position update of the wolves in NI-GWO requires $O(N \cdot D)$ time.

In summary, the total computational cost of the proposed NI-GWO algorithm equals $O(N \cdot D \cdot T)$, where T is the maximum number of iterations. Therefore, the computational complexity of the standard GWO and NI-GWO is the same.

4. Experimental Simulation and Results

4.1. Experimental Environment Setup

Based on the aforementioned algorithm design, simulation experiments were conducted on a laptop equipped with a 13th Gen Intel(R) Core(TM) i9-13900H processor and 16 GB RAM, running Windows 11 operating system, and MATLAB R2021b as the programming environment. The experiment utilized a map consisting of $1000 \times 1000 \times 20$ cubic cells, with obstacles modeled as cylinders. When setting up the scenario, it is assumed that the positions of the obstacles are known. When the center co-ordinates, radius, and height of the obstacles are known, they can be represented on the generated 3D map. The parameters related to the algorithm in this paper are set according to reference [41], as shown in Table A1. It is assumed that if a collision occurs or if the drone enters a no-fly zone, the trajectory generation for the drone will not be interrupted, however, every instance of collision and no-fly zone penetration will be recorded in detail. These records will be used for subsequent statistical analysis to assess the superiority of our enhanced algorithm in path planning tasks.

4.2. Comparison with Existing Algorithms

To evaluate the performance of the proposed algorithm in this paper, we selected the GWO and MP-GWO algorithms proposed in reference [41], as well as the ABC [52] and PSO [53] algorithms for comparison. The parameters of the algorithms were set according to the aforementioned references and the results of Experiment I in Section 4.3, as shown in Table A2. Each algorithm was run 20 times.

4.2.1. Algorithm Performance with Different Numbers of UAVs

To verify the performance of different algorithm models under varying numbers of UAVs, this experiment simulated path planning for 3, 5, and 7 UAVs in a fixed no-fly zone scenario. By changing the number of UAVs, it was verified whether the improved algorithm was effective in handling different numbers of UAVs and to demonstrate the superior performance of the algorithm in scenarios with different numbers of UAVs.

Figure 9 shows the convergence curves of the NI-GWO algorithm compared to the other algorithms. It can be observed that the NI-GWO algorithm proposed in this paper outperforms the other four algorithms in terms of fitness value convergence results. The statistical results are shown in Table 2. It can be observed from the charts that the NI-GWO algorithm has resolved the issue of local optima caused by excessively rapid convergence. The ABC and PSO algorithms are prone to fall into local optima due to premature convergence. The GWO and MP-GWO algorithms are overly dependent on the initial solution, resulting in the poorest convergence accuracy. Among the five algorithms, the NI-GWO algorithm is superior to the other algorithms in terms of average path length, computation time, number of times passing through no-fly zones, and number of collisions.

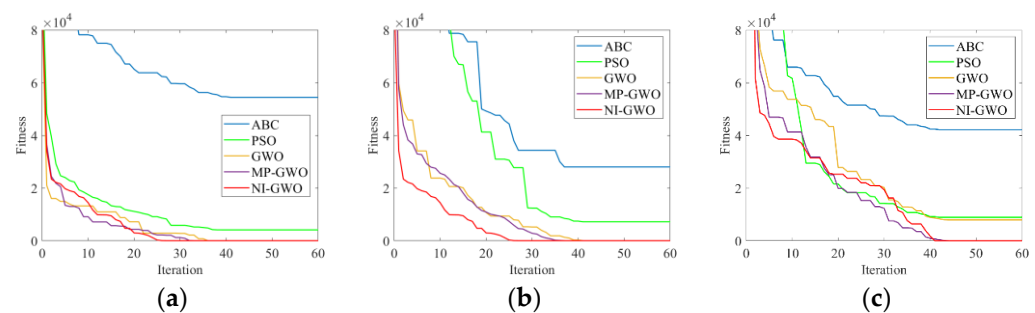


Figure 9. Fitness value graphs for UAV path planning: (a) 3 UAVs; (b) 5 UAVs; (c) 7 UAVs.

Table 2. Data results of five algorithms under different numbers of UAVs. The best results achieved among all algorithms are highlighted in bold.

UAV Count		ABC	PSO	GWO	MP-GWO	NI-GWO
11	Average path length	1303.48	1227.46	1203.40	1125.30	1095.43
	Average runtime (s)	4542.12	4003.4	3316.16	3583.52	3113.67
	Average no-fly zone crossings	0.19	0.17	0.04	0.03	0.01
	Average number of collisions	0.21	0.19	0.23	0.02	0.05
15	Average path length	1365.98	1320.64	1160.35	1117.56	1018.34
	Average runtime (s)	4735.01	4353.68	4090.96	4372.78	3189.37
	Average no-fly zone crossings	0.35	0.48	0.07	0.09	0.04
	Average number of collisions	0.32	0.28	0.24	0.13	0.03
19	Average path length	1393.48	1372.46	1249.71	1275.93	1209.23
	Average runtime (s)	6080.17	5895.43	3797.02	5741.34	4442.91
	Average no-fly zone crossings	0.62	0.56	0.65	0.56	0.25
	Average number of collisions	0.51	0.46	0.63	0.26	0.10

The results indicate that the proposed algorithm can meet the requirements of the path planning cost function, quickly generate reasonable flight paths, and satisfy the

prerequisites for obstacle avoidance. The algorithm is superior to other algorithms in terms of convergence speed, convergence accuracy, and computation time. Clearly, the algorithm is better than the other four algorithms.

4.2.2. Algorithm Performance in Different Static Obstacle Scenarios

To verify the path planning capabilities of the ABC, PSO, GWO, and MP-GWO algorithms when the path planning scenario changes, this experiment increased or decreased the number of no-fly zones in the basic path planning scenario. Simulations were conducted for 5 UAVs in scenarios with 11, 15, and 19 no-fly zones, recording and analyzing the effects of formation collaboration path planning to verify the superior performance of the proposed algorithm in different scenarios.

Figure 10 displays the convergence curves of the five algorithms in three scenarios. It can be observed that the NI-GWO algorithm demonstrates faster convergence speed and lower cost consumption compared to the other four algorithms. The NI-GWO algorithm ensures its reliability through the adjustment of its convergence factor and dynamic position update equation, achieving paths with lower costs. Among them, the ABC algorithm performed the worst, with the slowest convergence speed and the poorest convergence accuracy. The ABC algorithm is highly sensitive to parameters and requires high continuity in the search space. When facing complex high-dimensional optimization problems, it takes longer to find better solutions and is prone to falling into local optima, exhibiting relatively weak global optimization capabilities. Although the GWO and PSO algorithms also possess strong global search capabilities, their search process largely relies on the guidance of the global optimum. In complex or high-dimensional search spaces, as the complexity of the environment increases, this reliance leads to a higher likelihood of the algorithms falling into local optima in later stages, with the MP-GWO algorithm exhibiting slower convergence speed.

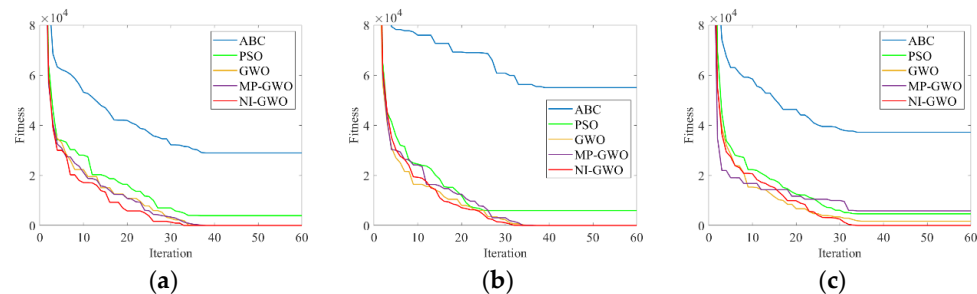


Figure 10. Fitness value graphs for UAV path planning: (a) 11 no-fly zones; (b) 15 no-fly zones; (c) 19 no-fly zones.

Table 3 presents the statistical results of the five algorithms over 20 independent iterations in three scenarios. It can be observed that the NI-GWO algorithm performs exceptionally well across all four scenarios, achieving the lowest path cost and the least time consumption compared to the other four algorithms. The smaller variance coefficient of the NI-GWO algorithm demonstrates its advantage in stability. The lower number of no-fly zone penetrations and collisions in the NI-GWO algorithm demonstrates its higher performance in planning paths within complex flight environments.

Table 3. Data results of five algorithms under different numbers of no-fly zones. The best results achieved among all algorithms are highlighted in bold.

UAV Count		ABC	PSO	GWO	MP-GWO	NI-GWO
11	Average path length	1401.23	1399.02	1318.92	1469.65	1178.80
	Average runtime (s)	5934.12	5732.01	5730.73	5899.78	4042.68
	Average no-fly zone crossings	0.45	0.51	0.19	0.11	0.06
	Average number of collisions	0.42	0.39	0.38	0.26	0.05

Table 3. Cont.

UAV Count		ABC	PSO	GWO	MP-GWO	NI-GWO
15	Average path length	1422.06	1421.01	1399.15	1479.32	1208.65
	Average runtime (s)	6722.76	6123.25	5984.46	6514.98	6001.99
	Average no-fly zone crossings	0.66	0.59	0.57	0.45	0.17
	Average number of collisions	0.60	0.51	0.45	0.33	0.12
19	Average path length	1471.56	1450.09	1444.32	1499.66	1352.49
	Average runtime (s)	8243.68	7777.19	6203.58	7801.10	6101.55
	Average no-fly zone crossings	0.78	0.66	0.63	0.59	0.19
	Average number of collisions	0.68	0.59	0.47	0.41	0.21

4.3. Algorithm Analysis

4.3.1. Algorithm Parameter

To ensure the best performance of the algorithm by co-ordinating various parameters, an analysis of the relevant parameters was conducted. In this section, the focus is on analyzing the maximum number of iterations (N_{iter}) and the number of wolves in the pack (N_{pop}). The experimental setup was identical to Experiment I in Section 4.2, conducted 20 times, and the results were averaged.

The test population size N_{pop} is set to 50, 100, 150, 200, 250, 300, and the number of iterations is set to 30, 60, 90, 120. The average fitness size of the NI-GWO algorithm is shown in Figure 11. It is evident that once the algorithm's population N_{pop} is below 200 or exceeds 300, the algorithm tends to converge too quickly, which may lead the population to fall into local optima. Wolf packs with a N_{pop} value between 200 and 300 have the ability to escape local optima, with the best effect occurring when $N_{iter} = 60$.

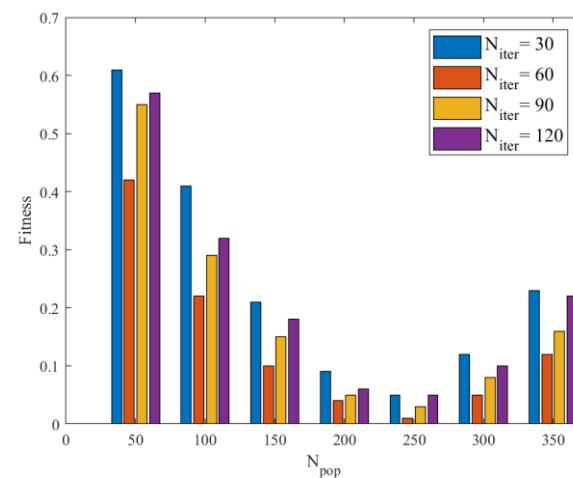


Figure 11. Average fitness value.

Based on this, the parameters in this paper are set as: $N_{pop} = 245$, $N_{iter} = 60$.

4.3.2. UAV Formation Generalization

To verify the path planning capability of the proposed algorithm when the UAV formation changes, simulations were conducted for v-shaped, echelon, and diamond formations of seven UAVs, based on the UAV formations discussed in Section 2.1.3. The number of iterations was kept constant at 245. The effectiveness of the collaborative path planning for the formations was recorded and analyzed to verify the algorithm's effectiveness in different UAV formation scenarios.

Figure 12 displays the path planning results of the NI-GWO algorithm under different UAV formation scenarios. It can be observed that, whether in v-shaped, echelon, or diamond formations, the NI-GWO algorithm is capable of quickly generating reasonable

flight paths while satisfying the prerequisites for obstacle avoidance. This proves that the proposed algorithm can accomplish reasonable path planning when facing different UAV formation configurations, demonstrating its generalization capability.

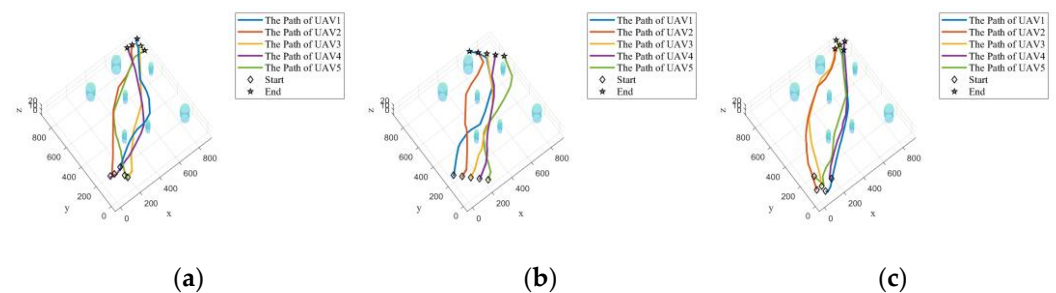


Figure 12. UAV path planning results: (a) v-shaped formation; (b) echelon formation; (c) diamond-shaped formation.

4.3.3. Dynamic Obstacle Avoidance Capability

In an environment with seven no-fly zones, dynamic obstacles (as indicated by the pink obstacles) were added. The obstacle avoidance speed suitable for UAVs can be formed relative to the speed of the dynamic obstacles, ensuring that while attempting to approach the original path, obstacles are avoided. Additionally, it was verified that in environments with three and five UAVs, the proposed algorithm in this paper can demonstrate dynamic obstacle avoidance capabilities after the introduction of DWA (Dynamic Window Approach).

Figure 13 displays the results of dynamic obstacle avoidance in UAV path planning, where the dashed lines represent the original planning of the NI-GWO algorithm, and the solid lines represent the flight paths of the UAVs after local re-planning with the incorporation of dynamic windows.

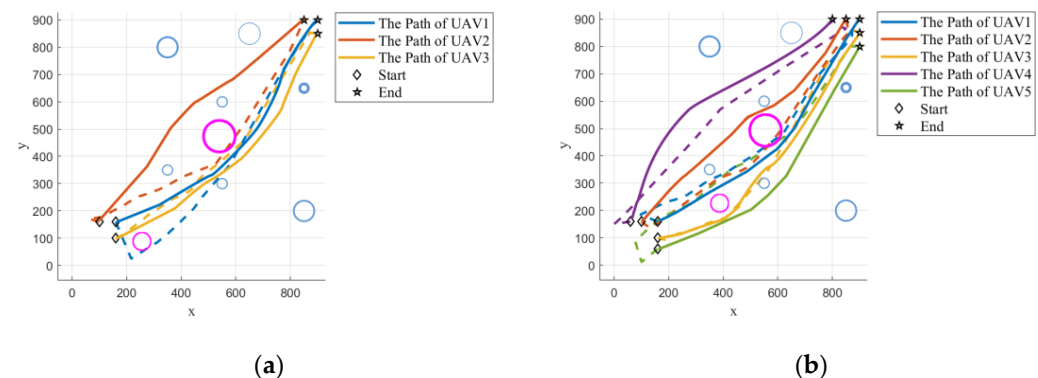


Figure 13. UAV path planning dynamic obstacle avoidance results: (a) three UAVs; (b) five UAVs.

4.4. Visualization of Planned Paths

Academic translation: This section primarily showcases the results of path planning under different algorithms to verify the superiority of the proposed algorithm in path planning issues. Five UAVs were set to use a v-shaped formation to plan the optimal path from the starting point to the destination in an environment with 11 no-fly zones. The three-dimensional visualizations of the path planning outcomes for different algorithms are depicted in Figure 14. It can be observed that the proposed algorithm in this paper can efficiently complete the flight mission under the collision constraints between obstacles and UAVs compared to other algorithms. In summary, the NI-GWO algorithm meets all the constraints required for all UAVs to safely, quickly, and collision-free complete their tasks, enabling it to execute flight path planning in complex environments with the presence of UAVs.

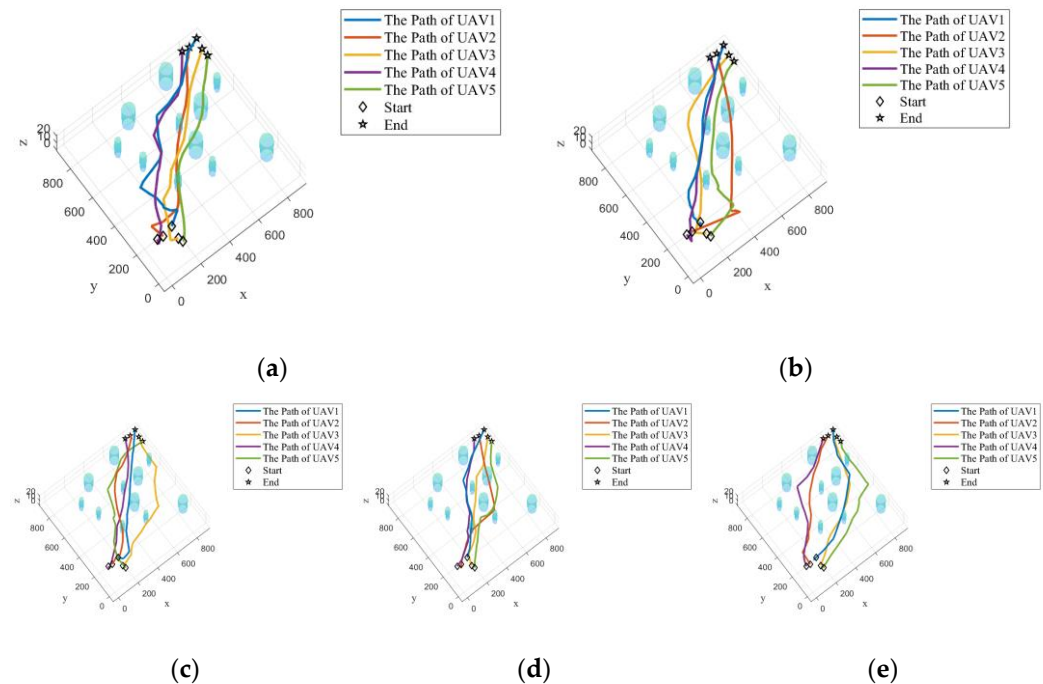


Figure 14. UAV paths planned by different algorithms: (a) path planning trajectory of the ABC algorithm; (b) path planning trajectory of the PSO algorithm; (c) path planning trajectory of the GWO algorithm; (d) path planning trajectory of the MP-GWO algorithm; (e) path planning trajectory of the NI-GWO algorithm.

5. Discussion

This study has provided innovative insights for further improving algorithm performance through an in-depth discussion of the path optimization method of the NI-GWO algorithm. However, there is still room for improvement in the proposed algorithm: (a) The current method mainly focuses on theoretical models and simulation validation, lacking support from real-world application cases or test data. To more closely simulate real flight conditions and real-world flight environments, future work should consider testing on actual UAV platforms or referencing existing real-world application cases to enhance the practical significance of the research. (b) This paper only discussed the usability of the algorithm in different UAV formations but is unable to achieve formation transitions in path planning. This requires in-depth research into dynamic UAV formation control technology, enabling UAVs to quickly and accurately adjust their formations and flight strategies according to mission requirements. Through research in this area, we can significantly improve the adaptability and combat capability of UAV formations in complex and variable environments. In summary, although this study has made certain progress in optimizing the GWO algorithm, further exploration is needed in terms of real environmental scenario data support and modeling of UAVs formation control. These research directions are crucial for promoting the comprehensive development of UAV technology, expanding its application prospects, and improving overall efficiency, and will lay a solid theoretical and technical foundation for future applications in the UAV field.

6. Conclusions

This paper investigates the NI-GWO algorithm based on a co-operative UAV path planning model. To enhance the performance of the GWO algorithm, we conducted research and improvements, perfecting a comprehensive trajectory cost function, convergence factors, and position update methods. Additionally, to achieve dynamic obstacle avoidance, we integrated the DWA algorithm into our approach. We performed a series of path planning simulations for UAVs. Due to the singularity of the fitness function settings, a more random approach to selecting convergence factors, and static position update meth-

ods, traditional GWO and MP-GWO algorithms have insufficient capabilities in avoiding no-fly zones. In contrast, the NI-GWO algorithm proposed in this paper constructs an integrated trajectory cost function, employs a nonlinear convergence factor mechanism, and dynamically updates the position weights of Alpha, Beta, and Delta wolves, significantly improving the performance of UAVs flight path planning, and incorporating DWA to achieve dynamic obstacle avoidance for UAVs. Therefore, the algorithm is capable of adapting to various combat scenarios. Ultimately, simulations verified that the strategy can generate the best UAV path planning scheme in complex and uncertain environments. In military operations, the robustness and efficiency of UAV path planning are key factors in improving mission execution speed, reducing resource consumption, and ensuring the safety of the entire journey. The NI-GWO algorithm designs efficient flight paths for UAVs, capable of avoiding obstacles. This not only accelerates the UAV's arrival at the battlefield, shortens the path length, effectively reduces operational costs, but also decreases the number of no-fly zone crossings and UAV collision incidents, thereby significantly enhancing the success rate of combat missions.

Author Contributions: Literature review, X.Z. and J.Z.; writing, X.Z. and J.Z.; editing, X.Z. and G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Aeronautical Science Foundation of China (20220013053005), the 2024 Northwestern Polytechnical University Graduate Student Innovation Fund Project: 06080-24GH01020101, the Natural Science Basic Research Program of Shaanxi: 2022JQ-593, and the Key R&D Program of the Shaanxi Provincial Department of Science and Technology: 2022GY-089.

Data Availability Statement: The original contributions presented in the study are included in the article/Appendix A; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Academic translation: The appendix provides a comprehensive set of tables detailing the simulation environment. Table A1 outlines the path planning model simulation environment. Table A2 provides the parameter settings for the comparison algorithms.

Table A1. Path planning model simulation environment table.

Parameter Name	Parameter Size
Speed limit	(0.3 Ma~0.7 Ma)
The angle of divergence constraint	(-60° ~ 60°)
Inclination constraints	(-45° ~ 45°)
Position x constraints	(0~1000 cubic cells)
Position y constraints	(0~1000 cubic cells)
Safety distance	25 cubic cells
Weights (w_1, w_2, w_3, w_4)	Weights (0.05, 0.15, 0.70, 0.10)

Table A2. Parameter settings for the comparison algorithms.

Algorithm	Parameter Size
ABC	$N_{iter} = 100, N_{pop} = 300, FoodNumber = 150$
PSO	$N_{iter} = 100, N_{pop} = 300, w = 0.8, c1 = 1.45, c2 = 1.5$
GWO	$N_{iter} = 60, N_{pop} = 245$
MP-GWO	$N_{iter} = 60, N_{pop} = 245$
NI-GWO	$N_{iter} = 60, N_{pop} = 245$

References

1. Telli, K.; Kraa, O.; Himeur, Y.; Ouamane, A.; Boumehraz, M.; Atalla, S.; Mansoor, W. A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems* **2023**, *11*, 400. [\[CrossRef\]](#)
2. Rezaldi, M.Y.; Yoganingrum, A.; Prasetyadi, A.; Sutyawan, A.G.; Febriandirza, A.; Trianggoro, C.; Suhud, R. UAV-Assisted Heavy Metal Tracking in Oil Palm Plantations: Present Applications and Future Prospects. *Remote Sens. Lett.* **2024**, *15*, 55–65. [\[CrossRef\]](#)
3. Nguyen, H.-T.; Ghazali, I. Trends of Unmanned Aerial Vehicles in Logistics Delivery. *Bincang Sains dan Teknologi* **2024**, *3*, 52–59. [\[CrossRef\]](#)
4. Zhou, X.; Zhang, J.; Liu, J.; Yang, Q.; Du, Z.; Wu, Y. Evaluation of Autonomous Capability of Ground Attack UAV Based on Hierarchical Analysis Method. In *Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022)*; Fu, W., Gu, M., Niu, Y., Eds.; Lecture Notes in Electrical Engineering; Springer Nature Singapore: Singapore, 2023; Volume 1010, pp. 1072–1081, ISBN 978-981-9904-78-5.
5. Xu, H.; Li, Y.; Lin, C.; Ning, Z.; Liu, C.; Mai, B.; Deng, X.; Fan, S.; Li, J. Unveiling Vertical Ozone Variation with UAV-Based Monitoring and Modeling: A New Challenge for City-Level Ozone Pollution Control in the Pearl River Delta Region. *Atmos. Environ.* **2024**, *322*, 120368. [\[CrossRef\]](#)
6. Wanner, D.; Hashim, H.A.; Srivastava, S.; Steinhauer, A. UAV Avionics Safety, Certification, Accidents, Redundancy, Integrity, and Reliability: A Comprehensive Review and Future Trends. *Drone Syst. Appl.* **2024**, *12*, 1–23. [\[CrossRef\]](#)
7. Sun, J.; Yuan, G.; Song, L.; Zhang, H. Unmanned Aerial Vehicles (UAVs) in Landslide Investigation and Monitoring: A Review. *Drones* **2024**, *8*, 30. [\[CrossRef\]](#)
8. Javed, S.; Hassan, A.; Ahmad, R.; Ahmed, W.; Ahmed, R.; Saadat, A.; Guizani, M. State-of-the-Art and Future Research Challenges in UAV Swarms. *IEEE Internet Things J.* **2024**, *11*, 19023–19045. [\[CrossRef\]](#)
9. Shi, Z.; Zhou, X.; Zhang, J.; Zhang, A.; Shi, G.; Yang, Q. UAV Trajectory Prediction Based on Flight State Recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *60*, 2629–2641.
10. Shi, Z.; Zhang, J.; Shi, G.; Ji, L.; Wang, D.; Wu, Y. Design of a UAV Trajectory Prediction System Based on Multi-Flight Modes. *Drones* **2024**, *8*, 255. [\[CrossRef\]](#)
11. Liu, Y.; Zhang, J.; Zhang, K.; Zhou, X. Research on Threat Assessment Method of Formation Cooperative Combat in a Complex Environment. In *Proceedings of the 2023 8th International Conference on Computational Intelligence and Applications (ICCIA)*; IEEE: Haikou, China, 2023; pp. 59–63.
12. Li, W.; Shi, F.; Zhang, W.; Liu, Y.; Zhao, Y.; Wang, Z. A Review of Manned/Unmanned Aerial Vehicle Cooperative Technology and Application in U.S. Military. *Int. J. Adv. Netw. Monit. Controls* **2024**, *9*, 100–107. [\[CrossRef\]](#)
13. Wang, X.; Wang, Y.; Su, X.; Wang, L.; Lu, C.; Peng, H.; Liu, J. Deep Reinforcement Learning-Based Air Combat Maneuver Decision-Making: Literature Review, Implementation Tutorial and Future Direction. *Artif. Intell. Rev.* **2024**, *57*, 1. [\[CrossRef\]](#)
14. Luo, J.; Tian, Y.; Wang, Z. Research on Unmanned Aerial Vehicle Path Planning. *Drones* **2024**, *8*, 51. [\[CrossRef\]](#)
15. Bashir, N.; Boudjit, S.; Dauphin, G.; Zeadally, S. An Obstacle Avoidance Approach for UAV Path Planning. *Simul. Model. Pract. Theory* **2023**, *129*, 102815. [\[CrossRef\]](#)
16. Mazaheri, H.; Goli, S.; Nourollah, A. A Survey of 3D Space Path-Planning Methods and Algorithms. *ACM Comput. Surv.* **2024**, *57*, 3673896. [\[CrossRef\]](#)
17. Pan, Z.; Zhang, C.; Xia, Y.; Xiong, H.; Shao, X. An Improved Artificial Potential Field Method for Path Planning and Formation Control of the Multi-UAV Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 1129–1133. [\[CrossRef\]](#)
18. Rutenbar, R.A. Simulated Annealing Algorithms: An Overview. *IEEE Circuits Devices Mag.* **1989**, *5*, 19–26. [\[CrossRef\]](#)
19. Li, X.; Tong, Y. Path Planning of a Mobile Robot Based on the Improved RRT Algorithm. *Appl. Sci.* **2023**, *14*, 25. [\[CrossRef\]](#)
20. Nasir, J.; Islam, F.; Ayaz, Y. Adaptive Rapidly-Exploring-Random-Tree-Star (RRT*)-Smart: Algorithm Characteristics and Behavior Analysis in Complex Environments. *Asia-Pac. J. Inf. Technol. Multimed.* **2013**, *02*, 39–51. [\[CrossRef\]](#)
21. Zhang, C.; Zhen, Z.; Wang, D.; Li, M. UAV Path Planning Method Based on Ant Colony Optimization. In *Proceedings of the 2010 Chinese Control and Decision Conference*; IEEE: Xuzhou, China, 2010; pp. 3790–3792.
22. Karaboga, D.; Akay, B. A Comparative Study of Artificial Bee Colony Algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [\[CrossRef\]](#)
23. Yang, Y.; Luo, X.; Li, W.; Liu, C.; Ye, Q.; Liang, P. AAPF*: A Safer Autonomous Vehicle Path Planning Algorithm Based on the Improved A* Algorithm and APF Algorithm. *Clust. Comput.* **2024**, *27*, 11393–11406. [\[CrossRef\]](#)
24. Deng, M.; Yang, Q.; Peng, Y. A Real-Time Path Planning Method for Urban Low-Altitude Logistics UAVs. *Sensors* **2023**, *23*, 7472. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Agrawal, S.; Patle, B.K.; Sanap, S. A Systematic Review on Metaheuristic Approaches for Autonomous Path Planning of Unmanned Aerial Vehicles. *Drone Syst. Appl.* **2024**, *12*, 1–28. [\[CrossRef\]](#)
26. Noreen, I.; Khan, A.; Habib, Z. Optimal Path Planning Using RRT* Based Approaches: A Survey and Future Directions. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *11*. [\[CrossRef\]](#)
27. Aslan, M.F.; Durdu, A.; Sabanci, K. Goal Distance-Based UAV Path Planning Approach, Path Optimization and Learning-Based Path Estimation: GDRRT*, PSO-GDRRT* and BiLSTM-PSO-GDRRT*. *Appl. Soft Comput.* **2023**, *137*, 110156. [\[CrossRef\]](#)
28. Yildiz, B.; Aslan, M.F.; Durdu, A.; Kayabasi, A. Consensus-Based Virtual Leader Tracking Swarm Algorithm with GDRRT*-PSO for Path-Planning of Multiple-UAVs. *Swarm Evol. Comput.* **2024**, *88*, 101612. [\[CrossRef\]](#)

29. Yang, F.; Wang, P.; Zhang, Y.; Zheng, L.; Lu, J. Survey of Swarm Intelligence Optimization Algorithms. In *Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS)*; IEEE: Beijing, China, 2017; pp. 544–549.
30. Zhang, J.; Zhu, X.; Li, J. Intelligent Path Planning with an Improved Sparrow Search Algorithm for Workshop UAV Inspection. *Sensors* **2024**, *24*, 1104. [[CrossRef](#)]
31. Prity, F.S.; Uddin, K.M.A.; Nath, N. Exploring Swarm Intelligence Optimization Techniques for Task Scheduling in Cloud Computing: Algorithms, Performance Analysis, and Future Prospects. *Iran J. Comput. Sci.* **2024**, *7*, 337–358. [[CrossRef](#)]
32. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
33. Global Path Planning for Multiple AUVs Using GWO. *Arch. Control Sci.* **2023**, *30*, 77–100. [[CrossRef](#)]
34. Bansal, J.C.; Singh, P.K.; Pal, N.R. (Eds.) *Evolutionary and Swarm Intelligence Algorithms*; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2019; Volume 779, ISBN 978-3-319-91339-1.
35. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary Grey Wolf Optimization Approaches for Feature Selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
36. Radmanesh, M.; Kumar, M.; Sarim, M. Grey Wolf Optimization Based Sense and Avoid Algorithm in a Bayesian Framework for Multiple UAV Path Planning in an Uncertain Environment. *Aerosp. Sci. Technol.* **2018**, *77*, 168–179. [[CrossRef](#)]
37. Zhao, D.; Cai, G.; Wang, Y.; Li, X. Path Planning of Obstacle-Crossing Robot Based on Golden Sine Grey Wolf Optimizer. *Appl. Sci.* **2024**, *14*, 1129. [[CrossRef](#)]
38. Yu, X.; Jiang, N.; Wang, X.; Li, M. A Hybrid Algorithm Based on Grey Wolf Optimizer and Differential Evolution for UAV Path Planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [[CrossRef](#)]
39. Rao, C.; Wang, Z.; Shao, P. A Multi-Strategy Collaborative Grey Wolf Optimization Algorithm for UAV Path Planning. *Electronics* **2024**, *13*, 2532. [[CrossRef](#)]
40. Chen, Y.; Yu, Q.; Han, D.; Jiang, H. UAV Path Planning: Integration of Grey Wolf Algorithm and Artificial Potential Field. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8120. [[CrossRef](#)]
41. Sun, Y.; Lv, B.; Yang, H.; Li, X. Multi-UAV Trajectory Planning Based on Improved Multi-Population Grey Wolf Optimizer Algorithm. In *Proceedings of the 2024 36th Chinese Control and Decision Conference (CCDC)*; IEEE: Xi'an, China, 2024; pp. 6142–6148.
42. Ogren, P.; Leonard, N.E. A Convergent Dynamic Window Approach to Obstacle Avoidance. *IEEE Trans. Robot.* **2005**, *21*, 188–195. [[CrossRef](#)]
43. Bes, J.; Dendarieta, J.; Riazuelo, L.; Montano, L. DWA-3D: A Reactive Planner for Robust and Efficient Autonomous UAV Navigation. *arXiv* **2024**, arXiv:2409.05421.
44. Wang, W. Path Planning of UAV Crossing Dense Obstacle Area Based on Improved Dynamic Window Approach. In *Proceedings of the 2023 5th International Conference on Electronic Engineering and Informatics (EEI)*, Wuhan, China, 23–25 June 2023.
45. Li, P.; Hao, L.; Zhao, Y.; Lu, J. Robot Obstacle Avoidance Optimization by A* and DWA Fusion Algorithm. *PLoS ONE* **2024**, *19*, e0302026. [[CrossRef](#)]
46. Gong, J.; Chang, T.-H.; Shen, C.; Chen, X. Flight Time Minimization of UAV for Data Collection over Wireless Sensor Networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1942–1954. [[CrossRef](#)]
47. Alkouz, B.; Bouguettaya, A. Formation-Based Selection of Drone Swarm Services. In *Proceedings of the MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Darmstadt, Germany, 7–9 December 2020.
48. Phung, M.D.; Ha, Q.P. Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [[CrossRef](#)]
49. Han, D.; Yu, Q.; Jiang, H.; Chen, Y.; Zhu, X.; Wang, L. Three-Dimensional Path Planning for Post-Disaster Rescue UAV by Integrating Improved Grey Wolf Optimizer and Artificial Potential Field Method. *Appl. Sci.* **2024**, *14*, 4461. [[CrossRef](#)]
50. Feng, J.; Sun, C.; Zhang, J.; Du, Y.; Liu, Z.; Ding, Y. A UAV Path Planning Method in Three-Dimensional Space Based on a Hybrid Gray Wolf Optimization Algorithm. *Electronics* **2023**, *13*, 68. [[CrossRef](#)]
51. Huang, H.; Jiang, Z.; Yan, T.; Bai, Y. Dynamic Task Allocation for Heterogeneous Multi-UAVs in Uncertain Environments Based on 4DI-GWO Algorithm. *Drones* **2024**, *8*, 236. [[CrossRef](#)]
52. Xue, Y.; Jiang, J.; Zhao, B.; Ma, T. A Self-Adaptive Artificial Bee Colony Algorithm Based on Global Best for Global Optimization. *Soft Comput.* **2018**, *22*, 2935–2952. [[CrossRef](#)]
53. Meng, Q.; Qu, Q.; Chen, K.; Yi, T. Multi-UAV Path Planning Based on Cooperative Co-Evolutionary Algorithms with Adaptive Decision Variable Selection. *Drones* **2024**, *8*, 435. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.