COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 04/06 pp 274−291
Volume 4, Number 3, September 2024
DOI: 10.23919/CSMS.2024.0013

Multi-UAV Collaborative Trajectory Planning for 3D Terrain Based on CS-GJO Algorithm

Taishan Lou, Yu Wang*, Zhepeng Yue, and Liangyu Zhao

Abstract: Existing solutions for collaborative trajectory planning using multiple UAVs suffer from issues such as low accuracy, instability, and slow convergence. To address the aforementioned issues, this paper introduces a new method for multiple unmanned aerial vehicle (UAV) 3D terrain cooperative trajectory planning based on the cuckoo search golden jackal optimization (CS-GJO) algorithm. A model for single UAV trajectory planning and a model for multi-UAV collaborative trajectory planning have been developed, and the problem of solving the models is restructured into an optimization problem. Building upon the original golden jackal optimization, the use of tent chaotic mapping aids in the generation of the golden jackal's initial population, thereby promoting population diversity. Subsequently, the position update strategy of the cuckoo search algorithm is combined for purpose of update the position information of individual golden jackals, effectively preventing the algorithm from getting stuck in local minima. Finally, the corresponding nonlinear control parameter were developed. The new parameters expedite the decrease in the convergence factor during the pre-exploration stage, resulting in an improved overall search speed of the algorithm. Moreover, they attenuate the decrease in the convergence factor during the post-exploration stage, thereby enhancing the algorithm's global search. The experimental results demonstrate that the CS-GJO algorithm efficiently and accurately accomplishes multi-UAV cooperative trajectory planning in a 3D environment. Compared with other comparative algorithms, the CS-GJO algorithm also has better stability, higher optimization accuracy, and faster convergence speed.

Key words: golden jackal optimization; multiple unmanned aerial vehicle (multi-UAV) collaboration; 3D track planning; tent chaos mapping; cuckoo search

1 Introduction

With the continuous development of science and technology, unmanned aerial vehicles (UAVs) have gained widespread utilization due to their advantages of low cost and high efficiency. Their applications span various domains, encompassing civilian areas such as agricultural protection, surveillance, law enforcement, as well as military operations like battlefield reconnaissance and bombing missions[1]. Trajectory planning plays a key role in UAVs mission execution and significantly contributes to enhancing the survivability and maneuverability of UAVs[2].

Classical planning methods for UAV trajectory planning are typically divided into two categories: signpost graphic based and raster-based planning methods[3]. Signpost graphic approaches include techniques such as Voronoi diagram, probabilistic roadmaps[4], and rapidly-exploring random trees[5]. Raster approaches include algorithms like the A* algorithm[6], artificial potential field[7], and intelligent

• Taishan Lou, Yu Wang, and Zhepeng Yue are with the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China. E-mail: tayzan@sina.com; wyu0531@163.com; Soper10050@163.com.

• Liangyu Zhao is with the School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China. E-mail: zhaoly@bit.edu.cn.

∗ To whom correspondence should be addressed.

※ This article was recommended by Associate Editor Lining Xing
Manuscript received: 2024-03-14; revised: 2024-05-31; accepted: 2024-06-04

optimization methods. However, signpost graphic approaches and A* algorithm suffer from computational complexity and limited real-time performance when solving 3D UAV path planning problems. In contrast, intelligent optimization algorithms are commonly used to address high-dimensional planning problems due to their advantages of simple parameter configuration, stability, and fast convergence speed.

In the military domain, the diversification of battlefield missions have resulted in decreased efficiency when using a single UAV to carry out combat missions. Consequently, the cooperation of multiple UAVs has become the focal point of future research[8]. At the moment, major military powers are intensifying their efforts in researching multi-UAV cooperative technology and conducting cluster flight tests, aiming to maintain their leadership in UAV cluster technology during actual combat scenarios[9]. To ensure that multiple UAVs in coordinated formations are able to reach the target area safely and quickly under complex conditions while reducing the risk of being detected and destroyed by enemy military weapons, it is essential to plan an optimal formation trajectory that satisfies battlefield constraints and performance criteria[10]. In this context, Zhou et al.[11] proposed the multiple population grey wolf optimizer algorithm that combines ideas from multiple population optimization techniques with the grey wolf optimizer (GWO) algorithm. This approach addresses the instability issue when planning cooperative trajectory for multi-UAV and realizes cooperative trajectory planning for unmanned combat air vehicle formations in the two-dimensional plane. Zhou et al.[12] introduced the cooperated non-dominated sorting genetic algorithms-II for multi-UAV collaborative trajectory planning. This algorithm treats the trajectory planning of each UAV as a subgroup to be optimized independently, ensuring that UAV formations are coordinated in time and space. Additionally, the algorithm generates multiple alternative solutions to meet different task requirements. However, the existing literature mentioned above solely focuses on investigating the problem of multi-UAV collaborative trajectory planning in two-dimensional space. Unfortunately, it overlooks the crucial factor of the high maneuverability of UAV formations.

In a separate study, Swartzentruber et al.[13] utilized particle swarm optimization (PSO) to produce various 3D paths based on predefined criteria within the simulation test scenario, aiming to assess the efficiency of the path planner. However, the construction of such a complex and costly simulation test scenario impedes extensive research into collaborative multi-UAV trajectory planning. Shan et al.[14] conducted simulation experiments on a 64-bit Windows 10 experimental platform equipped with MATLAB. They proposed the particle swarm optimization and Hook-Jeeves (PSO-HJ) search algorithm, which combines PSO and Hook-Jeeves techniques. The PSO-HJ algorithm is designed to enhance population diversity and accelerate planning speed. It aims to address the issues of computational complexity and low convergence accuracy in collaborative trajectory planning for multi-UAVs in three-dimensional space. Yang et al.[15] proposed a trajectory planning method that utilizes multiple swarm chaotic grey wolf optimization. This method employs the concept of multiple swarms to construct an initial trajectory set and utilizes the chaotic search strategy to overcome the issue of GWO getting trapped in local minima. Furthermore, Zhang et al.[16] designed three particle-updating modes, namely exploring, developing, and hybrid updating, to balance population diversity in the PSO algorithm.

Golden jackal optimization (GJO) is an innovative intelligent optimization algorithm proposed in 2022[17]. It is inspired by the cooperative behavior of male and female golden jackals in nature, specifically their strategy of searching for prey and encircling it. The GJO algorithm is highly regarded for its advantages in terms of convergence speed, ability to find optimal solutions, and robustness. This method is widely used. Houssein et al.[18] utilized an improved GJO algorithm for image segmentation in skin cancer imaging. Rezaie et al.[19] developed a modified GJO algorithm for parameter estimation of the proton exchange membrane fuel cell model. Hanafi et al.[20] used an enhanced binary GJO algorithm to detect intrusions in the Internet of Things. Furthermore, Lou et al.[21] proposed a hybrid strategy based on the GJO algorithm and applied it to mobile robot path planning in a two-dimensional plane. However, no proposal has been made for applying the improved GJO algorithm to the problem of multi-UAV trajectory planning.

Hence, this paper proposes the cuckoo search golden jackal optimization (CS-GJO) algorithm. Firstly, the population's diversity is enriched through the utilization of tent chaos mapping for population initialization[22]. Secondly, the global search of the cuckoo search (CS)[23] is integrated into the GJO to update the optimal solutions for golden jackals. Next, the nonlinear parameters corresponding to the improvement strategy are meticulously designed. The new parameters expedite the decrease in the convergence factor during the pre-exploration stage, resulting in an improved overall search speed of the algorithm. Moreover, the new parameters attenuate the decrease in the convergence factor during the post-exploration stage, effectively preventing the algorithm from falling into local minima. To evaluate the performance of the proposed CS-GJO algorithm, comparative experiments are conducted by using PSO[24], GWO[25], GJO, and CS-GJO algorithms in the single UAV trajectory planning model. The experimental results demonstrate that the CS-GJO algorithm not only is cost-effective in terms of planning, but also exhibits greater stability and a lower likelihood of getting stuck in local minima compared to the other algorithms being compared. Furthermore, the CS-GJO is applied to the cooperative trajectory planning model for multi-UAV and is then compared and analyzed with the GWO, PSO, and GJO. The experiments demonstrate that this algorithm possesses several advantages compared to the other three algorithms. These advantages include strong robustness, high optimization accuracy, and fast convergence speed. The CS-GJO algorithm provides a new scheme for generating flyable multi-UAV 3D cooperative trajectories.

The remaining work is as follows. Section 2 carries out the modelling of a multi-UAV cooperative trajectory, Section 3 describes the improved CS-GJO algorithm, Section 4 carries out the corresponding simulation experiments, and Section 5 concludes the whole paper.

2 UAV 3D Collaborative Trajectory Planning

In this section, the complex multi-UAV trajectory planning problem is decomposed into two layers by using the idea of hierarchical modeling. Firstly, the trajectory planning of a single UAV is carried out. Secondly, the objective function for the trajectory of multiple UAVs is obtained by incorporating the space and time constraints, while also ensuring that the planning of a single aircraft is satisfied.

2.1 Single UAV trajectory planning modelling

Toward improve practical applications, it is necessary to address two key issues when establishing a single UAV trajectory planning model. One is designing a suitable cost function, and another is to set constraints.

2.1.1 Fuel cost

Since the energy consumption during UAV flight is proportional to the length of the flight trajectory, a longer trajectory results in greater energy consumption[26]. Therefore, the fuel cost f_{fuel} for UAV can be expressed in terms of the trajectory length as follows:

$$f_{\text{fuel}} = \sum_{m=1}^{M-1} L_m \tag{1}$$

where L_m is the track length of the m-th track segment, and M is the number of waypoints.

2.1.2 Highly variable cost

UAV in flight need to repeatedly perform ascending and descending maneuvers to avoid accidentally entering a restricted area or colliding with mountains. However, this repeated lifting and lowering can affect the feasibility of UAV flight[27]. Therefore, to reflect the feasibility of flight, the cost of the UAV's altitude change f_{h} can be mathematically formulated as follows:

$$f_{\text{h}} = \frac{\sum\limits_{m=1}^{M} \left(z_m - \frac{1}{M} \sum\limits_{m=1}^{M} z_m \right)^2}{M} \tag{2}$$

where z_m is the height of the m-th waypoint.

2.1.3 Smoothing cost

If the deflection angle of the UAV is too large, the risk of the aircraft losing control increases. Therefore, this paper introduces a cost to reflect the stability and smoothness of the UAV trajectory[26]. That is, the UAV smoothing cost can be defined as follows:

$$f_{\text{smooth}} = \sum_{m=2}^{M} |\delta_m - \delta_{m-1}| \tag{3}$$

where δ_m is the deflection angle of the m-th waypoint.

2.1.4 Combined threat cost

When undertaking combat missions, the UAV faces threats from enemy air defense facilities and adverse weather conditions. To adapt to mission requirements and ensure survival, radar threats are equated with hemispheres, while other threats are equated with cylinders. In modern warfare, the threat level of detection radars is much lower than that of air defense weapons[28], the cost of this combined threat can be represented as follows:

$$f_{\text{threat}} = \sum_{m=1}^{M-1} \sum_{j=1}^{G} P_{\text{R}}^{j}(m) + \sum_{m=1}^{M-1} \sum_{j=1}^{U} P_{\text{M}}^{j}(m) \tag{4}$$

where G is the number of radars, U is the number of other threats. $P_{\mathrm{R}}^{j}(m)$ and $P_{\mathrm{M}}^{j}(m)$ represent the radar threat and other threats to the UAV in the m-th trajectory segment, respectively:

$$P_{\mathrm{R}}^{j}(m)=\begin{cases}0, & d_{\mathrm{R}}^{j}>d_{\mathrm{R\,max}}^{j};\\ \dfrac{\lambda^{2}}{\left(d_{\mathrm{R}}^{j}\right)^{4}}, & d_{\mathrm{R}}^{j}\leqslant d_{\mathrm{R\,max}}^{j}\end{cases}\tag{5}$$

$$P_{\mathrm{M}}^{j}(m)=\begin{cases}0, & d_{\mathrm{M}}^{j}>d_{\mathrm{M\,max}}^{j};\\ \dfrac{\lambda^{2}}{d_{\mathrm{M}}^{j}}, & d_{\mathrm{M}}^{j}\leqslant d_{\mathrm{M\,max}}^{j}\end{cases}\tag{6}$$

where d_{R}^{j} and d_{M}^{j} denote the distances between the UAV and the radar and other threats, respectively. Additionally, $d_{\mathrm{R\,max}}^{j}$ and $d_{\mathrm{M\,max}}^{j}$ represent the maximum radius of the radar's detection area and the maximum kill radius of the other threats, respectively. λ is the threat coefficient, which is taken as $\lambda=50$ in the next experiments.

In order to compare the size relationship between d_{R}^{j} and $d_{\mathrm{R\,max}}^{j}$, d_{M}^{j}, and $d_{\mathrm{M\,max}}^{j}$, the following methods are used to determine whether the flight trajectory intersects with the threat area.

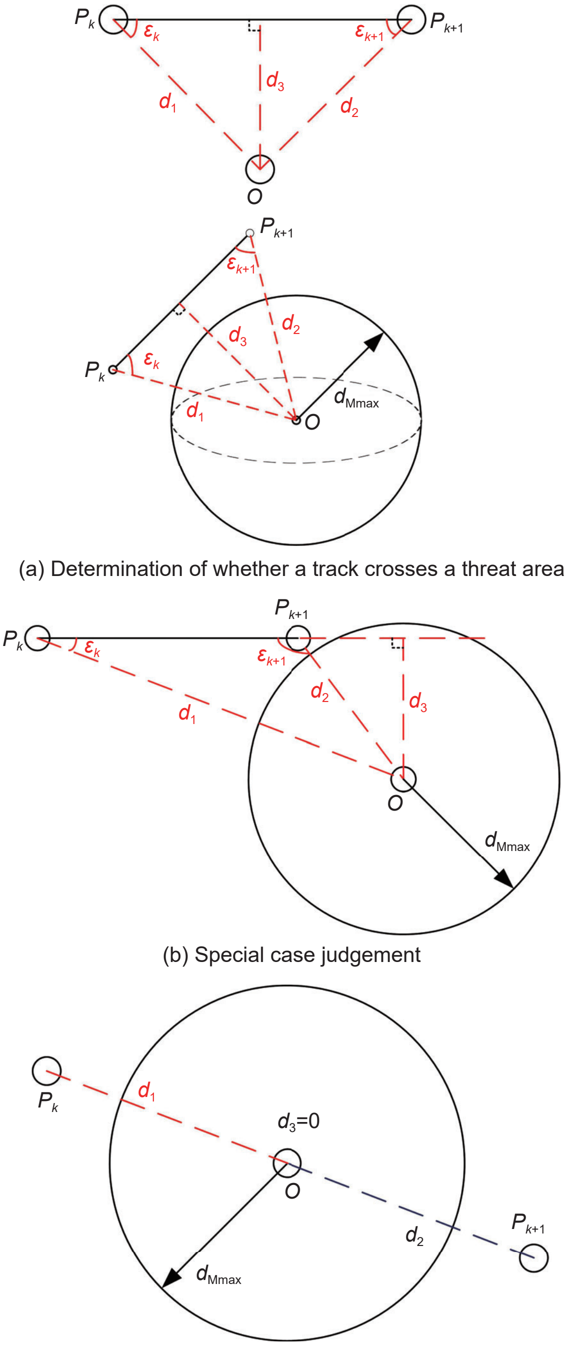
Step 1: Set the distance from the center of the threat area O to the two adjacent waypoints P_k and P_{k+1} to be d_1 and d_2, and the distance to the line connecting P_k and P_{k+1} to be d_3, $\angle OP_kP_{k+1}$ and $\angle OP_{k+1}P_k$ are denoted as ε_k, ε_{k+1}. The schematic is shown in Fig. 1a, where for cylindrical threats, the threat area, and trajectory need to be projected onto a 2D horizontal plane.

Step 2: If $d_1<d_{\mathrm{M\,max}}\,\&\,d_2<d_{\mathrm{M\,max}}$, the trajectory crosses the threat zone; otherwise, go to the next step.

Step 3: If $d_3\geqslant d_{\mathrm{M\,max}}$, the trajectory does not cross the threat zone; otherwise, go to the next step.

Step 4: If $d_3<d_{\mathrm{M\,max}}$, judge whether are acute angles. If both are acute angles, the trajectory traverses the threat zone; otherwise, the trajectory does not traverse the threat zone, as shown in Fig. 1b.

Step 5: If $d_3=0$, determine whether the length of P_kP_{k+1} is greater than OP_k and OP_{k+1} at the same time, if it is greater at the same time, the trajectory crosses the threat zone, as shown in Fig. 1c; otherwise, the trajectory does not cross the threat zone.



(a) Determination of whether a track crosses a threat area

(b) Special case judgement

(c) Co-linear judgement

Fig. 1 Threat area crossing judgement.

2.1.5 Flight level constraint

UAV encounters complex terrain environments during flight. The lower the UAV flight altitude, the higher the risk of collision between the UAV and the surrounding terrain. On the contrary, if the flying height is too high, the UAV will become more susceptible to radar detection by the enemy[29]. Therefore, this paper needs to establish both the minimum and maximum flight altitudes. The i-th UAV flight altitude constraint is formulated as

$$A_1\colon H_{\min}\leqslant z_i\leqslant H_{\max}\tag{7}$$

2.1.6 Flight range constraint

The farthest flight distance that can be supported by the UAV's own stored energy (fuel or battery) is denoted by $L_{\max}$. To ensure that the UAV can successfully reach its target point, the farthest range constraint that the trajectory length $L_{\rm i}$ of the i-th UAV should satisfy

$$A_2\text{: } L_i \leqslant L_{\max} \tag{8}$$

To ensure a smooth transition in the UAV's attitude, a minimum trajectory segment distance $L_{\min}$ must be set. The minimum flight path constraint for any trajectory segment L_m in the trajectory can be described as follows:

$$A_3 : L_m \geqslant L_{\min} \tag{9}$$

2.1.7 Maximum climb and turn angle constraints

Due to the limitations of the airframe material and structural design, there are maximum climb angle and maximum turn angle constraints for the UAV. The expressions for the that constraints on the turning angle φ_m and the climbing angle θ_m between adjacent trajectory segments of the UAV are given below:

$$A_4\text{: } |\varphi_m| - \varphi_{\max} \leqslant 0, \quad |\theta_m| - \theta_{\max} \leqslant 0 \tag{10}$$

where $\varphi_{\max}$ is the maximum turning angle and $\theta_{\max}$ is the maximum climb angle.

$$\varphi_m = \arccos\left(\frac{d'_{m-1}d'_m \times d'_m d'_{m+1}}{\left|d'_{m-1}d'_m\right| \cdot \left|d'_m d'_{m+1}\right|}\right) \tag{11}$$

$$\theta_m = \arccos\left(\frac{\left|d'_m d'_{m+1}\right|}{|d_m d_{m+1}|}\right) \tag{12}$$

where × stands for vector product, d_{m-1}, d_m, and d_{m+1} are the UAV's waypoints. d'_{m-1}, d'_m, and d'_{m+1} are the horizontal projection points of d_{m-1}, d_m, and d_{m+1}, respectively.

The relationship between deflection angle, climb angle, and turn angle is shown schematically in Fig. 2.

2.1.8 Velocity constraint

Flying a UAV too slowly will reduce its efficiency. On the contrary, if the UAV's speed is too fast, it is easy to lose control. The velocity constraint of the UAV can be developed as follows:

$$A_5 : v_{\min} < v_i < v_{\max} \tag{13}$$

where $v_{\min}$ is the minimum velocity, $v_{\max}$ is the maximum velocity, and v_i is the velocity of the i-th UAV.

2.1.9 Single UAV trajectory planning objective function

Combining the various cost functions and UAV constraints described above, the objective function for a single UAV trajectory planning can be shown as follows:

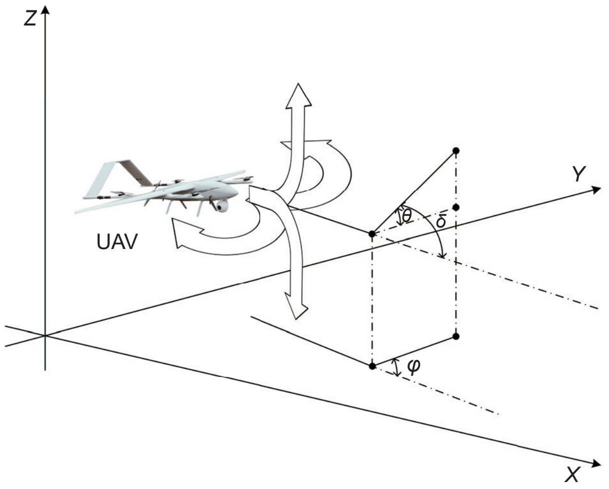


Fig. 2 Turning angle, climbing angle, and deflection angle relationship.

$$\begin{aligned}\min\ f = {}& \varepsilon_1 \cdot f_{\rm fuel} + \varepsilon_2 \cdot f_{\rm h} + \varepsilon_3 \cdot f_{\rm smooth} + \\ & \varepsilon_4 \cdot f_{\rm threat} + \eta\end{aligned} \tag{14}$$

where ε_i, $i = 1, 2, 3, 4$ is the weight corresponding to denoting different cost functions, $\sum\limits_{i=1}^{4} \varepsilon_i = 1$, and η is the penalty function. The penalty function is used to transform the UAV trajectory planning problem into an unconstrained function problem:

$$\eta = \begin{cases} 0, & \text{compliance with constraints;} \\ +\infty, & \text{non-compliance with constraints} \end{cases} \tag{15}$$

2.2 Multi-UAV collaborative trajectory planning modelling

To plan the collaborative trajectory for multiple UAVs, temporal and spatial constraints need to be set for each UAV, and then alternative trajectories that satisfy the requirements of single-aircraft planning are secondarily planned[26].

2.2.1 Time collaboration cost

In order for each UAV to arrive at the target position simultaneously or within a specified time frame, a time coordination constraint must be set. The collaborative arrival time (CAT) for the UAVs can be described as follows:

$$\text{CAT} = \bigcap_{i=1}^{N} T_i \tag{16}$$

$$T_i = T_{i,1} \bigcup T_{i,2} \bigcup \cdots \bigcup T_{i,h} \bigcup \cdots \bigcup T_{i,H} \tag{17}$$

where $T_{i,h}$=[$t^{i,h}_{\min}$,$t^{i,h}_{\max}$], $t^{i,h}_{\min} = \frac{L_{i,h}}{v^i_{\max}}$, $t^{i,h}_{\max} = \frac{L_{i,h}}{v^i_{\min}}$, $v^i_{\min}$ and $v^i_{\max}$ are the minimum and maximum velocities of the UAVs, respectively, $L_{i,h}$ is the length of the h-th alternative track of the i-th UAVs, N is the number of UAVs, and H is the number of alternative tracks per UAVs. Taking three UAVs as an example, the CAT generation mechanism is shown in Fig. 3.

The actual time spent by the UAV to arrive at the target location is t_i, and the minimum value of the CAT intersection is t_c, the flight time range of the i-th UAV is $T_i = \left[t^i_{\min}, t^i_{\max}\right]$. The time collaboration cost function is developed as

$$B_{\text{time}} = \sum_{i=1}^{N} a(i) \tag{18}$$

where

$$a(i) = \begin{cases} 0, & t^i_{\min} \leqslant t_c \leqslant t^i_{\max}; \\ |t_i - t_c|, & t^i_{\max} < t_c, t^i_{\min} > t_c \end{cases} \tag{19}$$

2.2.2 Spatial collaboration cost

The spatial collaboration constraint is designed to prevent collisions between UAVs by requiring that the flight distance between multiple UAVs is not less than a safe distance. This can be expressed mathematically as

$$d_{\text{s}} \leqslant d_{i-j}, \quad i \neq j \tag{20}$$

where d_{s} is safety distance, d_{i-j} is the minimum distance between the i-th UAV and the j-th UAV's waypoint.

The function that represents the spatial collaboration cost is as follows:

$$B_{\text{space}} = \sum_{i=1}^{N}\sum_{j=1}^{N} \max(d_{\text{s}} - d_{i-j}, 0) \tag{21}$$

2.2.3 Collaboration objective function

Toward ensuring that the UAV formation arrives at the target position simultaneously in the shortest time, a multi-UAV cooperative objective function is designed as

$$\min\ F = \sum_{i=1}^{N} f_i + t_c + \eta \cdot \left(B_{\text{time}} + B_{\text{space}}\right) \tag{22}$$

where $\sum_{i=1}^{N} f_i$ is the sum of costs of the trajectory planning objective function of each UAV in the UAV formation (obtained by Eq. (14)), and the number of UAVs is N; $t_c = \min(t^1_{\max}, t^2_{\max}, \ldots, t^i_{\max}) - \max(t^1_{\min}, t^2_{\min}, \ldots, t^i_{\min})$ is the minimum of the collaborative arrival time.

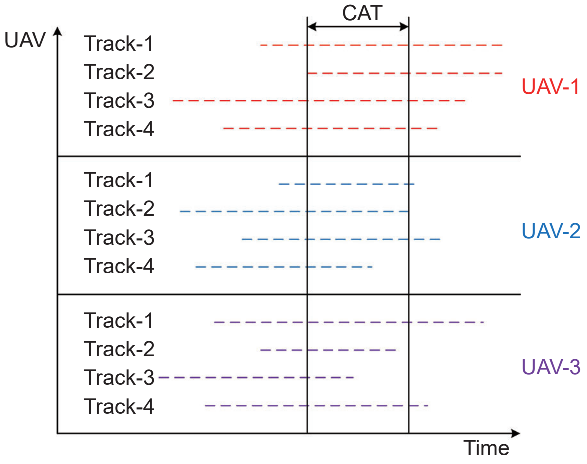


Fig. 3 CAT generation mechanism.

3 CS-GJO Algorithm

In order to plan better multi-UAV cooperative trajectories, this paper improves the GJO algorithm by using the CS algorithm and other related strategies.

3.1 Standard GJO algorithm

Golden jackals are highly social animals, known for its excellent cooperative hunting abilities. The golden jackals are approximately three times more successful in cooperative hunting than in hunting alone. The golden jackals often exhibit monogamy, with pairs hunting together in an amazingly synchronized manner. The GJO algorithm consists of two basic steps: exploration phase and exploitation phase.

During the hunting process, the female jackal typically follows the male. Consequently, the search for prey by a pair of golden jackals are given as follows:

$$Y_1(t) = Y_{\text{M}}(t) - E \cdot \left|Y_{\text{M}}(t) - \text{rl} \cdot \text{Prey}(t)\right| \tag{23}$$

$$Y_2(t) = Y_{\text{F}}(t) - E \cdot \left|Y_{\text{F}}(t) - \text{rl} \cdot \text{Prey}(t)\right| \tag{24}$$

where t is the number of iterations. $\text{Prey}(t)$ is the position of the prey. $Y_{\text{M}}(t)$ and $Y_{\text{F}}(t)$ are the positions of the optimal and suboptimal individuals in the population, respectively. $Y_1(t)$ and $Y_2(t)$ are the updated positions.

E is the captured animal's own energy, which can be described mathematically as follows:

$$E = e_1 \cdot e_0 \tag{25}$$

where e_1 is used to decrement the captured animal's own energy and e_0 is the initial energy of the captured animal.

$$e_0 = 2r - 1 \tag{26}$$

$$e_1 = c_1\left(1 - \frac{t}{T}\right) \tag{27}$$

where $r \in [0,1]$, T is the maximum number of iterations, and c_1 is 1.5.

rl is a random vector based on the Lévy distribution, which is given by

$$\mathrm{rl} = 0.05 \times 0.01 \mu\sigma / v^{(1/\beta)} \tag{28}$$

$$\sigma = \left[\frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\beta 2^{\frac{\beta-1}{2}} \Gamma\left(\frac{1+\beta}{2}\right)} \right]^{1/\beta} \tag{29}$$

where μ and v are random numbers in the range (0, 1), β is a constant set to 1.5, and Γ is the Gamma function.

In summary, the equation to update the position of the golden jackal can be represented as follows:

$$Y(t+1) = \frac{Y_1(t) + Y_2(t)}{2} \tag{30}$$

where $Y(t+1)$ is the updated position.

When golden jackals surround their prey, the prey's energy levels decrease. After the encirclement, the golden jackals begin to attack and feed on the prey. This hunting behavior can be mathematically described as follows:

$$Y_1(t) = Y_{\mathrm{M}}(t) - E \cdot \left| \mathrm{rl} \cdot Y_{\mathrm{M}}(t) - \text{ Prey}(t) \right| \tag{31}$$

$$Y_2(t) = Y_{\mathrm{FM}}(t) - E \cdot \left| \mathrm{rl} \cdot Y_{\mathrm{FM}}(t) - \text{ Prey}(t) \right| \tag{32}$$

Finally, the golden jackal position update is still calculated according to Eq. (30).

3.2 Improved CS-GJO algorithm

The GJO algorithm has several advantages, including simplicity of design, few computational formulas, and ease of implementation. However, the single structure of the GJO algorithm population leads to limited search capability and is currently mostly used for single-objective optimization problems. Therefore, to further improve the search ability of the algorithm in multidimensional space, the CS-GJO algorithm is proposed in this paper.

3.2.1 Tent chaotic map

GJO algorithms typically initialize their population with randomly generated data. However, this approach often leads to suboptimal results. In contrast, chaotic motion exhibits characteristics such as randomness, regularity, and ergodicity, which are conducive to maintaining population diversity and enhancing global search ability. Existing chaotic mappings, such as the tent mapping and the logistic mapping, differ in their ability to improve function optimization. The logistic mapping sequence is shown in Fig. 4. The tent mapping sequence is shown in Fig. 5. By comparing Figs. 4 and 5, it can be found that the logistic mapping has a higher concentration of values in the intervals $[0, 0.1]$ and $[0.9, 1]$. This uneven distribution hampers the algorithm's convergence speed and reduces its efficiency. On the other side, studies have demonstrated that the tent mapping exhibits better traversal uniformity compared to the logistic mapping[30]. In recent years, tent chaos mapping has been applied more and more widely, such as introducing tent chaos mapping into the modified whale optimization algorithm to increase the diversity and randomness of the initial population[31]. Consequently, this study employs the tent mapping for population initialization.

The tent mapping expression can be expressed as follows:

$$x_{t+1} = \begin{cases} \dfrac{x_t}{0.5}, & 0 \leqslant x_t < 0.5; \\ \dfrac{1-x_t}{1-0.5}, & 0.5 \leqslant x_t \leqslant 1 \end{cases} \tag{33}$$

Assume that the initial population consists of d-dimensional individuals. Firstly, d-dimensional vectors

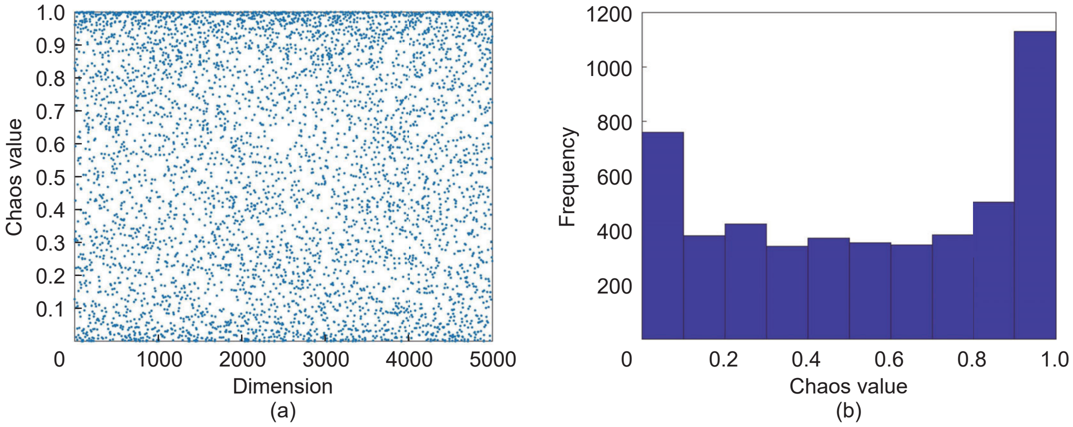


Fig. 4 Logistic chaotic map.

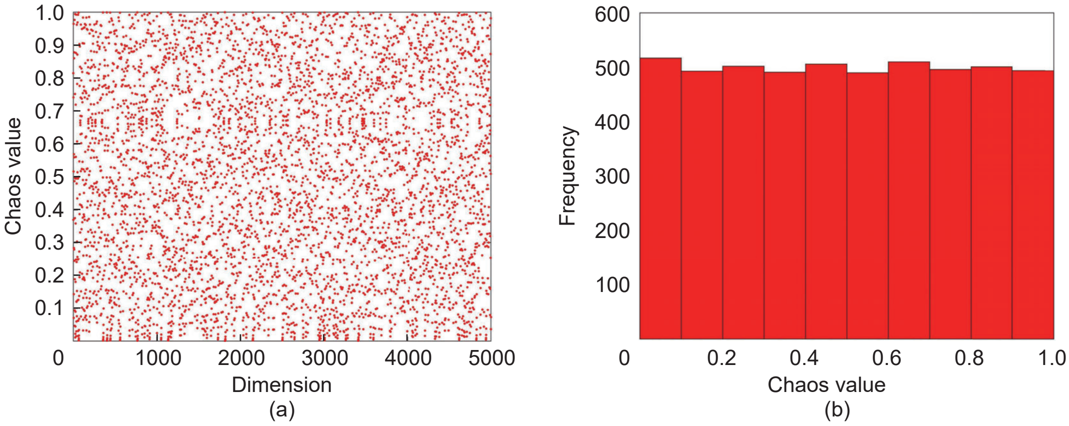


Fig. 5 Tent chaotic map.

with values within the interval $[0,1]$ are randomly generated as the initial individuals in the chaotic space and labeled with $t=1$. The t-th generation of individuals is iterated according to Eq. (33) with $t=t+1$ until $t=M$, yielding chaotic sequences. Finally, the resulting chaotic sequences are mapped to the population individuals according to Eq. (34).

$$Y = X_{\text{lb}} + (X_{\text{ub}} - X_{\text{lb}})\,x_{t+1} \tag{34}$$

where Y is the mapped golden jackal individual, and X_{ub} and X_{lb} are the upper and lower bounds of the individual in each dimension, respectively.

3.2.2 Nonlinear control parameter design

According to Eq. (23), the parameter E plays a key role in coordinating the global and local exploitation of the GJO. When $|E|>1$, the population expands the search range to find better candidate solutions, i.e., global exploration capability. When $|E|<1$, the population narrows the search scope to perform fine search in the local area, i.e., local exploitation capability. Meanwhile, according to Eq. (25), it is known that the E value varies with the change of the control parameter e_1 during the evolutionary process, so the exploration and exploitation capability of the GJO depends on the change of the control parameter e_1. By Eq. (27), it can be seen that the control parameter e_1 decreases linearly with the increase in the number of iterations. However, the optimization process of the GJO is very complex, and the linearly decreasing parameter e_1 changes cannot fully reflect the actual optimization search process of the algorithm.

Therefore, a new nonlinear control parameter e_1 is designed by

$$e_1(t) = e_{\text{ini}} - (e_{\text{ini}} - e_{\text{fin}})\left(\frac{t}{T_{\max}}\right)^{\frac{1}{2}} \tag{35}$$

where e_{ini} and e_{fin} are the initial and final values of the control parameters, respectively.

In order to show the performance of the proposed control parameter e_1, they are compared with the linear control parameter e_2 in this paper. The linear control parameters is as follows:

$$e_2(t) = e_{\text{ini}} - (e_{\text{ini}} - e_{\text{fin}})\left(\frac{t}{T_{\max}}\right) \tag{36}$$

The control parameters e_1 and e_2 and convergence factors E_1 and E_2 are depicted in Fig. 6. Figure 6 clearly demonstrates the superior nonlinear characteristics of the proposed control parameter e_1. Moreover, the proposed convergence factor E_1 decays rapidly in the early stage, enhancing the local search and expediting the optimization speed of the algorithm. Subsequently, the convergence factor E_1 decays slower in the later stage, preventing the algorithm from becoming trapped in local minima.

3.2.3 CS algorithmic strategy

The GJO algorithm location updating approach exhibits noticeable flaws, including a weak global search and

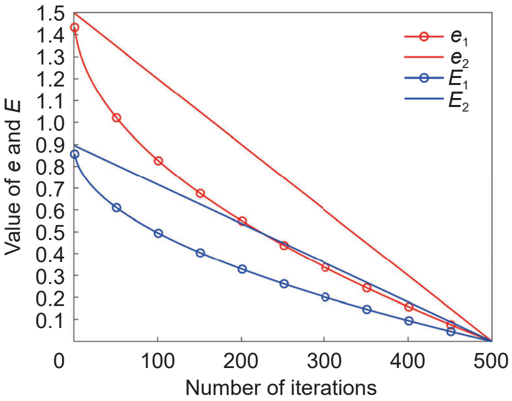


Fig. 6 Variation of control parameters and convergence factors.

susceptibility to becoming trapped in local optimal solutions. In contrast, the proposed CS algorithm, based on the parasitic feeding mechanism of cuckoos and Lévy flight, can solve this problem well. The CS algorithm has several advantages, including the elimination of frequent parameter matching in solving optimization problems, as well as requiring fewer parameter settings and being easy to implement. Also, the algorithm demonstrates nearly equal probability in selecting the search step size and a high level of randomness in direction selection. The algorithm easily transitions from the current region to other regions, thus facilitating global search. Taking advantage of the above advantages of the CS algorithm, this section combines it with the GJO algorithm in order to improve the performance of the GJO algorithm and mitigate its tendency to converge on local minima. In the CS algorithm, the initial solution represents the existing eggs in the host nest, while the new solution generated represents the location where the cuckoo lays its egg. The implementation of the algorithm follows three assumption rules:

(1) Cuckoos randomly select a host nest to lay their eggs, and they only lay one egg at a time. Consequently, there will ultimately be only one optimal solution.

(2) The best host nest, along with its cuckoo egg having the highest priority (i.e., the optimal solution), is retained until the next iteration.

(3) The host bird decides, based on probability, whether to search for cuckoo eggs.

Based on these assumptions, the foraging path and location update method of the cuckoo's nest can be deduced[23]. Additionally, the iteration update of the host nest can be developed as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda), \quad i = 1, 2, \ldots, N \tag{37}$$

where N is the number of host nests, $\oplus$ denotes entry-wise multiplications, x_i^t denotes the position coordinate of the i-th host nest in the t-th iteration. α takes the value of $O(1)$. The $\text{Lévy}(\lambda)$ can be described mathematically as

$$\text{Lévy} \sim u = t^{-\lambda}, \quad 1 < \lambda < 3 \tag{38}$$

3.3 CS-GJO algorithm flow and planning method framework

The detailed flow of the CS-GJO algorithm is shown in Fig. 7.

The pseudocode for the CS-GJO algorithm is shown in Algorithm 1.

In summary, Fig. 8 illustrates the overall framework of the collaborative trajectory planning method for multi-UAV 3D terrain based on the CS-GJO algorithm.

In this paper, hierarchical modeling is used to plan the flight path of single UAV and multi-UAV, respectively. The specific steps of track planning are as follows:

(1) Construct the mission environment, set the terrain, threat, target, and other information, and define the parameters of the UAV itself.

(2) The objective function of single UAV track planning is established.

(3) The algorithm is initialized and the initial population is generated using tent chaotic mapping.

(4) The value of the objective function is calculated, and the nonlinear control parameter design is adopted for E.

(5) Perform the GJO algorithm optimization process.

(6) The CS strategy was adopted to update the population, and the results were output after the maximum number of iterations was reached.

(7) Multiple alternative flight paths are generated, and cubic B-splines are used to smooth the path.

(8) The objective function of multi-UAV cooperative flight path planning is constructed.

(9) CS-GJO algorithm was used to solve the problem Steps (3)−(6), the final flight path of the UAV was determined, and the optimal collaborative planning scheme was obtained.

4 Simulation Verification and Analysis

All experiments in this study were conducted by using an AMD Ryzen 7 5700X processor running at 3.40 GHz with 8 GB of RAM. MATLAB-R2022b was used for the simulation tests.

The simulation experiments focused primarily on two aspects. The first part involved single UAV trajectory planning experiment, wherein trajectory planning was performed using GWO, PSO, GJO, and CS-GJO algorithms individually. This aimed to verify the performance of the proposed improved algorithms and the effectiveness of the single UAV planning model. The second part involved multi-UAV collaborative trajectory planning experiments, which integrated the results from the single UAV trajectory planning. Multi-UAV experiments further validate the performance of the CS-GJO algorithm. The parameter settings for all

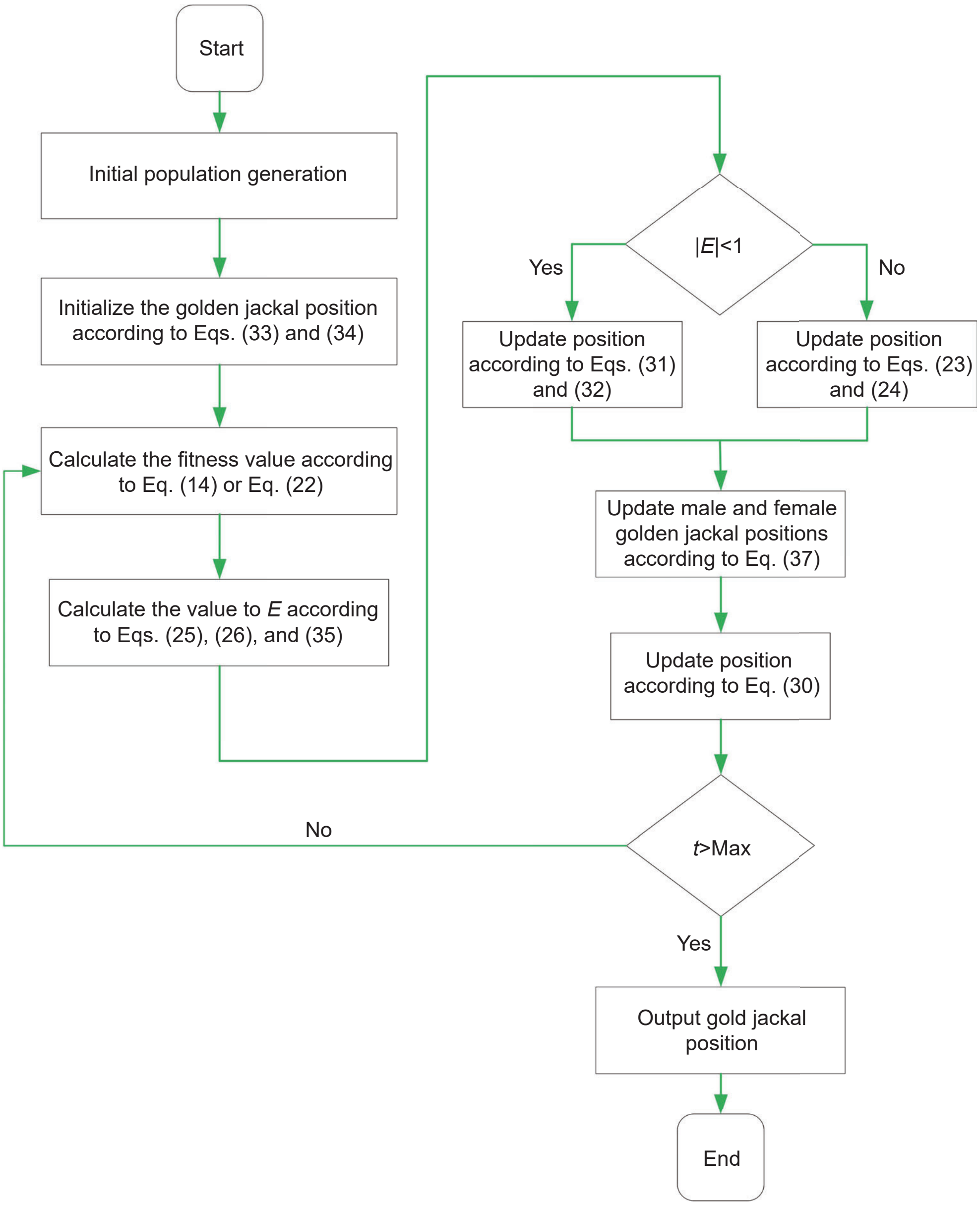

Start
Initial population generation
Initialize the golden jackal position according to Eqs. (33) and (34)
Calculate the fitness value according to Eq. (14) or Eq. (22)
Calculate the value to E according to Eqs. (25), (26), and (35)
|E|<1
Yes
No
Update position according to Eqs. (31) and (32)
Update position according to Eqs. (23) and (24)
Update male and female golden jackal positions according to Eq. (37)
Update position according to Eq. (30)
t>Max
No
Yes
Output gold jackal position
End

Fig. 7 Flowchart of the CS-GJO algorithm.

the algorithms are shown in Table 1.

The experiment difined the task area as $50\,\text{km}\times 50\,\text{km}$, $H_{\min}=0.02\,\text{km}$, $H_{\max}=10\,\text{km}$, flight velocity $v_{\rm i}\in[103\,\text{m/s}, 240\,\text{m/s}]$, the minimum track distance $L_{\min}=0.5\,\text{km}$, the maximum flight distance $L_{\max}$ is 1.5 times the starting distance. Flight turning angle $\varphi_m\in(-60^\circ, 60^\circ)$, flight climb angle $\theta_m\in(-60^\circ, 60^\circ)$, weighting factors $\varepsilon_1=0.7$, $\varepsilon_2=0.05$, $\varepsilon_3=0.05$, and $\varepsilon_4=0.2$, and constraint penalty factor $\eta=10^6$.

4.1 Trajectory smoothing

By using an intelligent optimization algorithm to solve the trajectory planning model, a set of discrete waypoint sequences that satisfy the constraints can be obtained. Connecting these waypoints linearly enables the generation of a pre-planned trajectory for the UAV. However, as the pre-planned trajectory consists of a continuous folded line, it may result in increased flight losses for the UAV. Therefore, the use of the cubic B-spline curve fitting algorithm enables a smoother trajectory that can enhance the flight stability and overall performance of the UAV. The geometric invariance, convexity preservation, variance reduction, and local support properties make the cubic B-spline curve an excellent option for improving the quality of the pre-planned trajectory.

The cubic B-spline function in 3D space can be

Algorithm 1 CS-GJO pseudo-code

```
Input: number of UAV waypoints $M$; search population size $N$; the maximum number of search steps $t_{\max}$; and dimension $D$
Output: optimal golden jackal position $Y_{\text{best}}(t)$ and its fitness value
Initial the golden jackal population according to Eqs. (33) and (34) (tent chaotic map)
if conduct single UAV track planning experiment
    Calculate the fitness value according to Eq. (14), get the optimal position $Y_{\text{best}}(t)$
else
    Calculate the fitness value according to Eq. (22), get the optimal position $Y_{\text{best}}(t)$
end if
while $t < t_{\max}$
    Calculate the value of $E$ according to Eqs. (25), (26), and (35) (nonlinear control parameter design)
    if $|E| < 1$
        Update position according to Eqs. (31) and (32)
    else
        Update position according to Eqs. (23) and (24)
    end if
    Update male and female golden jackal positions according to Eq. (37) (CS algorithmic strategy)
    Update position according to Eq. (30)
    Fitness evaluation and update $Y_{\text{best}}(t)$
    $t = t+1$
end while
Return $Y_{\text{best}}(t)$
```

expressed as follows:

$$p(u)=\sum_{m=0}^{M} L_m N_{m,3}(u) \tag{39}$$

$$N_{m,0}(u)=\begin{cases}1, & u_m \leqslant u \leqslant u_{m+1};\\ 0, & \text{other}\end{cases} \tag{40}$$

$$\begin{aligned}N_{m,k}(u)=&\frac{u-u_m}{u_{m+k}-u_m}N_{m,k-1}(u)+\\ &\frac{u_{m+k+1}-u}{u_{m+k+1}-u_{m+1}}N_{m+1,k-1}(u),\\ &k=1,2,3\end{aligned} \tag{41}$$

where L_m represents the m-th track point, $N_{m,3}(u)$ denotes the basis function of the cubic B-spline curve.

$$\begin{cases}u_0=u_1=\cdots=u_k=0;\\ u_j=u_{j-1}+\dfrac{1}{m-2k}, j\in[k+1,m-k-1];\\ u_{m-k}=u_{m-k+1}=\cdots=u_m=1\end{cases} \tag{42}$$

The smoothing results of the trajectory after cubic B-spline curve fitting are shown in Fig. 9.

From Fig. 9, the cubic B-spline curve fitting algorithm effectively reduces the trajectory length, thereby minimizing fuel consumption. Moreover, the fitted trajectory is smoother and closer to the real flight trajectory of the UAV.

4.2 Terrain construction

In this paper, a digital elevation map is used to establish a three-dimensional mountainous terrain that is essentially in line with the actual environment, as described as follows:

$$\varTheta=\left\{(x,y,z)\left|\ 0\leqslant x\leqslant x_{\max}, 0\leqslant y\leqslant y_{\max},\ 0\leqslant z\leqslant z_{\max}\right.\right\} \tag{43}$$

where (x,y,z) represents the 3D coordinates of any location in the terrain, $x_{\max}$, $y_{\max}$, and $z_{\max}$ represent the boundary values of the terrain. At the meantime, to enhance the realism and credibility of the simulation results, this paper incorporates real-world scenarios by defining threat areas. Please refer to Table 2 for detailed information on these threat areas.

The 3D map with threat areas, as constructed in this paper, is displayed in Fig. 10, where the yellow sphere represents radar threats and the red cylinder represents other threats such as no-fly zones.

4.3 Experimental analysis of ablation by CS-GJO algorithm

Set the starting position of the UAV trajectory to be $(3\text{ km}, 6\text{ km}, 8\text{ km})$, the end position is $(48\text{ km}, 49\text{ km}, 5\text{ km})$. The number of UAV waypoints is 20, the dimension is 30, the maximum number of search steps is 400, and the search population size is 90. The CS-GJO algorithm, GJO algorithm, and three GJO algorithms with a single enhancement strategy—CS-GJO1 algorithm (tent chaos map), CS-GJO2 algorithm (nonlinear control parameter design), and CS-GJO3 algorithm (CS algorithm strategy) were used in twenty independent ablation experiments as comparative algorithms. Table 3 shows the experimental results, and Figs. 11 and 12 show the iteration curve and the flight path planned by each algorithm, respectively.

The results shown in Table 3 show that the objective function values calculated by CS-GJO1, CS-GJO2, CS-GJO3, and CS-GJO algorithms are always superior to the GJO algorithm. Among them, the results of the CS-GJO algorithm are more stable and have the best objective function value. Overall, these experimental results highlight the effectiveness of reinforcement strategies.

As shown in Fig. 11, the convergence accuracy of the GJO algorithm is the worst, and it is easy to fall into

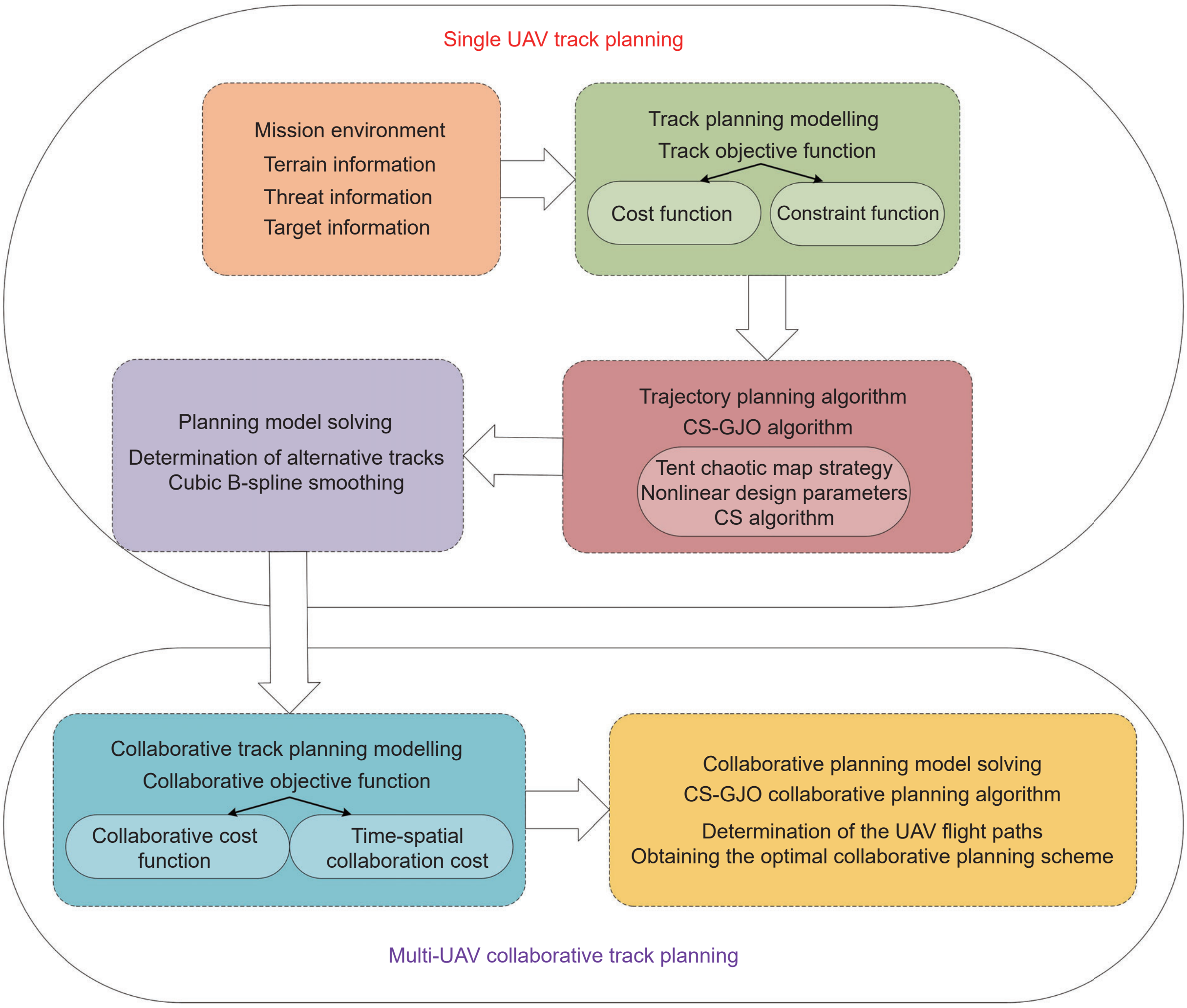

Single UAV track planning
Mission environment
Terrain information
Threat information
Target information
Track planning modelling
Track objective function
Cost function
Constraint function
Trajectory planning algorithm
CS-GJO algorithm
Tent chaotic map strategy
Nonlinear design parameters
CS algorithm
Planning model solving
Determination of alternative tracks
Cubic B-spline smoothing
Collaborative track planning modelling
Collaborative objective function
Collaborative cost function
Time-spatial collaboration cost
Collaborative planning model solving
CS-GJO collaborative planning algorithm
Determination of the UAV flight paths
Obtaining the optimal collaborative planning scheme
Multi-UAV collaborative track planning

Fig. 8 Multi-UAV 3D terrain collaborative trajectory planning framework.

Table 1 Algorithmic parameter.

Algorithm	Parameter
GWO	$a = (2 \rightarrow 0)$
PSO	$c_1 = 2$ $c_2 = 2$ $W_{\min} = 0.2$ $W_{\max} = 0.9$
GJO	$e_1 = (1.5 \rightarrow 0)$
CS-GJO	$e_1 = (1.5 \rightarrow 0)$ $P_a = 0.1$

local optimization. The initial population generated by CS-GJO1 is more uniform, which improves the algorithm's performance. CS-GJO2 and CS-GJO3 have obvious enhancement effects, effectively preventing the algorithm from stagnating. The CS-GJO algorithm combines the advantages of the improved strategy and greatly improves the convergence performance of the algorithm. It can be seen from Figs. 12a and 12b that all algorithms can plan feasible flight paths. However, the flight path planned by the GJO algorithm has too much redundancy. CS-GJO1, CS-GJO2, and CS-GJO3 reduce the redundancy, but they are not optimal. The flight path planned by CS-GJO has no redundancy and is the optimal flight path. In summary, the ablation

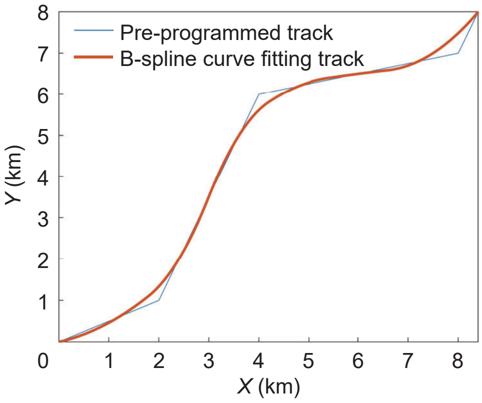


Fig. 9 Cubic B-spline curve smoothing effect.

Table 2 Threat area parameter settings.

Threat type	Coordinate (km)	Height (km)	Threat radius (km)
Radar threat	(12, 25)	0	5
Missile, artillery, meteorological threats	(45, 32)	10	4
	(16, 6)	10	4
	(29, 35)	10	5
	(33, 18)	10	5

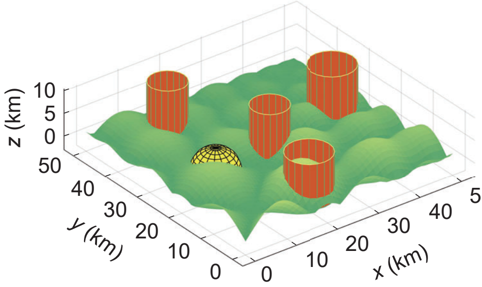


Fig. 10 3D topographic maps with threat areas.

Table 3 Results of single UAV trajectory planning ablation experiment.

Algorithm	Average time spent (s)	Objective function value			
		Optimum value	Bottom value	Average value	Standard deviation
GJO	37.9320	46.6042	60.7699	53.2528	4.0826
CS-GJO1	37.8564	45.3700	59.1816	52.8434	2.0758
CS-GJO2	37.7277	46.2535	58.5856	51.8413	4.3523
CS-GJO3	38.4479	45.7386	56.6078	49.1970	2.9424
CS-GJO	38.0967	**44.5822**	**52.5934**	**47.3766**	**2.0758**

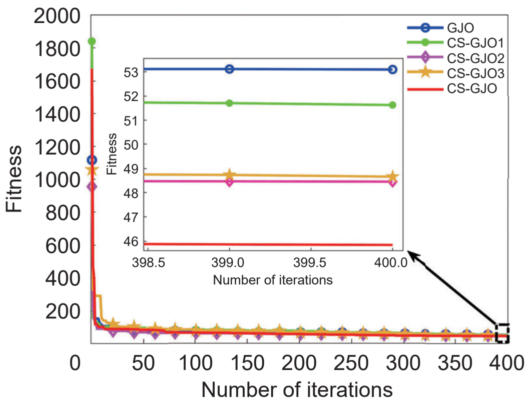


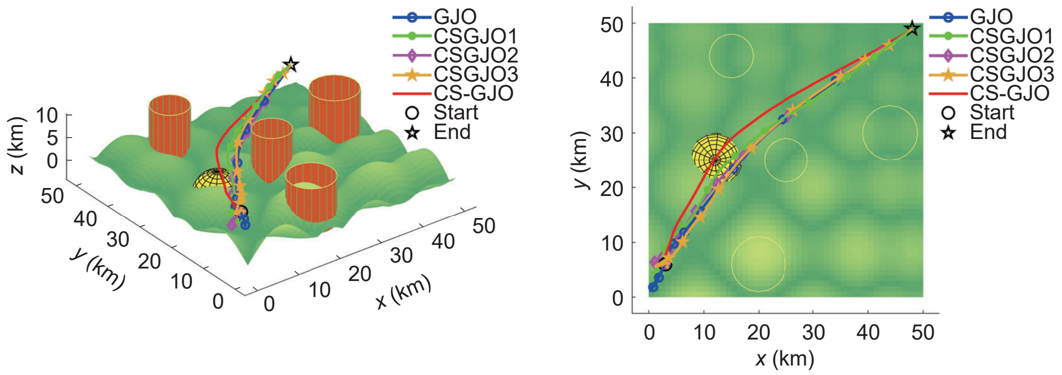
Fig. 11 Convergence curve of the objective function of ablation experiment.

experiments prove the effectiveness of the proposed tent chaos map, nonlinear parameter design, and CS algorithm strategy.

4.4 Experimental analysis of single UAV trajectory planning

The algorithm parameter settings are the same as in Section 4.3. The GWO algorithm[25], PSO algorithm[24], GJO algorithm[17], and CS-GJO algorithm are employed in the single UAV trajectory planning model. To validate the optimization performance of the proposed improvement strategy and the CS-GJO algorithm, each algorithm is independently executed for 20 iterations. The simulation results are presented in Table 4. Figure 13 displays the trajectories map and top view of the trajectory generated by each algorithm in the single UAV trajectory planning model. Additionally, Fig. 14 illustrates the convergence curve plot of the objective function.

Based on the data provided in Table 4, the CS-GJO algorithm is superior to the GWO, PSO, and GJO



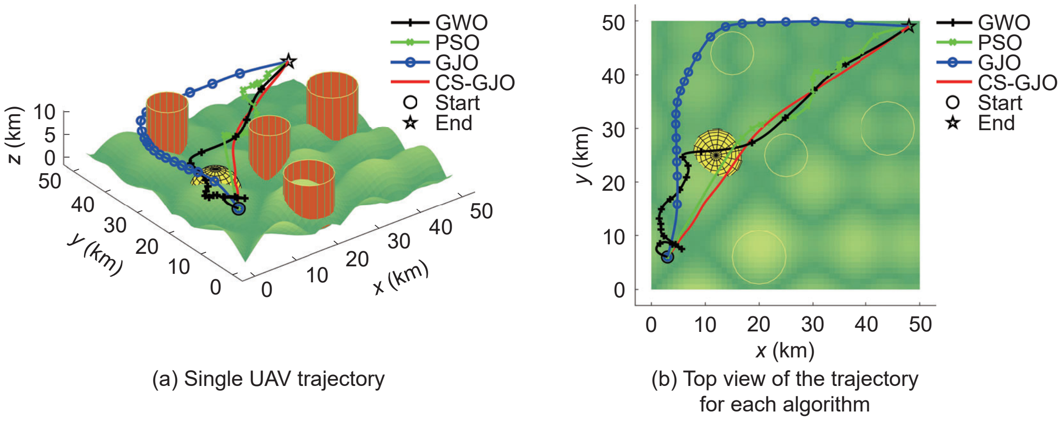
(a) Single UAV trajectory

(b) Top view of the trajectory for each algorithm

Fig. 12 Side and top views of the single UAV trajectory from the ablation experiment.

Table 4 Table of statistical results of single UAV trajectory planning information.

Algorithm	Average time spent (s)	Objective function value			
		Optimum value	Bottom value	Average value	Standard deviation
GWO	36.6177	47.6226	55.1905	49.2123	2.9462
PSO	36.6791	48.4725	54.5627	49.1787	0.6149
GJO	37.5859	45.3583	57.1209	52.7080	3.2419
CS-GJO	38.0007	**44.7725**	**50.5844**	**47.9822**	1.2379



(a) Single UAV trajectory

(b) Top view of the trajectory for each algorithm

Fig. 13 Side and top views of the single UAV trajectory map.

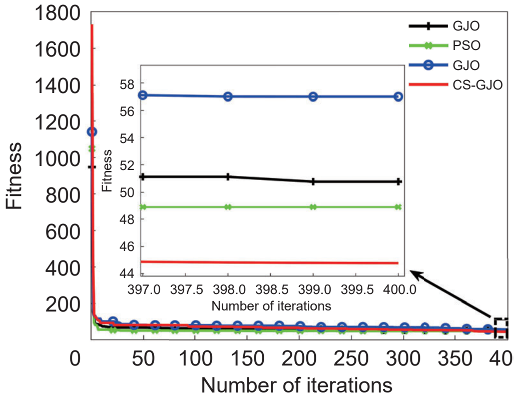


Fig. 14 Convergence plot of the objective function for single UAV trajectory planning.

algorithms in all aspects. This indicates its strong ability to find the optimal solution. The standard deviation of the CS-GJO algorithm is the second only to that of the PSO algorithm, but the PSO algorithm's convergence values are generally poor, suggesting that the PSO is stuck in local minima at this point. Furthermore, the improved algorithm demonstrates an approximate 9% improvement in optimization accuracy compared to the GJO algorithm, thereby confirming the efficacy of the proposed enhancement strategy. Although the CS-GJO algorithm takes the longest time, it trades a small amount of time consumption for a significant improvement in algorithm performance. In summary, under the same environmental conditions, the CS-GJO algorithm is able to complete the trajectory planning under various constraints, and the planning line is shorter than other comparative algorithms. Additionally, the CS-GJO algorithm is more prone to escaping local minima, showcasing superior search accuracy and algorithmic robustness compared to other algorithms.

As depicted in Fig. 13a, the CS-GJO algorithm effectively avoids threatening areas and hills, and plans smoother routes. Figure 14 shows that GWO, PSO, and GJO algorithm all fall into a local minima. In contrast, the CS-GJO algorithm both enhances the group's ability to seek the global minima and jumps out of the local minima. This improvement is achieved by incorporating the tent chaotic mapping strategy and the nonlinear control parameter design method into the GJO algorithm and combining it with the CS algorithm. From Fig. 13b, it is evident that the CS-GJO exhibits shorter planned routes and smoother trajectories. The CS-GJO algorithm effectively reduces flight losses for the UAV, improves UAV maneuverability, and increases mission completion efficiency. The above experiments demonstrate that the CS-GJO algorithm outperforms all comparison algorithms.

4.5 Experimental analysis of multi-UAV collaborative trajectory planning

In this section, the CS-GJO algorithm will be used to address the model for cooperative trajectory planning in a multi-UAV scenario. The starting positions are UAV_1: $(3\,\mathrm{km}, 4\,\mathrm{km}, 8\,\mathrm{km})$, UAV_2: $(6\,\mathrm{km}, 3\,\mathrm{km}, 7\,\mathrm{km})$, UAV_3: $(4\,\mathrm{km}, 6\,\mathrm{km}, 6\,\mathrm{km})$, respectively, the end positions are UAV_1: $(49\,\mathrm{km}, 48\,\mathrm{km}, 5\,\mathrm{km})$, UAV_2: $(46\,\mathrm{km}, 50\,\mathrm{km}, 4\,\mathrm{km})$, UAV_3: $(49\,\mathrm{km}, 45\,\mathrm{km}, 4.5\,\mathrm{km})$. Now the UAVs are required to fly from their respective starting positions to the target end position to perform the cooperative work task. The minimum safe distance between each UAVs set to $d_{\mathrm{s}} = 0.5\,\mathrm{km}$. The number of alternative trajectories is set to 20, and the other parameters of the UAVs and algorithms are set as in Section 4.3. The results of the comparison of the objective functions for multi-UAV alternative trajectory planning and the results for multi-UAV cooperative trajectory planning are presented in Tables 5 and 6, respectively.

Figure 15a illustrates the 3D roadmap for collaborative trajectory planning using the CS-GJO algorithm. Figure 15b presents the top view of the trajectory. Figure 16 displays the convergence of the fitness curves for the multi-UAVs. Figure 17 depicts the variation in the deflection angles among the three UAVs. Finally, Fig. 18 shows the variation in the altitude differences observed among the three UAVs.

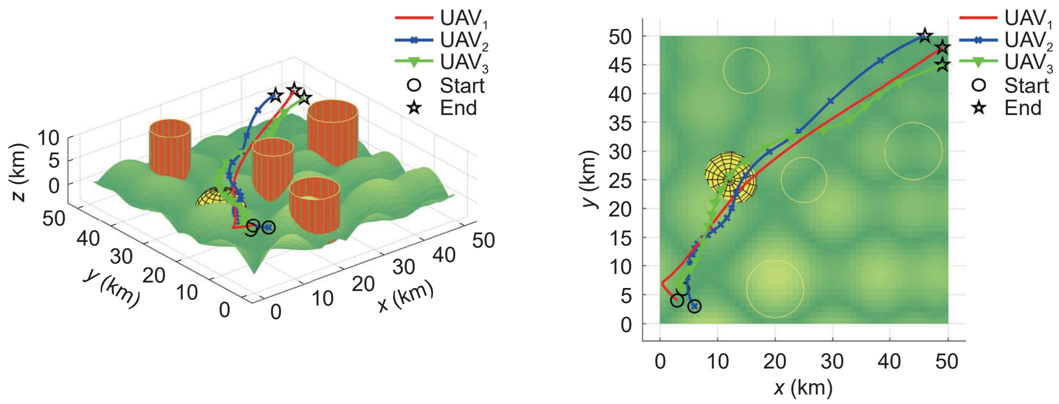
As observed in Fig. 15, even in the context of multi-UAV cooperative planning, the CS-GJO algorithm effectively avoids threat areas, overcomes constraints, and plans smooth flight paths. Figure 16 demonstrates that the CS-GJO algorithm exhibits rapid convergence speed, continuously escaping local minima to converge towards more optimal solutions. The data from Fig. 18 exhibit a relatively stable flight with a small variation in altitude difference. In addition, the information in Fig. 17 shows that the deflection angle stays within the specified range, i.e., it is less than the set maximum turning angle and climb angle $\left(\frac{\pi}{3}\right)$. This indicates that the trajectories planned using the CS-GJO algorithm are smoother. These outcomes collectively suggest that

Table 5 Comparative results of multi-UAV alternative trajectory planning.

Algorithm	UAV No.	Objective function value			
		Optimum value	Bottom value	Average value	Standard deviation
GWO	1	49.1228	54.8243	51.6714	2.1991
	2	48.5175	53.5848	50.1253	3.0524
	3	47.3773	52.3197	49.6172	1.2875
PSO	1	49.6621	52.9893	51.1878	0.9010
	2	47.9278	51.8991	49.8737	1.0257
	3	46.9482	50.6835	48.737	0.8877
GJO	1	49.5255	65.4658	53.4693	5.1322
	2	44.6583	54.2795	49.7867	5.0824
	3	43.6932	58.1553	51.2396	4.0379
CS-GJO	1	**46.0684**	**52.6829**	**48.5677**	2.5872
	2	**44.6197**	**51.2897**	**47.8171**	3.9861
	3	**43.6384**	53.3382	**47.2746**	3.2422

Table 6 Comparative results of multi-UAV collaborative trajectory planning.

Algorithm	UAV No.	Tracks cost	Voyage (km)	Arrival time (s)	Coordinated arrival time (s)	Collaboration cost
GWO	1	50.89	74.95	[314.92, 734.81]	[320.50, 734.81]	434.29
	2	51.50	76.24	[320.32, 747.42]		
	3	50.59	76.28	[320.50, 747.84]		
PSO	1	51.56	75.83	[349.32, 763.97]	[349.32, 754.66]	469.06
	2	48.94	72.47	[319.75, 754.66]		
	3	48.81	71.36	[308.71, 796.38]		
GJO	1	56.67	84.51	[355.07, 828.49]	[355.07, 762.35]	477.74
	2	53.08	77.76	[326.72, 762.35]		
	3	54.27	80.85	[339.72, 792.68]		
CS-GJO	1	**47.07**	**69.96**	[293.93, 685.85]	**[313.92, 685.85]**	**391.73**
	2	**48.18**	**71.82**	[313.92, 732.49]		
	3	**47.06**	**70.71**	[301.75, 704.08]		



(a) CS-GJO algorithm for multi-UAV 3D collaborative trajectory planning

(b) Top view of flight path

Fig. 15 CS-GJO algorithm for multi-UAV 3D collaborative trajectory planning roadmap.

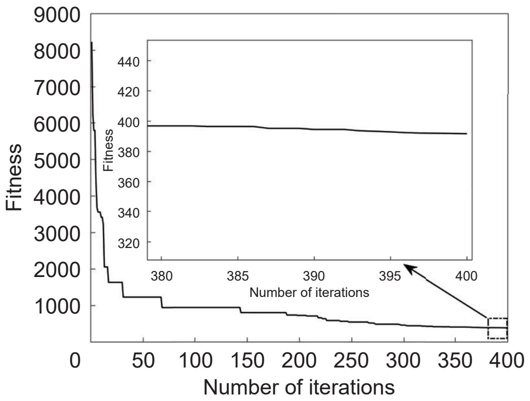


Fig. 16 Convergence curve of multi-UAV collaboration objective function.

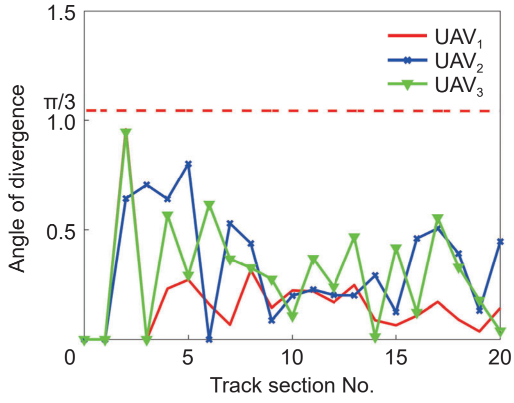


Fig. 17 Change in deflection angle.

the CS-GJO algorithm is effective in ensuring trajectory smoothness and stability.

Table 5 illustrates that all three algorithms consistently generate alternative trajectory groups which meet the constraints for each UAV, with the CS-GJO algorithm yielding better alternatives compared to GWO, PSO, and GJO algorithm. Based on the results presented in Table 5, Table 6 showcases the obtained multi-UAV cooperative planning scheme. The CS-GJO algorithm outperforms GWO, PSO, and GJO algorithm in terms of planned trajectory length for all three UAVs. Additionally, the collaboration cost of the CS-GJO algorithm is the lowest among the four algorithms, amounting to 391.73. The UAVs using the CS-GJO algorithm can reach the end point within the time range of [313.92 s, 685.85 s], indicating enhanced task efficiency. It is worth noting that the trajectory selected by each UAV may not be the optimal one among the alternative trajectory groups. This suggests that the cooperative trajectory planning scheme prioritizes trajectories that fulfill cooperative requirements and are relatively better, rather than strictly selecting the optimal trajectories.

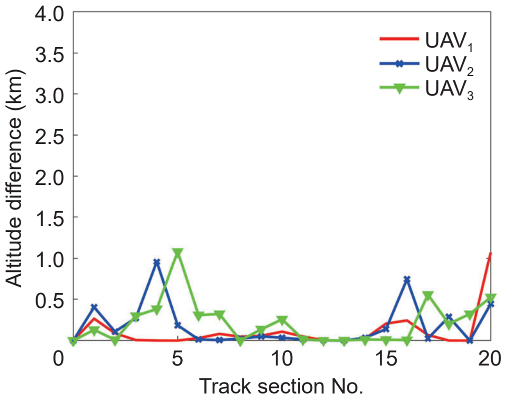


Fig. 18 Altitude difference variation.

In conclusion, the CS-GJO algorithm outperforms GWO, PSO, and GJO algorithm in complex environments, and is capable of generating stable and smooth flyable paths quickly. This confirms the strong algorithmic performance of the CS-GJO algorithm and presents a novel approach for solving multi-UAV trajectory planning problems.

5 Conclusion and Recommendation

To address the issues of low accuracy, instability, and slow convergence speed in multi-UAV 3D collaborative trajectory planning solutions, this paper introduces the CS-GJO algorithm, which incorporates three key modifications. Firstly, the tent chaotic mapping is utilized for population initialization to enhance population diversity. Secondly, the CS algorithm is integrated with the GJO algorithm to enhance the GJO algorithm’s performance. Finally, specific nonlinear parameters are designed. In the early stages of the algorithm, these parameters help to accelerate convergence. In later stages, this tuning enhances the global search of the algorithm.

In this paper, a single UAV trajectory planning and a multi-UAV collaborative trajectory planning model are established by using a hierarchical planning method. The CS-GJO algorithm is proposed and applied to carry out simulation experiments on these models, and the following conclusions are drawn:

(1) In single UAV trajectory planning, the CS-GJO algorithm has higher convergence accuracy and stronger robustness compared to other algorithms such as the GWO, PSO, and GJO algorithms. It consistently generates high-quality flight trajectories even in complex mission environments, effectively meeting the mission requirements of UAV.

(2) The CS-GJO algorithm is better than the GWO, PSO, and GJO algorithms in multi-UAV collaborative trajectory planning. The CS-GJO algorithm can shorten the planning time, optimize the planning paths, and enhance stability.

The proposed multi-UAV 3D collaborative trajectory planning method performs well in static planning. However, the algorithm is time-consuming and less effective when applied to real-time collaborative trajectory planning with multiple UAVs. To address this issue, future work will focus on enhancing both the model and algorithm specifically for dynamic planning problems.

Acknowledgment

This work was supported by the Key Research and Development Program of Henan Province (No. 241111222900), Natural Science Foundation of Henan (No. 242300421716), Key Science and Technology Program of Henan Province (Nos. 242102220044 and 242102210034), National Natural Science Foundation of China (No. 62103379), and Maker Space Incubation Project (No. 2023ZCKJ102).

References

[1] F. Santoso, M. A. Garratt, and S. G. Anavatti, State-of-the-art intelligent flight control systems in unmanned aerial vehicles, *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 613–627, 2018.

[2] H. Chen, X. Wang, Y. Jiao, and Y. Li, An algorithm of coverage flight path planning for UAVs in convex polygon areas, (in Chinese), *Acta Aeronautica et Astronautica Sinica*, vol. 31, no. 9, pp. 1802–1808, 2010.

[3] G. Jia and J. Wang, Research review of UAV swarm mission planning method, (in Chinese), *Systems Engineering and Electronics*, vol. 43, no. 1, pp. 99–111, 2021.

[4] D. Hsu, J. C. Latombe, and R. Motwani, Path planning in expansive configuration spaces, in *Proc. Int. Conf. Robotics and Automation*, Albuquerque, NM, USA, 1997, pp. 2719–2726.

[5] X. Wu, L. Xu, R. Zhen, and X. Wu, Biased sampling potentially guided intelligent bidirectional RRT algorithm for UAV path planning in 3D environment, *Math. Probl. Eng.*, vol. 2019, no. 1, p. 5157403, 2019.

[6] D. Mandloi, R. Arya, and A. K. Verma, Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3D environment, *Int. J. Syst. Assur. Eng. Manag.*, vol. 12, no. 5, pp. 990–1000, 2021.

[7] J. Sun, J. Tang, and S. Lao, Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm, *IEEE Access*, vol. 5, pp. 18382–18390, 2017.

[8] A. Tahir, J. Böling, M. H. Haghbayan, H. T. Toivonen, and J. Plosila, Swarms of unmanned aerial vehicles—A survey, *J. Ind. Inf. Integr.*, vol. 16, p. 100106, 2019.

[9] L. Liu, D. Liu, X. Wang, J. Chen and D. Liu, Development status and outlook ot UAV clusters and anti-UAV clusters, (in Chinese), *Acta Aeronautica et Astronautica Sinica*, vol. 43, no. S1, p. 726908, 2022.

[10] Q. Zong, D. Wang, S. Shao, B. Zhang, and Y. Han, Research status and development of multi UAV coordinated formation flight control, (in Chinese), *Journal of Harbin Institute of Technology*, vol. 49, no. 3, pp. 1–14, 2017.

[11] R. Zhou, C. Huang, Z. Wei, and K. Zhao, Application of MP-GWO algorithm in multiple cooperating UCAV path planning, (in Chinese), *Journal of Air Force Engineering University*, vol. 18, no. 5, pp. 24–29, 2017.

[12] D. Zhou, P. Wang, X. Li, and K. Zhang, Cooperative path planning of multi-UAV based on multi-objective optimization algorithm, (in Chinese), *Systems Engineering and Electronics*, vol. 39, no. 4, pp. 782–787, 2017.

[13] L. Swartzentruber, J. L. Foo, and E. Winer, Multi-objective UAV path planning with refined reconnaissance and threat formulations, in *Proc. 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Orlando, FL, USA, 2010, p. 2758.

[14] W. Shan, N. Cui, B. Huang, X. Wang, and Y. Bai, Multiple UAV cooperative path planning based on PSO-HJ method, (in Chinese), *Journal of Chinese Inertial Technology*, vol. 28, no. 1, pp. 122–128, 2020.

[15] L. Yang, J. Guo, and Y. Liu, Three-dimensional UAV

cooperative path planning based on the MP-CGWO algorithm, *International Journal of Innovative Computing, Information and Control*, vol. 16, no. 3, pp. 991–1006, 2020.

[16] X. Zhang, S. Xia, X. Li, and T. Zhang, Multi-objective particle swarm optimization with multi-mode collaboration based on reinforcement learning for path planning of unmanned air vehicles, *Knowl. Based Syst.*, vol. 250, p. 109075, 2022.

[17] N. Chopra and M. M. Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, *Expert Syst. Appl.*, vol. 198, p. 116924, 2022.

[18] E. H. Houssein, D. A. Abdelkareem, M. M. Emam, M. A. Hameed, and M. Younan, An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm, *Comput. Biol. Med.*, vol. 149, p. 106075, 2022.

[19] M. Rezaie, K. Karamnejadi Azar, A. Kardan Sani, E. Akbari, N. Ghadimi, N. Razmjooy, and M. Ghadamyari, Model parameters estimation of the proton exchange membrane fuel cell by a Modified Golden Jackal Optimization, *Sustain. Energy Technol. Assess.*, vol. 53, p. 102657, 2022.

[20] A. V. Hanafi, A. Ghaffari, H. Rezaei, A. Valipour, and B. Arasteh, Intrusion detection in Internet of Things using improved binary golden jackal optimization algorithm and LSTM, *Clust. Comput.*, vol. 27, no. 3, pp. 2673–2690, 2024.

[21] T. S. Lou, Z. P. Yue, Y. Z. Jiao, and Z. D. He, A hybrid strategy-based GJO algorithm for robot path planning, *Expert Syst. Appl.*, vol. 238, p. 121975, 2024.

[22] Z. Teng, J. Lv, L. Guo, and Y. Xu, An improved hybrid grey wolf optimization algorithm based on tent mapping, (in Chinese), *Journal of Harbin Institute of Technology*, vol. 50, no. 11, pp. 40–49, 2018.

[23] 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, 2013.

[24] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. ICNN'95—Int. Conf. Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.

[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[26] S. Yan, P. Yang, W. Liu, X. Li, J. Lei, and C. Zhao, Multi-UAV trajectory planning for complex terrain based on GPSSA algorithm, (in Chinese), *Journal of Beijing University of Aeronautics and Astronautics*, doi: 10.13700/j.bh.1001-5965.2022.0984.

[27] M. D. Phung and Q. P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization, *Appl. Soft Comput.*, vol. 107, p. 107376, 2021.

[28] L. Liu, C. Yu, Z. Wang, Z. Qi, and T. Long, Fast 3Droute planning method for small UAV, (in Chinese), *Systems Engineering and Electronics*, vol. 35, no. 12, pp. 2521–2526, 2013.

[29] H. Bai, T. Fan, Y. Niu, and Z. Cui, Multi-UAV cooperative trajectory planning based on many-objective evolutionary algorithm, *Complex System Modeling and Simulation*, vol. 2, no. 2, pp. 130–141, 2022.

[30] L. Shan, H. Qiang, J. Li, and Z. Wang, Chaotic optimization algorithm based on tent map, (in Chinese), *Control and Decision*, vol. 20, no. 2, pp. 179–182, 2005.

[31] Y. Li, M. Han, and Q. Guo, Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization, *KSCE J. Civ. Eng.*, vol. 24, no. 12, pp. 3703–3713, 2020.

Taishan Lou received the PhD degree in engineering mechanics from Beihang University, Beijing, China in 2015, the master degree in applied mathematics from Beihang University, Beijing, China in 2009, and the BS in mathematics and applied mathematics from Beihang University, Beijing, China in 2003. He is currently an associate professor at the School of Electrical and Information Enginering, Zhengzhou University of Light Industry, China. His main research interests are filtering theory, control theory, navigation, and data fusion. He has published and co-authored more than 60 papers in both national and international publications.

Zhepeng Yue received the BEng degree from Zhengzhou University of Industrial Technology, Zhengzhou, China, in 2020. He is currently pursuing the MS degree at the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His main research interests include path planning and nonlinear filtering.

Yu Wang received the BS degree in engineering from Zhoukou Normal University, Zhoukou, China in 2022. He is currently pursuing the MS degree at the School of Electrical and Information Enginering, Zhengzhou University of Light Industry, Zhengzhou, China. His main research interests are multi-UAV collaboration and nonlinear filtering.

Liangyu Zhao received the BSc and PhD degrees from Beijing Institute of Technology, China in 2003 and 2008, respectively. Now he is a professor at Beijing Institute of Technology, China. His main research interests include civil aircraft design, navigation guidance and control, and system integration and simulation.