Journal Pre-proof

Ground threat prediction-based path planning of unmanned autonomous helicopter using hybrid enhanced artificial bee colony algorithm

Zengliang Han, Mou Chen, Haojie Zhu, Qingxian Wu

PII: S2214-9147(23)00107-1

DOI: https://doi.org/10.1016/j.dt.2023.04.010

Reference: DT 1232

To appear in: *Defence Technology*

Received Date: 12 February 2023

Revised Date: 20 March 2023

Accepted Date: 11 April 2023

Please cite this article as: Han Z, Chen M, Zhu H, Wu Q, Ground threat prediction-based path planning of unmanned autonomous helicopter using hybrid enhanced artificial bee colony algorithm, *Defence Technology* (2023), doi: https://doi.org/10.1016/j.dt.2023.04.010.

Ground threat prediction-based path planning of unmanned autonomous helicopter using hybrid enhanced artificial bee colony algorithm

Zengliang Han, Mou Chen*, Haojie Zhu, Qingxian Wu

School of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

*Corresponding author：chenmou@nuaa.edu.cn

Ground threat prediction-based path planning of unmanned autonomous helicopter using hybrid enhanced artificial bee colony algorithm

Abstract

Unmanned autonomous helicopter (UAH) path planning problem is an important component of the UAH mission planning system. Aiming to reduce the influence of non-complete ground threat information on UAH path planning, a ground threat prediction-based path planning method is proposed based on artificial bee colony (ABC) algorithm by collaborative thinking strategy. Firstly, a dynamic threat distribution probability model is developed based on the characteristics of typical ground threats. The dynamic no-fly zone of the UAH is simulated and established by calculating the distribution probability of ground threats in real time. Then, a dynamic path planning method for UAH is designed in complex environment based on the real-time prediction of ground threats. By adding the collision warning mechanism to the path planning model, the flight path could be dynamically adjusted according to changing no-fly zones. Furthermore, a hybrid enhanced ABC algorithm is proposed based on collaborative thinking strategy. The proposed algorithm applies the leader-member thinking mechanism to guide the direction of population evolution, and reduces the negative impact of local optimal solutions caused by collaborative learning update strategy, which makes the optimization performance of ABC algorithm more controllable and efficient. Finally, simulation results verify the feasibility and effectiveness of the proposed ground threat prediction path planning method.

Keywords: UAH; ABC; Path planning; Ground threat prediction; Hybrid enhanced

1. Introduction

In recent years, it has been demonstrated that unmanned autonomous helicopter (UAH) represents one of the most challenging and high-potential equipments in aeronautics [1-3]. Meanwhile, the path planning mission is becoming one of the key technologies of UAHs and has been widely investigated by scholars around the world [4]. The main objective of UAH path planning is to design a flight path to reach the target point with minimal comprehensive costs, i.e., minimal probability of being destroyed while meeting the UAH performance requirements. The current solution is to plan and generate flight paths offline based on available global and local information, and subsequently adjust the pre-planned paths based on changes in environment, and mission, etc [5]. However, how to combine all possible constraints to generate flight paths efficiently and stably is still an open problem [6].

In general, the path planning problems of UAH can be divided into global flight path planning and local dynamic flight path planning [7]. Before the UAH takes off, the ground command system pre-plans a flight path for UAH on the mission based on the current information available about the mission. So far, many achievements have been made in the research of static path planning in known environment, such as artificial potential field

(APF) [8], rapidly-exploring random tree (RRT) [9], A* algorithm [10], voronoi diagram [11] and probabilistic roadmaps (PRM) [12]. In addition, with the advancement of technology, a large number of computational intelligence (CI) methods have emerged one after another [13,14]. In Ref. [15], a new algorithm named spherical vector-based particle swarm optimization (SPSO) was developed to deal with the problem of path planning for unmanned aerial vehicles (UAVs) in complicated environments subjected to multiple threats. In [16], an improved fireworks algorithm (FWA) and particle swarm optimization (PSO) cooperation algorithm were designed to deal with the UAV global path planning problem. In Ref. [17], a path planning method was proposed based on the multi-strategy evolutionary learning artificial bee colony (MSEL-ABC) algorithm for producing the UAH path planning problem. The results of numerous studies have shown that the above algorithms and their improvements can indeed solve the global path planning problem [18].

However, the modern air combat environment is so varied that the ideal path planning environment does not exist. In order to ensure that UAH can survive in a dynamic environment, its flight path must be dynamically adjusted according to the actual situation, i.e., local dynamic planning [19]. The common planning methods include dynamic path planning method [20], artificial neural network method [21], fuzzy logic method [22], velocity obstacle method [23] and so on. In Ref. [24], a dynamic path planning approach based on neural networks was proposed to evacuation planning in large public buildings. In Ref. [25], the hybrid technology of the dynamic window approach (DWA) and the teaching learning-based optimization (TLBO) was designed for the NAO humanoid robot navigation. In Ref. [26], a hazardous flight region prediction system for small UAVs operated in urban areas was developed using a deep neural network (DNN) to support a risk assessment and safe trajectory planning. Unlike offline global path planning, online dynamic planning depends more on the performance of on-board sensors and computers. Real-world experience shows that for unexpected unknown threats, frequent dynamic planning will inevitably reduce the stability and reliability of the planning system. In fact, the occurrence of this situation is caused by the inadequacy of the global planning system to grasp the battlefield threat information.

Obviously, battlefield information is an important guarantee to support the UAH path planning system. Information about enemy threats is obtained through pre-war intelligence collection and analysis on the one hand [27]. On the other hand, it relies on real-time detection by UAH's own sensor equipment during mission execution. However, pre-war information gathering and dynamic threat detection exacerbate the strain on the path planning system. More reliable ground threat information will directly affect the effectiveness of the UAH path planning system [28]. In other words, the global path planning system with more threat information reference can reduce the number of local dynamic planning, and the whole flight process of UAH can be safer and more stable. Therefore, it is necessary and important to intelligently and efficiently process and predict non-complete ground threat information based on the available information and compensate for the adverse effects caused by ground threats in global path planning.

In this paper, we focus on the path planning problem for UAH based on ground threat prediction. The designed dynamic ground threat prediction algorithm and hybrid enhanced ABC algorithm are able to provide a safe flight path for UAH. The main innovations and contributions of this work are summarized as follows:

(1) A ground threat prediction method based on probability density distribution is proposed. The distribution probabilities of ground threats are calculated and updated in real time based on their instantaneous orientation, velocity, intent and other non-complete information.

(2) A probabilistic diffusion-based dynamic no-fly zones modelling approach is designed to support path planning for UAH with the ground threat prediction. The boundaries of the dynamic no-fly zone are determined and updated in real time by extracting the edge of the probability spread area.

(3) To process the UAH path planning efficiently based on ground threat prediction, a hybrid enhanced ABC algorithm based on collaborative thinking strategy is proposed. Through the leader-member thinking mechanism and collaborative learning strategy, the negative impact of the local optimal solution is weakened, which effectively improves the quality of the UAH flight path.

The rest of this paper is arranged as follows. Section 2 demonstrates the problem formulation. Section 3 investigates the ground threat prediction-based path planning. Section 4 investigates the hybrid enhanced ABC algorithm based on collaborative thinking strategy. Section 5 provides the feasibility and effectiveness of the proposed method by simulation experiment. Finally, a brief summary is given in Section 6.

2. Problem formulation

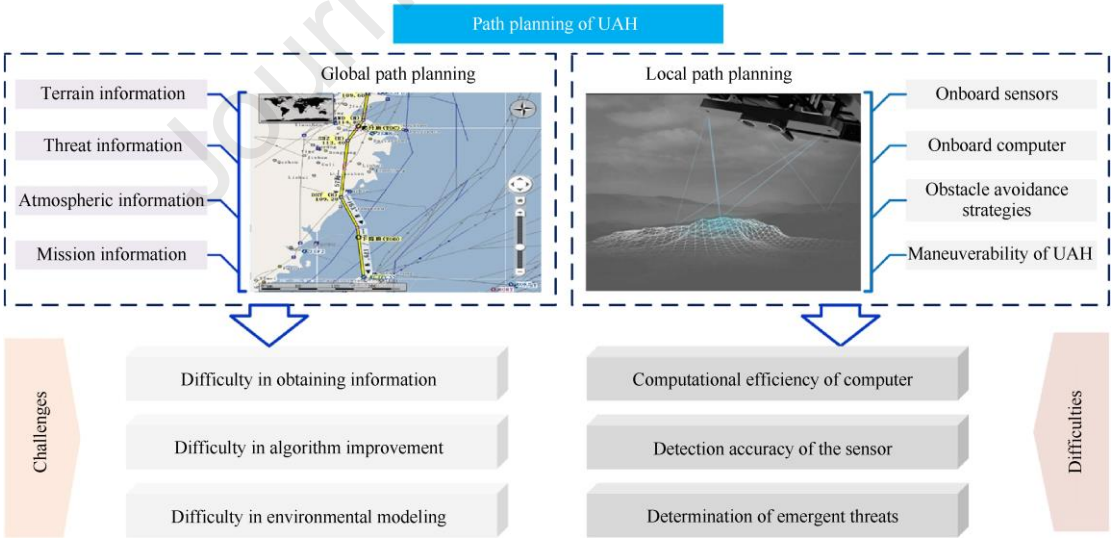


Fig.1. Schematic diagram of path planning classification.

As shown in Fig. 1, the path planning problems can be divided into global planning and local planning [29]. The globally planned flight paths provide the relevant reference confidence for the dynamic adjustment of local planning. In practice, global path planning faces the problem of difficulty in obtaining accurate threat information. In addition, local planning poses a huge challenge to the performance and reliability of the computer and sensors.

Therefore, in order to reduce the frequency of UAH local dynamic planning and improve the stability of the path planning system, it is particularly important to accurately predict dynamic ground threats in global planning.

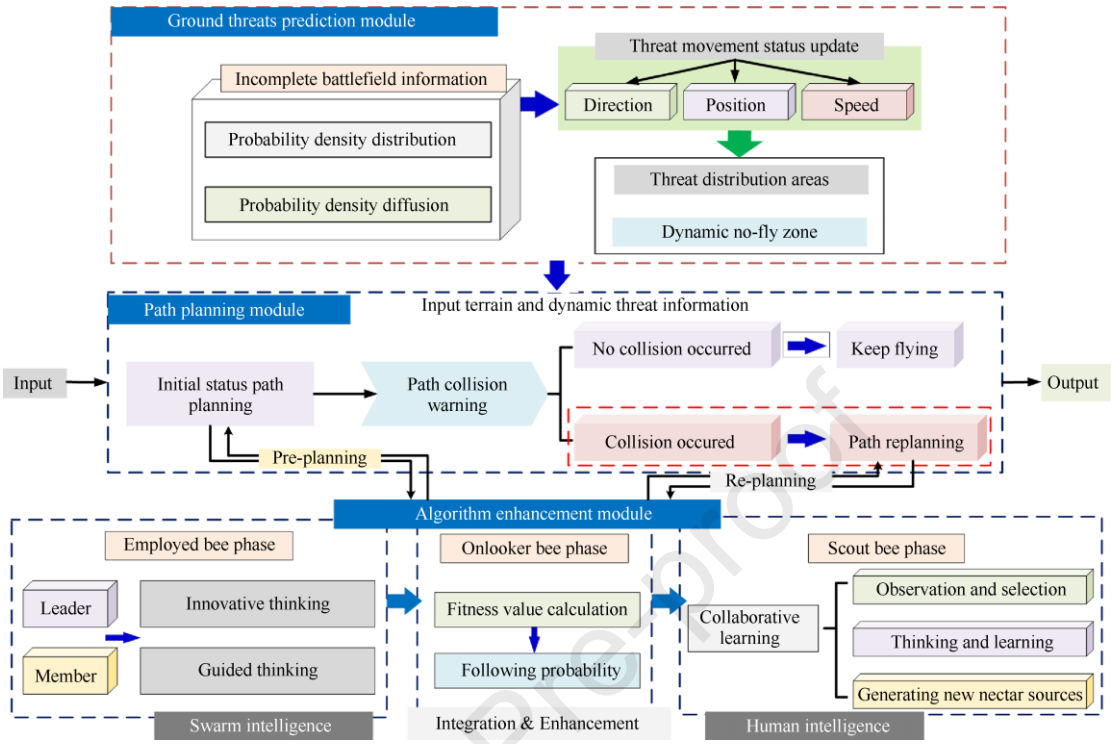


Fig. 2. Framework of ground threat prediction-based UAH path planning.

To solve the above problems, a path planning method based on ground threat prediction is proposed in this paper. As shown in Fig. 2, the proposed path planning method of the UAH can be divided into the following three modules.

➢ Ground threat prediction module

In this module, the probability of dynamic threat distribution is calculated based on non-complete threat information. The range of possible the ground threat is simulated and predicted, thus the UAH dynamic no-fly zone can be established.

➢ Path planning module

The module solves the UAH flight path by a hybrid enhanced ABC algorithm based on the collaborative thinking strategy. By embedding the collision warning mechanism into the loop of the optimization algorithm, it improves the efficiency of dynamic path planning.

➢ Algorithm enhancement module

In this module, the traditional ABC algorithm can be enhanced by introducing human intelligence. The optimization performance of ABC algorithm can be upgraded by introducing leader-member thinking mechanism and collaborative learning update strategy.

3. Dynamic ground threat prediction-based path planning

3.1. Modeling of UAH environment

As shown in Fig. 3, Path planning is an important part of UAH mission planning for finding an optimal flight path that meets constraints in urban, battlefield, and other environments.

First of all, the coordinate of *S* and *T* are set as (x_{ps}, y_{ps}, z_{ps}) and (x_{pt}, y_{pt}, z_{pt}). The *x*-axis range has been divided into *n*+1 equal portions according to the *S* and *T*. The perpendicular planes $(P_s, P_1, P_2, ..., P_n, P_T)$ of the *x*-axis are established according to the corresponding split points.

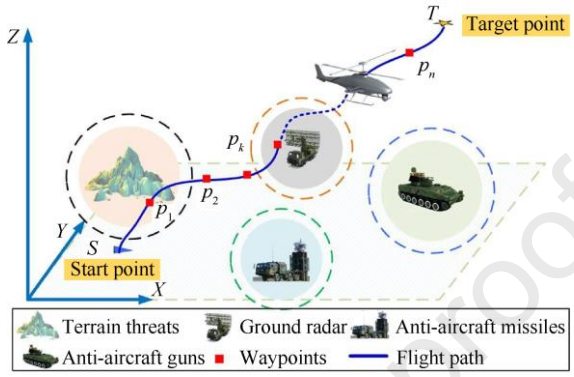


Fig. 3. Schematic of UAH battlefield model.

Put a discrete point in each vertical plane, and generate a collection of discrete points $C_{wp} = \{S, p_1(x_{p1}, y_{p1}, z_{p1}), p_2(x_{p2}, y_{p2}, z_{p2}), ..., p_n(x_{pn}, y_{pn}, z_{pn}), T\}$ [29]. In order to simplify the solution to the optimization problem, the optimization variable *E* is defined as

$$E = [y_1, y_2, ..., y_n, z_1, z_2 ..., z_n] \tag{1}$$

In this way, the path planning problem is transformed to 2*n* dimensional optimization problem.

3.2. Modeling of dynamic ground threats

In this section, a dynamic threat distribution probability model is designed based on non-complete information about the ground threat. Moreover, the dynamic ground threat distribution is used to build and update the dynamic no-fly zone of the UAH in real time.

3.2.1. Calculation of dynamic ground threat distribution probability

Direction is particularly important for dynamic ground threats movement decisions under non-complete information conditions [30]. Assume that the heading angle of the dynamic ground threat is chosen in the range $[0, 2\pi]$ and there is a constraint on the maximum turning angle of the ground threat. To simplify the calculation, the possible chosen heading angles of the dynamic ground threat at moment *t*+1 can be discretized as $d_{az}^{0}, d_{az}^{1}, d_{az}^{-1}, d_{az}^{2}, d_{az}^{-2}$.

As shown in Fig. 4, d_{az}^{0} represents the same maneuvering direction as the dynamic ground threat at the previous moment. $\alpha_i, i = (-2, -1, 1, 1)$ denote the angles between direction d_{az}^{i} and d_{az}^{0}. Assume that the probability of the threat manoeuvre direction at the current moment is the same as at the previous moment is *p*, that is, $P_{d_{az}^{0}(t+1)} = p$. Since $\alpha_{-2} = \alpha_2 = 2\alpha_{-1} = 2\alpha_1$, the probability of $d_{az}^{1}, d_{az}^{-1}, d_{az}^{2}, d_{az}^{-2}$ being selected at moment *t*+1 can be obtained by normalization.

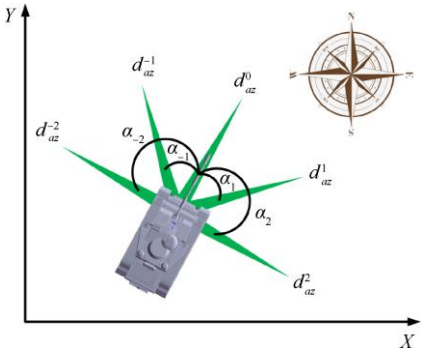


Fig. 4. Diagram of movement direction selection.

$$\begin{cases} P_{d_{az}^{1}(t+1)} = P_{d_{az}^{-1}(t+1)} = \dfrac{1-P_{d_{az}^{0}(t+1)}}{3} = \dfrac{1-p}{3} \\ P_{d_{az}^{2}(t+1)} = P_{d_{az}^{-2}(t+1)} = \dfrac{1-P_{d_{az}^{0}(t+1)}}{6} = \dfrac{1-p}{6} \end{cases} \tag{2}$$

The main factors affecting the ground threat direction prediction include the previous moment's movement speed, maneuver intention, and current terrain. The direction selection probability of the ground threat at each moment can be expressed as

$$p = \Pi_i^{ty} \cdot \Pi_v^{ty} \cdot \Pi_t^{ty} \tag{3}$$

where, Π_i^{ty} is the intention impact factor, Π_v^{ty} is the speed impact factor and Π_t^{ty} is the topographic impact factor. The detailed definitions of Π_i^{ty} , Π_v^{ty} and Π_t^{ty} are shown as follows:

(1) Intention impact factor: Π_i^{ty}

As shown in Fig. 5, we summarize the maneuvering intentions of the dynamic ground threat into the following two categories.

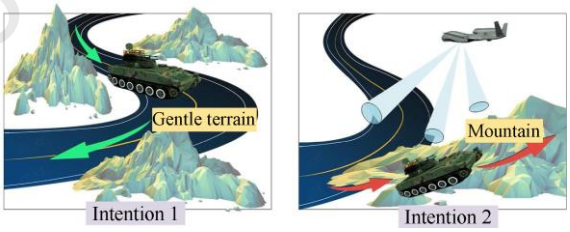


Fig. 5. Schematic of ground threat driving intentions.

➢ Intention 1: Manoeuvres towards a specific destination

Mobile air defence units choose to travel along roads or flat terrain when the dynamic ground threat has a clear driving target point.

➢ Intention 2: Evade reconnaissance or strikes

When the mobile air defense unit is evading our reconnaissance or strikes, the dynamic ground threat moves away from roads or gentle terrain.

Thus, different kinds of threats receive relative Π_i^{ty} according to different intentions, as shown in Table 1.

Table 1

Values of intention impact factor.

	Intention 1	Intention 2
Ground radar	C_1^r	C_2^r
Anti-aircraft missile	C_1^m	C_2^m
Anti-aircraft gun	C_1^g	C_2^g

where, C_1^r , C_1^m , C_1^g are the intent factors when Intention 1 is adopted for different types of dynamic ground threats, respectively. C_2^r , C_2^m , C_2^g are the intention factors for the dynamic ground threat when Intention 2 is adopted, respectively.

The initial locations of dynamic ground threats are assumed to be flat terrain. When the dynamic ground threat takes Intention 1, Π_c^{ty} is the expectation that the ground threat will remain unchanged in the direction of movement with a greater probability. When the dynamic ground threat takes Intention 2, the value of Π_c^{ty} should be relatively small. Therefore, the value of Π_c^{ty} should be constrained by

$$\Pi_c^{ty}=\{C_1^r, C_2^r, C_1^m, C_2^m, C_1^g, C_2^g\} \in (0,1) \tag{4}$$

$$C_1^r > C_2^r, C_1^m > C_2^m \text{ and } C_1^g > C_2^g \tag{5}$$

(2) Speed impact factor: Π_v^{ty}

If the dynamic ground threat moves faster at the previous moment, it can be inferred that the probability that it will change its manoeuvre direction at the next moment is lower [31]. Assuming the average speed of the threat as the judgement criterion, the speed impact factor for different threats can be described as

$$\Pi_v^r = \begin{cases} \tau_1^r, & v_{t-1}^r \geq \overline{v}_r \\ \tau_2^r, & v_{t-1}^r < \overline{v}_r \end{cases} \tag{6}$$

$$\Pi_v^m = \begin{cases} \tau_1^m, & v_{t-1}^m \geq \overline{v}_m \\ \tau_2^m, & v_{t-1}^m < \overline{v}_m \end{cases} \tag{7}$$

$$\Pi_v^g = \begin{cases} \tau_1^g, & v_{t-1}^g \geq \overline{v}_g \\ \tau_2^g, & v_{t-1}^g < \overline{v}_g \end{cases} \tag{8}$$

where, Π_v^r , Π_v^m , Π_v^g are the speed impact factors for ground radar, anti-aircraft missile and anti-aircraft gun respectively, v_{t-1}^r , v_{t-1}^m , v_{t-1}^g are the speeds of movement at the previous moment corresponding to the above ground threats, and $\overline{v}_r$, $\overline{v}_m$, $\overline{v}_g$ are the average speed of ground radar, anti-aircraft missile and anti-aircraft gun, respectively.

Due to the inertia of a moving object, the larger the speed impact factor, the greater the probability *p* that the dynamic ground threat will remain constant in its original direction. Therefore, the value of Π_v^{ty} should be constrained by

$$\begin{cases} 0<\tau_2^r<0.5<\tau_1^r<1, & \text{Ground radar} \\ 0<\tau_2^m<0.5<\tau_1^m<1, & \text{Anti-aircraft missile} \\ 0<\tau_2^g<0.5<\tau_1^g<1, & \text{Anti-aircraft gun} \end{cases} \tag{9}$$

In addition, the values of τ_1^r, τ_2^r, τ_1^m, τ_2^m, τ_1^g, τ_2^g can be adjusted according to the needs of the actual simulation experiment.

(3) Topographic impact factor: Π_t^{ty}

As shown in Fig. 6, assume that the battlefield terrain can be divided into three categories: hillside, valley and plain. Therefore, depending on the different intentions, the terrain impact factor Π_t^{ty} can be calculated by

$$\Pi_{t\to 1}^{ty}=\begin{cases} \varsigma^{ty}, & |\theta_k|=0 \\ (1-\varsigma^{ty})\cdot(1-\dfrac{|\theta_k|}{\sum_{i=1}^{n}|\theta_n|}), & 0<|\theta_k|<\max_s^{ty} \\ 0, & |\theta_k|>\max_s^{ty} \end{cases} \tag{10}$$

$$\Pi_{t\to 2}^{ty}=\begin{cases} 1-\varsigma^{ty}, & |\theta_k|=0 \\ \varsigma^{ty}\cdot(1-\dfrac{|\theta_k|}{\sum_{i=1}^{n}|\theta_n|}), & 0<|\theta_k|<\max_s^{ty} \\ 0, & |\theta_k|>\max_s^{ty} \end{cases} \tag{11}$$

where, $\Pi_{t\to 1}^{ty}$ and $\Pi_{t\to 2}^{ty}$ denote the terrain impact factors when Intention 1 and Intention 2 are used for ground threats, respectively. ς^{ty} is the topographic impact factor value of three threats, and $0.5<\varsigma^{ty}<1$. θ_k represents the angle of the *k*th terrain with respect to the horizontal plane. $\max_s^{ty}$ represents the maximum climbing angle for ground threats.

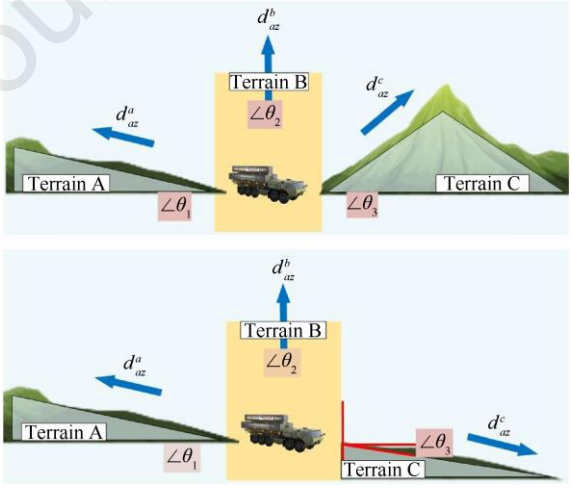


Fig. 6. Diagram of ground threats climb and descent.

3.2.2. Predictive update mechanism for the dynamic ground threat status

The state update process can be divided into updating the direction of movement, updating the speed of movement and updating the distribution location.

(1) Update of movement direction

The movement direction update mechanism based on the roulette strategy can be defined as

$$d_{az}(t+1)=\begin{cases} d_{az}^{-2}(t), & 0<\varphi\le P_{d_{az}^{-2}(t+1)} \\ d_{az}^{j}(t), & \sum_{k=-2}^{j+1} P_{d_{az}^{k}(t+1)}<\varphi\le \sum_{k=-2}^{j} P_{d_{az}^{k}(t+1)} \end{cases} \tag{12}$$

where, $d_{az}(t+1)$ is the direction chosen by the dynamic ground threat at the current moment in time, $P_{d_{az}^{-2}(t+1)}$ is the probability of being selected in direction d_{az}^{-2}, $\sum_{k=-2}^{j} P_{d_{az}^{k}(t+1)}$ denotes the cumulative probability of directional selection, *j* = -1, 0, 1, -2, and $\varphi=\mathrm{rand}(0,1)$.

(2) Update of movement speed

When the travel direction at moment *t*+1 is the same as at moment *t*, the travel speed of the dynamic ground threat can be understood to remain constant.

$$v_{ty}(t+1)=v_{ty}(t) \tag{13}$$

where, $v_{ty}(t)$ and $v_{ty}(t+1)$ denote the instantaneous velocity of the dynamic ground threat at the previous moment and the current moment, respectively.

When the direction of the dynamic ground threat changes, its travel speed is reduced according to the turning angle. At the end of a turning manoeuvre, if $v_{ty}(t)<\overline{v}_{ty}$ ($\overline{v}_{ty}$ is the average speed of the dynamic threat), the ground threat will perform an acceleration manoeuvre proportional to the difference in current speed. Therefore, the speed update strategy for the dynamic ground threat can be calculated by

$$v_{ty}(t+1)=v_{ty}(t)+\Gamma_{az}\cdot[\overline{v}_{ty}-v_{ty}(t)] \tag{14}$$

where, Γ_{az} is the speed adjustment parameter. The speed adjustment parameter Γ_{az} is influenced by terrain and intention, with different acceleration effects for different terrain under different intentions. Γ_{az} can be expressed as

$$\Gamma_{az}=\Xi_{i}^{ty}\cdot\Xi_{t}^{ty} \tag{15}$$

$$\begin{cases} \Xi_{i}^{ty}=\dfrac{t(1-\Pi_{i}^{ty})}{T} \\ \Xi_{t}^{ty}=\left[\pi-\arctan\left(\dfrac{h(t+1)-h(t)}{v_{ty}(t+1)\cdot T_{\mathrm{step}}}\right)\right]\Big/\pi \end{cases} \tag{16}$$

where, Ξ_{i}^{ty} and Ξ_{t}^{ty} are the intention factor and topography factor of the speed adjustment parameter Γ_{az}, *T* and *t* are the total simulation steps and the current update steps, *h*(*t*+1) represents the height of the terrain where the ground threat is located at the current moment, and T_{step} is the time step of the update.

Remark 1: As the number of updates increases, the cumulative amount of velocity also increases. Ξ_{i}^{ty} needs to grow larger over time to compensate for the previous cumulative velocity difference. In addition, the velocity

increment of Intention 2 should be higher than that of Intention 1, which is also consistent with the definition of Π_i^{ty}. For Ξ_t^{ty}, the gentler the terrain slope, the greater the speed increase. Conversely, the smaller the increase in velocity. This is in accordance with the fundamental laws of physics.

(3) Update of the distribution location

After determining the direction of manoeuvre and speed of travel of the dynamic ground threat, the coordinates of the threat's position in the map can be updated. The location update mechanism can be expressed as

$$\begin{cases} x_{t+1}^{ty} = x_t^{ty} + v_t^{ty} \cdot \cos\left(\frac{1}{4}\pi \cdot N\left(d_{az}(t+1)\right)\right) \cdot T_{\text{step}} \\ y_{t+1}^{ty} = x_t^{ty} + v_t^{ty} \cdot \sin\left(\frac{1}{4}\pi \cdot N\left(d_{az}(t+1)\right)\right) \cdot T_{\text{step}} \\ z_{t+1}^{ty} = z_{t+1}^{\text{terrain}} \end{cases} \tag{16}$$

where, x_{t+1}^{ty}, y_{t+1}^{ty}, z_{t+1}^{ty} are the coordinates of the dynamic ground threat position at the current moment, $z_{t+1}^{terrain}$ indicates the height of the terrain where the current dynamic ground threat is located, and $N\left(d_{az}(t+1)\right)$ represents the reference number of the direction selected by the dynamic ground threat at the current moment, $N\left(d_{az}(t+1)\right) \in \{0,1,2,3,4\}$.

3.2.3. Probabilistic diffusion mechanisms for the dynamic ground threat

Based on the laws of probability statistics, the distribution of the dynamic ground threat in a certain region basically obeys the Gaussian distribution [32].

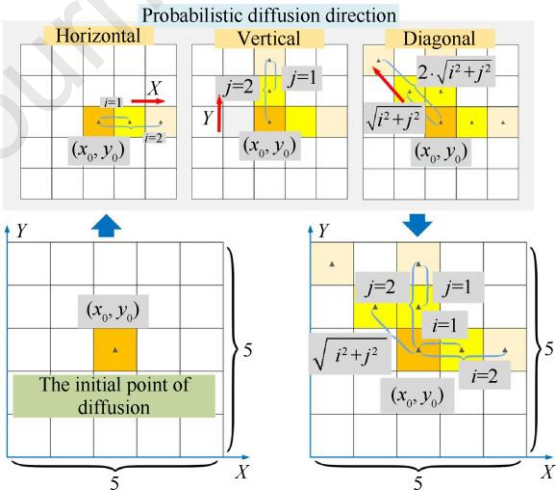


Fig. 7. Schematic of distribution probability diffusion.

As shown in Fig. 7, we take the endpoint coordinates of each prediction and spread its probability values through a Gaussian distribution, ultimately forming a probability distribution region rather than a single coordinate point. The specific diffusion can be calculated by

$$Kp(x_0+i, y_0+j) = Kp(x_0, y_0) \cdot \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{\left(l_{ij}-\mu\right)^2}{2\sigma^2}\right] \tag{17}$$

$$l_{ij}=\begin{cases} i, & \text{Lateral} \\ j, & \text{Vertical} \\ \sqrt{i^2+j^2}, & \text{Diagonal} \end{cases} \tag{18}$$

where, $Kp(x_0, y_0)$ denotes the probability of the current coordinate point, l_{ij} is the diffusion step size, *i*, *j* respectively represent the horizontal and vertical diffusion steps, and μ , σ respectively represent the mean and variance of the error.

The dynamic ground threat distribution probability calculation and diffusion process is shown in Fig. 8.

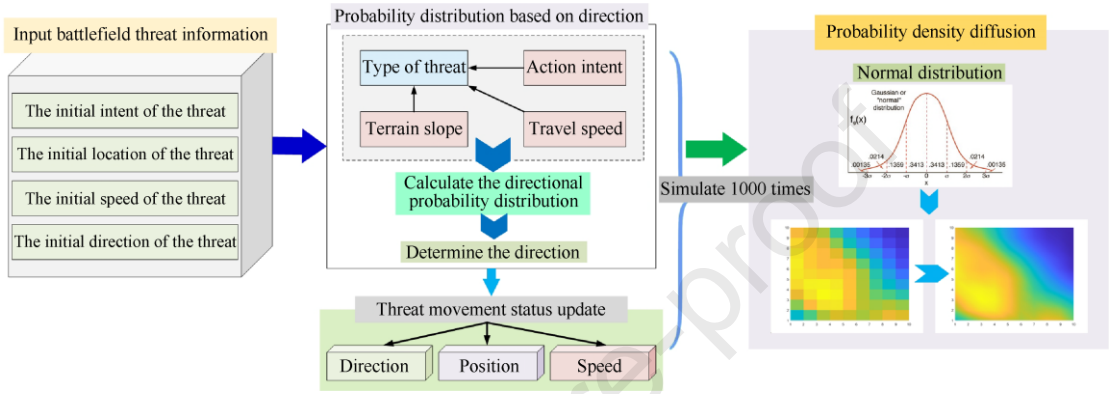


Fig. 8. Flowchart of the ground threat prediction algorithm.

3.2.4. Modelling of UAH dynamic no-fly zones

Areas where the dynamic ground threat may exist should be considered the no-fly zone to ensure the safety of UAH flights. As shown in Fig. 9, the different colours in the heat map indicate differences in the probability of a threat occurring at that location.

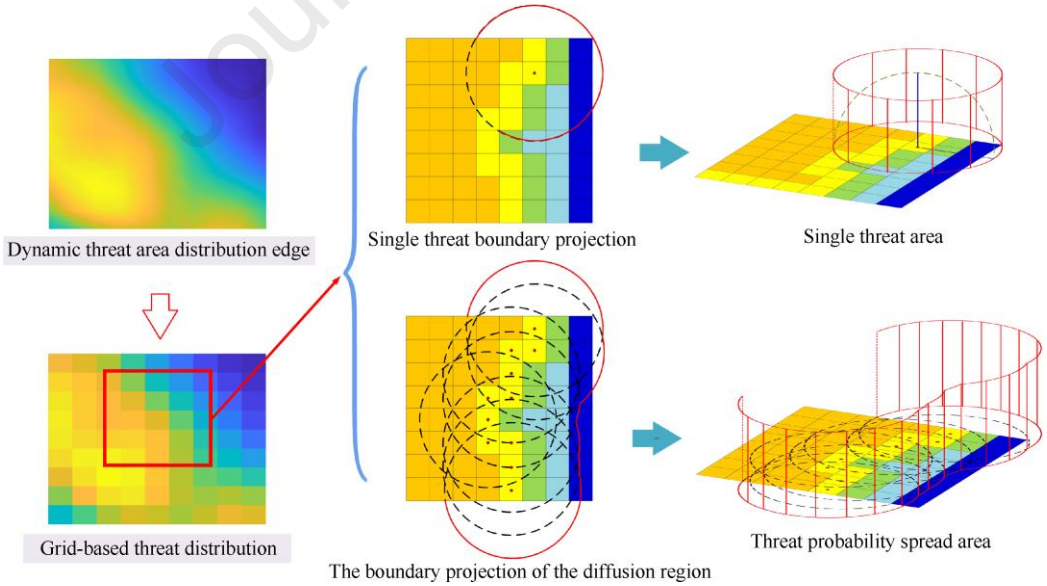


Fig. 9. Dynamic no-fly zone modeling.

The dynamic no-fly zone can be determined by establishing the boundaries of the probability spread area. It is important to note that the no-fly zone in this area is dynamically irregular and can change as the area of spread changes. The dynamic no-fly zone flow chart is shown in Fig. 10.

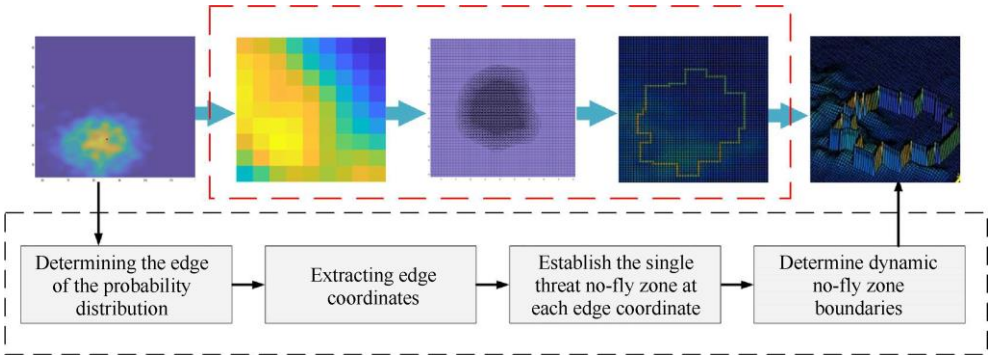


Fig. 10. Flowchart of dynamic no-fly zone modeling.

3.3. Objective function and performance constraints

In this paper, the cost function of a flight path is described as the sum of three optimization criteria and three penalty terms. Thus, the objective function for predictive path planning can be expressed as [33]

$$obj(x) = \min J = \sum_{i=1}^{3} f_i^{oc} + \sum_{i=1}^{3} g_i^{pt} \tag{19}$$

where, i = 1, 2, 3, f_i^{oc} is the optimization criterion to evaluate the flight path, and g_i^{pt} is the penalty function to ensure he flight path meets the UAH performance constraints.

In addition, the UAH cannot fly beyond the map area boundary.

$$p_k(x_{pk}, y_{pk}, z_{pk}) \in \Omega\left(x_{pn}, y_{pn}, z_{pn}\right) \rightarrow \begin{cases} L_{\min} \le x_{pk} \le L_{\max} \\ W_{\min} \le y_{pk} \le W_{\max} \\ H_{\min} \le z_{pk} \le H_{\max} \end{cases} \tag{20}$$

where, k = 1, 2,..., n, Ω is the solution space, $[L_{\min}, L_{\max}]$, $[W_{\min}, W_{\max}]$ and $[H_{\min}, H_{\max}]$ are the boundaries of UAH flight area, respectively.

The optimization criteria for the flight path of the UAH are as follows:

- Cost of the flight path length f_1^{oc}

A shorter flight path means less flight time and fuel consumption. f_1^{oc} can be described as [33]

$$f_1^{oc} = \sum_{k=1}^{n} l_k \tag{21}$$

With

$$l_k = \sqrt{\Delta x_p^2 + \Delta y_p^2 + \Delta z_p^2} \tag{22}$$

$$\begin{cases} \Delta x_p = x_{pk} - x_{p(k-1)} \\ \Delta y_p = y_{pk} - y_{p(k-1)} \\ \Delta z_p = z_{pk} - z_{p(k-1)} \end{cases} \tag{23}$$

where, k = 1, 2,..., n, n is the number of the waypoints.

- Cost of the ground threats f_2^{oc}

Considering the three threat categories of ground radar, anti-aircraft missiles and anti-aircraft gun, f_2^{oc} can be described as [17]

$$f_2^{oc} = \sum_{k=1}^{n} (c_k^{rt} + c_k^{mt} + c_k^{gt}) \tag{24}$$

With

$$c_k^{rt} = \begin{cases} \dfrac{cv_{rt}}{d_{b\to rt}^k}, & d_{b\to rt}^k > R_{rt}^{\max} \\ cv_{rt} + \dfrac{cv_{rt}}{1+\kappa_1((d_{b\to rt}^k)^2/\mathrm{RCS})^{\kappa_2}}, & d_{b\to rt}^k \le R_{rt}^{\max} \end{cases} \tag{25}$$

$$c_k^{mt} = \begin{cases} \dfrac{cv_{mt}}{d_{b\to mt}^k}, & d_{b\to mt}^k > R_{mt}^{\max} \\ cv_{mt} + \dfrac{cv_{mt}\cdot(R_{mt}^{\max})^4}{(d_{o\to mt}^k)^4 + (R_{mt}^{\max})^4}, & d_{b\to mt}^k \le R_{mt}^{\max} \end{cases} \tag{26}$$

$$c_k^{gt} = \begin{cases} \dfrac{cv_{gt}}{d_{b\to gt}^k}, & d_{b\to gt}^k > R_{gt}^{\max} \\ cv_{gt}\cdot(1+\dfrac{1}{d_{b\to gt}^k}), & d_{b\to gt}^k \le R_{gt}^{\max} \end{cases} \tag{27}$$

where, c_k^{rt}, c_k^{mt} and c_k^{gt} denote the threat cost of the ground radar, anti-aircraft missile and anti-aircraft gun at P_k, respectively. cv_{rt}, cv_{mt} and cv_{gt} are the basic threat parameters of these dynamic threats. RCS is the radar cross-section, κ_1 and κ_2 are the inherent parameters of the radar. $R_{rt}^{\max}$, $R_{mt}^{\max}$ and $R_{gt}^{\max}$ denote the maximum action distance of the ground radar, anti-aircraft missile and anti-aircraft gun, respectively. $d_{b\to rt}^k$, $d_{o\to mt}^k$, and $d_{b\to gt}^k$ denote the vertical distance of P_k from the centreline of the ground radar, anti-aircraft missile and anti-aircraft gun, respectively. $d_{o\to mt}^k$ is the distance of P_k from the centre of the anti-aircraft missiles.

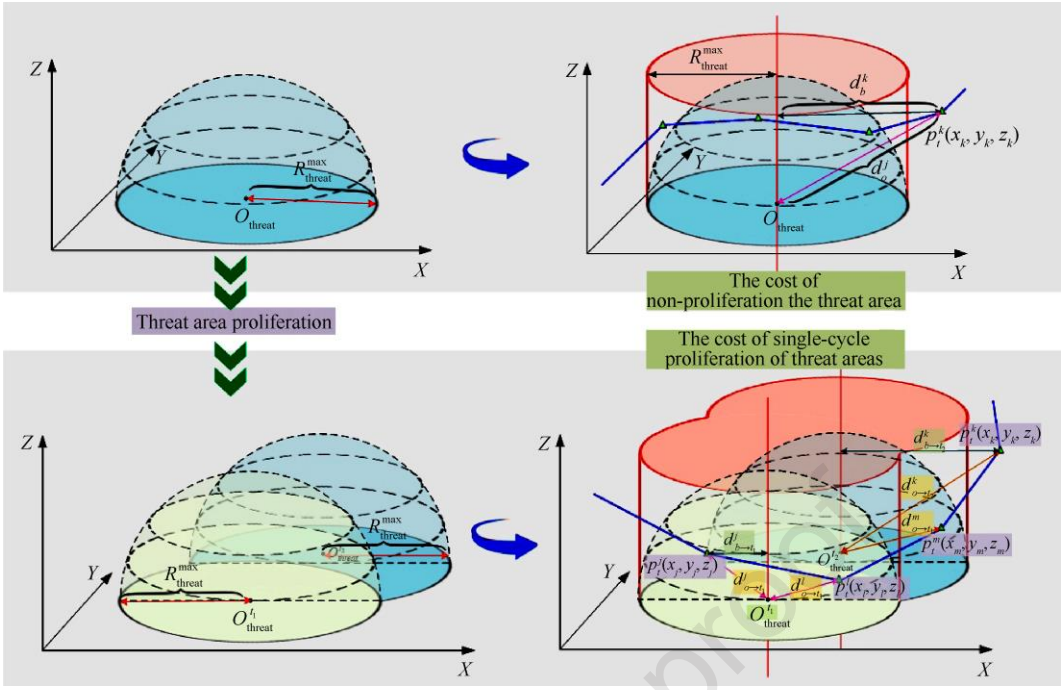


Fig. 11. Schematic diagram of waypoint threat cost.

Remark 2: As shown in Fig. 11, $d_{b\to ty}^{k}$ represents the vertical distance of waypoint P_k from the centre line of the nearest threat spreading point. The criterion for selecting the nearest threat spreading point is to determine whether the distance $d_{o\to ty}^{k}$ between the waypoint P_k and the neighbouring spreading point is minimal. The cost of threats in an irregular no-fly zone can be calculated using the above rules.

- Cost of the flight altitude f_3^{oc}

On the premise of ensuring the safe flight of UAH, a lower flying altitude can improve its concealment performance. Therefore, f_3^{oc} can be calculated by [33]

$$f_3^{oc} = \sum_{k=1}^{n} c_k^{\text{con}} \tag{28}$$

With

$$c_k^{\text{con}} = \begin{cases} H_{\text{con}}, & z_{pk} - H_t(P_k) - H_{\text{safe}} \le H_{\text{con}} \\ z_{pk} - H_t(P_k), & z_{pk} - H_t(P_k) - H_{\text{safe}} > H_{\text{con}} \end{cases} \tag{29}$$

where, z_{pk} is the *z*-axis coordinate of waypoint P_k, $H_t(P_k)$ is the terrain altitude at P_k, H_{safe} is the minimal safe flight height, and H_{con} is the threshold for concealed height.

The constraints for the flight path of the UAH are as follows:

- Constraint of the yawing angle g_1^{pt}

The penalty function g_1^{pt} is the sum of those waypoints that can not satisfy the UAH yawing angle constraint, which can be calculated by [34]

$$g_1^{pt} = \sum_{k=1}^{n} g_{pk}^{\text{yaw}} \tag{30}$$

With

$$g_{pk}^{\text{yaw}} = \begin{cases} \lambda_{\text{yaw}}, & |\varphi_k| > \varphi_k^{\max} \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

$$\varphi_k^{\max} = g \cdot \frac{n^{\max}}{v^2} \cdot \sqrt{\left(x_{pk} - x_{p(k-1)}\right)^2 + \left(y_{pk} - y_{p(k-1)}\right)^2} \tag{32}$$

where, φ_k is the flight path yawing angle of the waypoint P_k, $\varphi_k^{\max}$ is the maximum flight path yawing angle of the UAH, λ_{yaw} is the penalty coefficient, *g* is the gravitational acceleration, $n^{\max}$ is the maximum lateral overload and *v* is the flight velocity of the UAH.

- Constraint of the pitching angle g_2^{pt}

The slope α_k of the sub-track can not scale out the maximum pitching angle $\alpha_k^{\max}$. g_2^{pt} can be calculated by [34]

$$g_2^{pt} = \sum_{k=1}^{n} g_{pk}^{\text{pitch}} \tag{33}$$

With

$$g_{pk}^{\text{pitch}} = \begin{cases} \lambda_{\text{pitch}}, & |\alpha_k| > \alpha_k^{\max} \\ 0, & \text{otherwise} \end{cases} \tag{34}$$

where, λ_{pitch} is the penalty coefficient. α_k at the waypoint P_k can be calculated by

$$\alpha_k = \frac{z_{pk} - z_{p(k-1)}}{\sqrt{(x_{pk} - x_{p(k-1)})^2 + (y_{pk} - y_{p(k-1)})^2}} \tag{35}$$

- Constraint of collision g_3^{pt}

The penalty function g_3^{pt} is introduced to penalize the nonfeasible paths that are likely to crash into mountains.

$$g_3^{pt} = \sum_{k=1}^{n} g_{pk}^{\text{ter}} \tag{36}$$

With

$$g_{pk}^{\text{ter}} = \begin{cases} \lambda_{\text{collision}}, & z_{pk} < H_t(P_k) + H_{\text{safe}} \\ 0, & z_{pk} \geq H_t(P_k) + H_{\text{safe}} \end{cases} \tag{37}$$

where, $\lambda_{\text{collision}}$ is the penalty coefficient and H_{safe} is the minimal safe flight height.

4. Hybrid enhanced ABC algorithm based on collaborative thinking strategy

The ABC algorithm has been widely used in the field of path planning due to its excellent optimization capability [35]. However, as the path planning problem becomes more and more complex, the traditional ABC algorithm can no longer meet the performance requirements of current path planning algorithms.

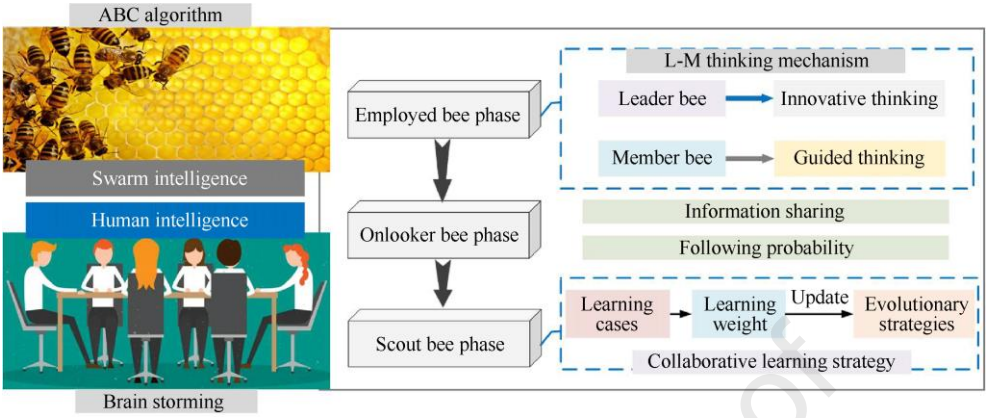


Fig. 12. Schematic diagram of hybrid enhanced swarm intelligence.

In this paper, a hybrid enhanced ABC algorithm is designed based on collaborative thinking strategy (CTS). As shown in Fig. 12, by introducing the leader-member thinking mechanism and the collaborative learning strategy, the CTS-ABC algorithm can effectively address the shortcomings of the traditional ABC algorithm, significantly improve the optimization performance of the ABC algorithm.

4.1. Initialization

All the vectors of the population of food sources $\vec{x}_i$ are initialized $\vec{x}_i (i \in 1,2,...,SN)$ by scout bees and control parameters are set. One nectar source represents a feasible solution to the path planning problem. Each nectar source $\vec{x}_i$ has *D* parameters to be optimized $(x_i^j, j = 1,2,...,D)$. The initial source locations are randomly initialized by [36]

$$x_i^j = x_i^{\text{lb}} + \text{rand}(0,1)(x_i^{\text{ub}} - x_i^{\text{lb}}) \tag{38}$$

where, x_i^{ub} and x_i^{lb} are the upper and lower limits of the parameter x_i^j, respectively.

The fitness value $\text{fit}(x)$ of x_i^j can be calculated by [36]

$$\text{fit}(x) = \begin{cases} 1/(1+\text{obj}(x)), & \text{if obj}(x) \geq 0 \\ 1+\text{abs}(\text{obj}(x)), & \text{othewrise} \end{cases} \tag{39}$$

4.2. Employed bee phase based on leader-member thinking mechanism

For the traditional ABC algorithm, an employed bee searches a new food position $\vec{v}_i$ using its current food position $\vec{x}_i$ as follows [36]:

$$v_i^j = x_i^j + \phi(x_i^j - x_k^j) \tag{40}$$

where, $i,k \in \{1,2,...,SN\}$, $j \in \{1,2,...,D\}$, $i \neq k$, and $\phi = \text{rand}(-1,1)$.

The necessity of mining near the current optimal nectar source is often higher than that of other nectar

sources. The optimal food source should be searched in a different way from other nectar sources in the same iteration cycle.

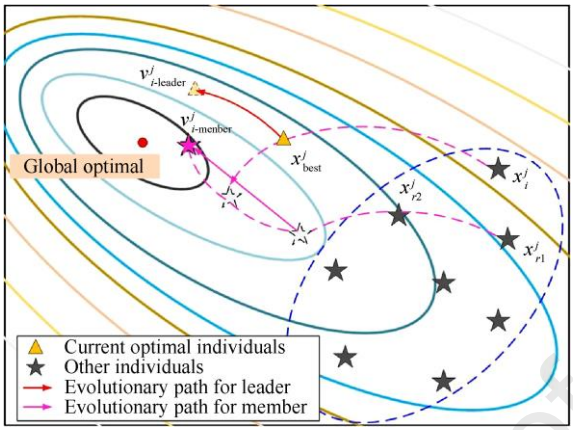


Fig. 13. The schematic of leader-member thinking mechanism.

As shown in Fig. 13, the best individual in the current population is the leader and uses its own excellent position to evolve using innovative thinking. The rest of the individuals become members and adopt a guided thinking approach to position updating.

(1) Search strategy based on innovative thinking

This search strategy is suitable for the optimal individual in the current iteration cycle. In order to fully explore the development potential for the highest quality individual, the search strategy based on innovative thinking is defined as

$$v_{i-\text{leader}}^{j} = \omega x_{\text{best}}^{j} + \phi(1-\omega)(x_{\text{best}}^{j} - x_{k}^{j}) \tag{41}$$

With

$$\omega = \frac{\text{iter}}{\text{Maxcycle}} \tag{42}$$

where, x_{best}^{j} represents the current optimal individual, ω is the adaptive adjustment parameter, iter is the number of current iterations, Maxcycle represents the maximum number of iterations of the algorithm, $k \in \{1,2,...,SN\}$, and $j \in \{1,2,...,D\}$.

(2) Search strategy based on guided thinking

The experience of the leader often plays a positive guiding role for other employed bees. This kind of guidance will make the ABC algorithm converge quickly to the optimal solution, thereby avoiding too many invalid searches. This search strategy is defined as

$$v_{i-\text{member}}^{j} = (1-\omega)x_{i}^{j} + \omega[\tau_1\phi(x_{\text{best}}^{j} - x_{i}^{j}) + \tau_2\phi(x_{r1}^{j} - x_{r2}^{j})] \tag{43}$$

where, $i \in \{1,2,...,SN-1\}$, τ_1 and τ_2 are the learning factors, x_{r1}^{j} and x_{r2}^{j} are randomly selected

individuals from the population.

The specific steps of the employed bee phase are shown in Fig. 14.

Algorithm 1 Employed bee phase

Require: Adaptive adjustment parameter: ω; Learning factors: τ_1, τ_2;
1: Calculate the fitness values of all individuals in the population
2: *Neighborhood search for current optimal individual*
3: $x_{best}^j = \max fit(x)$
4: **for** $j = 1$ to D **do**
5: $v_{i-leader}^j = \omega x_{best}^j + \phi(1-\omega)(x_{best}^j - x_k^j)$
6: **end for**
7: *Neighborhood search for other individuals*
8: **for** $i = 1$ to $SN - 1$ **do**
9: **for** $j = 1$ to D **do**
10: $v_{i-member}^j = (1-\omega)x_i^j + \omega[\tau_1\phi(x_{best}^j - x_i^j) + \tau_2\phi(x_{r1}^j - x_{r2}^j)]$
11: **end for**
12: **end for**
13: Update food sources by greedy criterion

Fig. 14. Specific steps of the employed bee phase.

4.3. Onlooker bee phase

After completing the employed bees phase, a probability value is computed as follow [36]:

$$p(x) = \frac{\mathrm{fit}(x)}{\sum_{i=1}^{SN} \mathrm{fit}(x)} \tag{44}$$

4.4. Scout bee phase based on collaborative learning strategy

When a nectar source has been explored several times and still not renewed, this employed bee will be transformed into the scout bee to explore a new nectar source by [37]. It is worth mentioning that the local optimum solution is updated by randomly selecting a new nectar source in its vicinity. As shown in Fig. 15, if the evolutionary direction of the local optimal solution does not improve the quality of this individual, the utilization value of this individual will be reduced and this situation may continue [38-40].

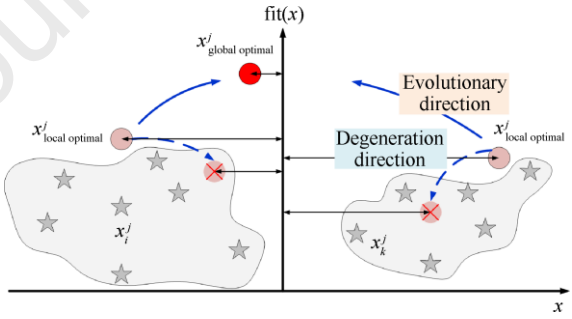


Fig. 15. The schematic of local optimal solution evolution/degeneration.

In order to reduce the negative impact of randomness on the scout bee phase, a collaborative learning based evolutionary strategy is proposed. As shown in Fig. 16, during the scout bee phase, the locally optimal individuals generally go through three processes to cooperatively learn other individuals' knowledge before proposing their own ideas.

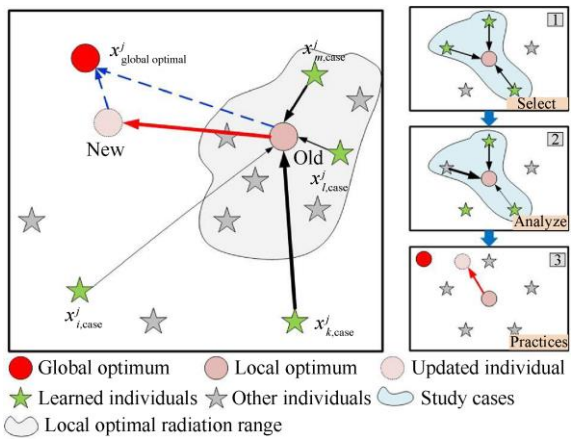


Fig. 16. The schematic of collaborative learning process.

The core idea of collaborative learning search strategy is to adopt a new method to simulate the process of human swarm intelligence generating new ideas [41]. Taking individual x_i^j as an example, the specific steps for its adoption of collaborative learning strategies are as follows:

(1) Local optimal individual determination

The collaborative learning strategy is only applicable to locally optimal individuals. This judgment criterion of the locally optimal individuals can be described as

$$\text{Num}_{x_i^j}^{\text{iter}} \geq \text{limit} \tag{45}$$

where, limit is the maximum search threshold for nectar sources, $\text{Num}_{x_i^j}^{\text{iter}}$ represents the number of developments of x_i^j by employed bees. For individual x_i^j, if it satisfies the above conditions, then $x_i^j \to x_{l-o,i}^j$.

(2) Observation and selection

For each candidate solution, randomly select m other candidate solutions $x_1^j, x_2^j, \ldots, x_m^j$ in the current iteration cycle as the collaborative learning cases X_{case}.

$$X_{\text{case}} = \{x_1^j, x_2^j, \ldots, x_m^j\} \tag{46}$$

(3) Analysis and reflection

The degree of influence on $x_{l-o,i}^j$ will vary due to the different levels of individuals in the learning case. For each collaborative learning case $x_k^j \in X_{\text{case}}$, the collaborative learning weight ζ is calculated by normalizing the fitness values $\text{fit}(x_1^j), \text{fit}(x_2^j), \cdots, \text{fit}(x_m^j)$ of the collaborative learning cases. The collaborative learning weigh ζ is calculated by

$$\zeta_i = \left(\text{fit}(x_k^j) - \mu\right) / \sigma \tag{47}$$

where, $k \in m$, μ and σ are the mean and standard deviation of the fitness values $\text{fit}(x_1^j)$, $\text{fit}(x_2^j)$,..., $\text{fit}(x_m^j)$, respectively.

ζ describes the degree of influence of different learning objects on the evolution of $x_{l-o,i}^{j}$. The individual $x_{l-o,i}^{j}$ can be updated to the transition individual $x_{vir,i}^{j}$ using ζ, which will serve as an excess region for the convergence of the local optimal solution to the global optimal solution.

(4) Learning and practice

The transition individual $x_{vir,i}^{j}$ can be seen as a `springboard' for the evolution of the locally optimal individual $x_{l-o,i}^{j}$ to the vicinity of the global optimal solution. The transitional individual $x_{\mathrm{vir},i}^{j}$ can be calculated by

$$x_{\mathrm{vir},i}^{j}=\sum_{i=1,j=1}^{m}\zeta_i x_{l-o,i}^{j} \tag{48}$$

where, ζ_i is the learning weight for the locally optimal individual $x_{l-o,i}^{j}$.

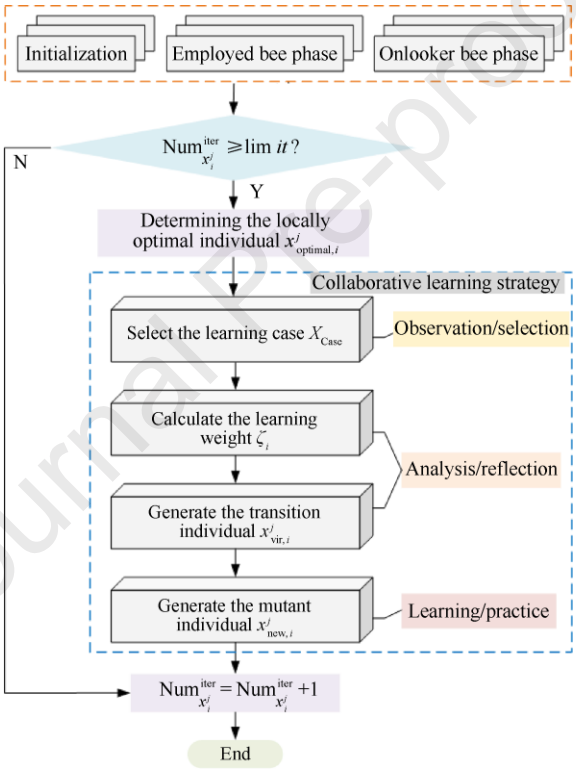


Fig. 17. Flowchart of the collaborative learning strategy.

In order to make the ABC algorithm more efficient in jumping out of the influence of the local optimal solution, the scout bee uses the following formula to generate a new nectar source x_{new}^{j} to replace the local optimal solution x_{l-o}^{j} through the collaborative learning.

$$x_{\mathrm{new},i}^{j}=x_{\mathrm{vir},i}^{j}+\phi(x_{\mathrm{best}}^{j}-x_{\mathrm{vir},i}^{j}) \tag{49}$$

where, x_{best}^{j} is denoted as the optimal individual of the current population except for the local optimal solution.

Fig. 17 is the flowchart of the collaborative learning strategy, and Fig. 18 is the specific steps of the CTS-ABC

algorithm.

4.5. Computing complexity analysis

The computing complexity of the proposed method includes the dynamic threat prediction algorithm computing complexity and the CTS-ABC algorithm computing complexity.

(1) The dynamic threat prediction algorithm

The dynamic threat prediction algorithm can be viewed as an inference process that runs with two loops nested in it. The computing complexity of the dynamic threat prediction algorithm can be calculated by

$$C^d = O(N^2 \cdot D) \tag{50}$$

where, $O(\cdot)$ is the asymptotic time complexity.

(2) The CTS-ABC algorithm

The CTS-ABC algorithm consists of the initialization phase and the optimization phase [17]. The computing complexity of these two components can be calculated separately by

$$C_1^c = \max[O(1), O(N \cdot D)] = O(N \cdot D) \tag{51}$$

$$\begin{aligned} C_2^c &= \max[O(\lg N \cdot D), O(N \cdot D), O(N \cdot D), O(N^2 \cdot D), O(n \lg N \cdot D)] \\ &= O(N^2 \cdot D) \end{aligned} \tag{52}$$

Thus, we can get the maximum computing complexity of the CTS-ABC algorithm.

$$\begin{aligned} C^c &= \max[C_1^c(N), C_2^c(N)] = \max[O(N \cdot D), O(N^2 \cdot D)] \\ &= O(N^2 \cdot D) \end{aligned} \tag{53}$$

Similarly, the maximum computing complexity of the traditional ABC algorithm can be described as [33]

$$\begin{aligned} C^{\mathrm{ABC}} &= \max[O(1), O(N \cdot D), O(N^2 \cdot D), O(n \lg N \cdot D)] \\ &= O(N^2 \cdot D) \end{aligned} \tag{54}$$

The analysis results show that the computing complexity of the dynamic threat prediction algorithm and the CTS-ABC algorithm are relatively low. The efficiency of solving the UAH path planning problem can still be guaranteed.

Algorithm 2 CTS-ABC algorithm pseudo-code

Require: Population: SN; Dimension: D; Iterations: $Maxcycle$; Search threshold: $limit$

Randomly produce the population of the SN individuals by $x_i^j = x_i^{lb} + rand(0,1)(x_i^{ub} - x_i^{lb})$
Evaluate all the individuals and calculate fitness values $fit(x_i^j)$
Initialize the number of failed trials l_{tri}
while $t < Maxcycle$ **do**
  $//Employed\ bee\ phase:$
  for $i = 1, 2, ..., SN$ **do**
    The best individual x_{best}^j
    $v_{i-leader}^j = \omega x_{best}^j + \phi(1-\omega)(x_{best}^j - x_k^j)$
    The other individuals x_i^j
    $v_{i-member}^j = (1-\omega)x_i^j + \omega[\tau_1\phi(x_{best}^j - x_i^j) + \tau_2\phi(x_{r1}^j - x_{r2}^j)]$
    if $fit(v_i^j) > fit(x_i^j)$ **then**
      $x_i^j \leftarrow v_i^j, l_{tri} \leftarrow 0$
    else
      $l_{tri} \leftarrow l_{tri} + 1$
    end if
  end for
  $//Onlooker\ bee\ phase:$
  for $i = 1, 2, ..., SN$ **do**
    Select x_i^j with the probability $p(x) = \frac{fit(x)}{\sum_{i=1}^{SN} fit(x)}$
    Select a random food source and conduct a neighborhood search
    The best individual x_{best}^j:
    $v_{i-leader}^j = \omega x_{best}^j + \phi(1-\omega)(x_{best}^j - x_k^j)$
    The other individuals x_i^j:
    $v_{i-member}^j = (1-\omega)x_i^j + \omega[\tau_1\phi(x_{best}^j - x_i^j) + \tau_2\phi(x_{r1}^j - x_{r2}^j)]$
    if $fit(v_i^j) > fit(x_i^j)$ **then**
      $x_i^j \leftarrow v_i^j, l_{tri} \leftarrow 0$
    else
      $l_{tri} \leftarrow l_{tri} + 1$
    end if
  end for
  $//Scout\ bee\ phase:$
  for $i = 1, 2, ..., SN$ **do**
    if $l_{tri} \geq limit$ **then**
      Randomly select m candidate solution as the collaborative learning cases X_{Case}
      for each $x_k^j \in X_{Case}$ **do**
        $\zeta_i = \left(fit(x_k^j) - \mu\right)/\sigma$
        $x_{vir,i}^j = \sum_{i=1,j=1}^{m} \zeta_i x_{l-o,i}^j$
      end for
      Generate the new candidate solution by: $x_{new,i}^j = x_{vir,i}^j + \phi(x_{best}^j - x_{vir,i}^j)$
      $x_i^j \leftarrow x_{new}^j$
      Let $l_{tri} = 0$
    end if
  end for
  Record the optimal solution achieved so far
  if $t = Maxcycle$ **then**
    Output the optimal solution X
  else
    $t = t + 1$
  end if
end while

Fig. 18. Specific steps of the CTS-ABC algorithm.

5. Simulation results

In order to verify the efficiency of the proposed model and algorithm for the dynamic threat path planning problem, we have introduced the two scenarios with the different level (the higher level has the more complex threats and no-flight zones). The detailed information of scenarios is shown in Table 2, main parameters of dynamic threats distribution model are shown in Table 3, and main parameters of the CTS-ABC algorithm are shown in Table 4. Furthermore, to make the experiment results more objective, we make the following assumptions for each simulation: 1) Assume that the initial location of all dynamic threats is on the flat terrain; 2) Assume that the UAHs in all scenarios fly at a constant speed; 3) The threat prediction termination time is the moment when the UAH simulation reaches the end point.

Table 2

Main parameters of scenarios.

Terrain	Map 1	Map 2
Map size/km	200*200*1000	400*400*1000
Start point	[16,17,34]	[18,72,127]
Target point	[200,203,26]	[381,363,46]
Initial location	[21,117,42]	[58,220,418]
	[80,40,33]	[80,40,393]
	[157,135,26]	[157,190,45]
	[180,30,17]	[164,288,144]
	-	[229,227,55]
	-	[255,622,15]
Threat radius/km	40/35/35/30	35/35/30/40/35/40

Table 3

Main parameters of dynamic threats distribution model.

Parameters	Types	Values
Intention impact factor Π_c^{ty}	C_1^r / C_1^m / C_1^g	0.7/0.65/0.6
	C_2^r / C_2^m / C_2^g	0.5/0.45/0.4
Target point Π_v^{ty}	τ_1^r / τ_1^m / τ_1^g	0.8/0.7/0.6
	τ_2^r / τ_2^m / τ_2^g	0.45/0.4/0.35
Topographic impact factor Π_t^{ty} /(°)	ζ^r / ζ^m / ζ^g	0.8/0.7/0.6
	$\max_{\text{slope}}^r$ / $\max_{\text{slope}}^m$ / $\max_{\text{slope}}^g$	30/40/50
Average speed $\overline{v}_{ty}$ /(km·h^{-1})	$\overline{v}_r$ / $\overline{v}_m$ / $\overline{v}_g$	25/30/35

Table 4

Main parameters of scenarios.

Main parameter	Values
Swarm size *SN*	50
Predefined threshold *Limit*	5
Lower/upper boundary x_i^{lb} / x_i^{ub}	repmat((ub-lb),*SN*)/repmat(lb,*SN*)
Maximum iterations Maxcycle	40/200
Learning factors τ_1 / τ_2	0.7/0.4

5.1. Scenario 1

For the first scenario, Fig. 19 shows the diffusion process of dynamic threats in Map 1. Over time, the distribution of dynamic threats expands and the probability of distribution decreases. The diffusion process lasted a total of 16 s, which is the time taken for the UAH to reach the target point during the planning process. The colour change in the area of the probability distribution of dynamic threats in the graph illustrates this process.

Fig. 20 describes the final shape of the dynamic no-fly zone. In order to reduce the computer rendering time, the boundaries of the dynamic no-fly zone are drawn to describe the extent and size of the entire no-fly zone. Due to the relatively flat terrain, the dynamic no-fly zone is relatively concentrated in scope.

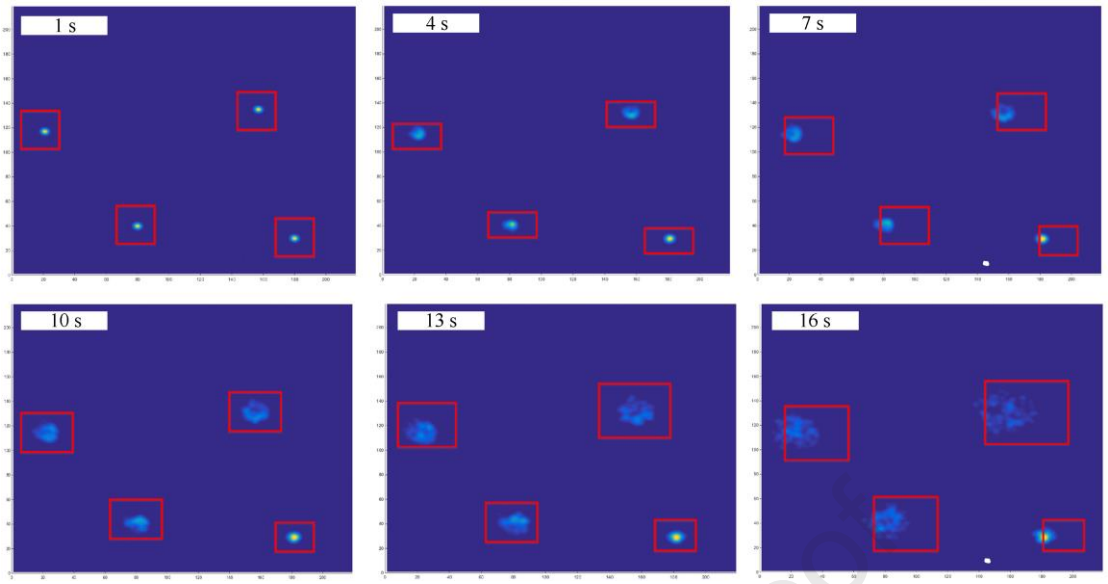


Fig. 19. The diffusion process of dynamic threat in Map 1.

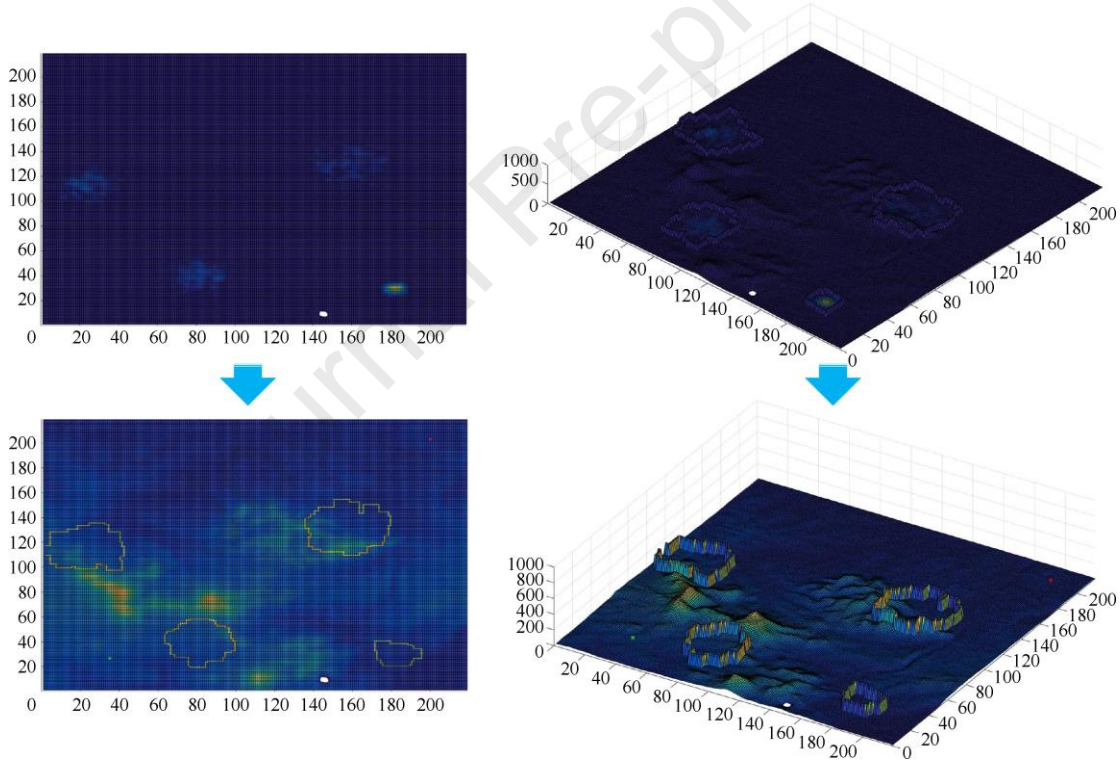


Fig. 20. The final shape of the dynamic no-fly zone Map 1.

Fig. 21 shows the process of the UAH path planning based on the CTS-ABC algorithm in Map 1. The red path indicates the pre-planned flight path and the yellow path indicates the actual flight path of the UAH. The green waypoints in the path are collision warning points, i.e. location nodes for path replanning. The red paths in Fig. 21(5), Fig. 21(7) and Fig. 21(8) indicate the replanned UAH flight path. The dynamic no-fly zone changes over time during the path planning process. The yellow path in Fig. 21(12) indicates the actual path flown by the UAH. It is important to note that although part of this path crosses the no-fly zone, the UAH has flown out of the area before the arrival of this no-fly zone, so this path does not affect the safety of the UAH.

Fig. 22 shows the final path planning results of the CTS-ABC algorithm in Map 1. The CTS-ABC algorithm can plan a smooth predicted flight path for UAH based on the incomplete information of the battlefield. The predicted path enables the UAH to avoid ground threats that could affect flight safety, significantly reducing the computational stress on onboard computers and sensors.

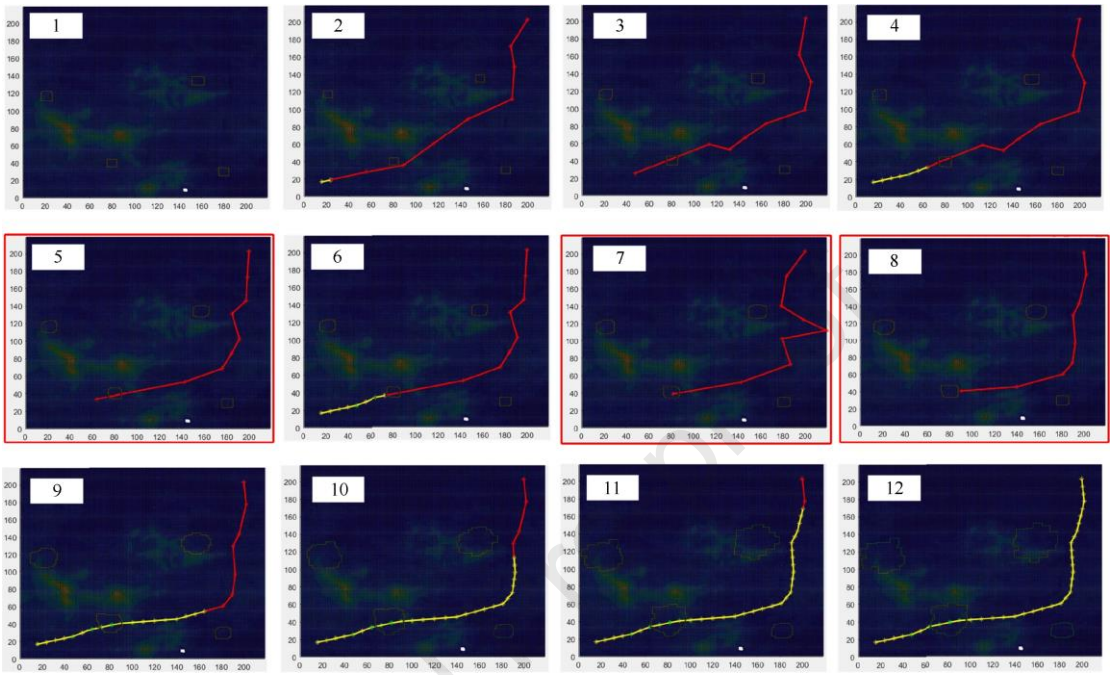


Fig. 21. The final shape of the dynamic no-fly zone Map 1.

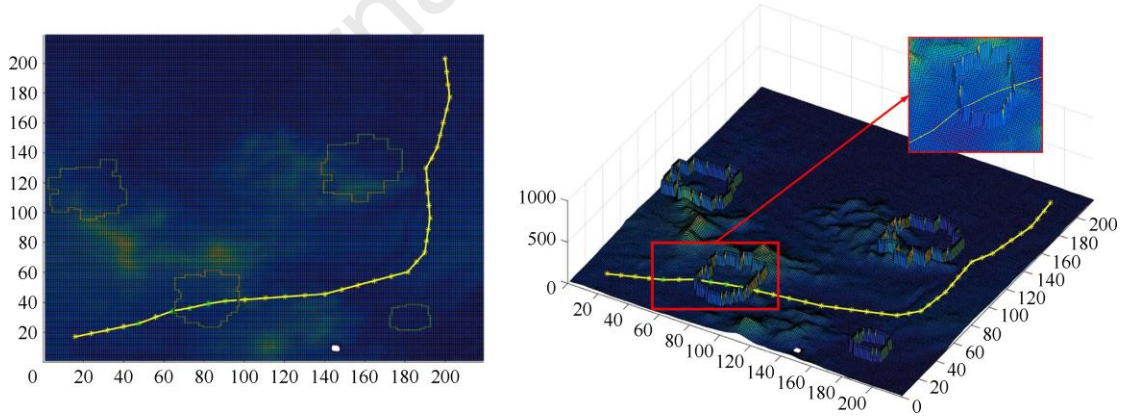


Fig. 22. The final flight path by the CTS-ABC algorithm in Map 1.

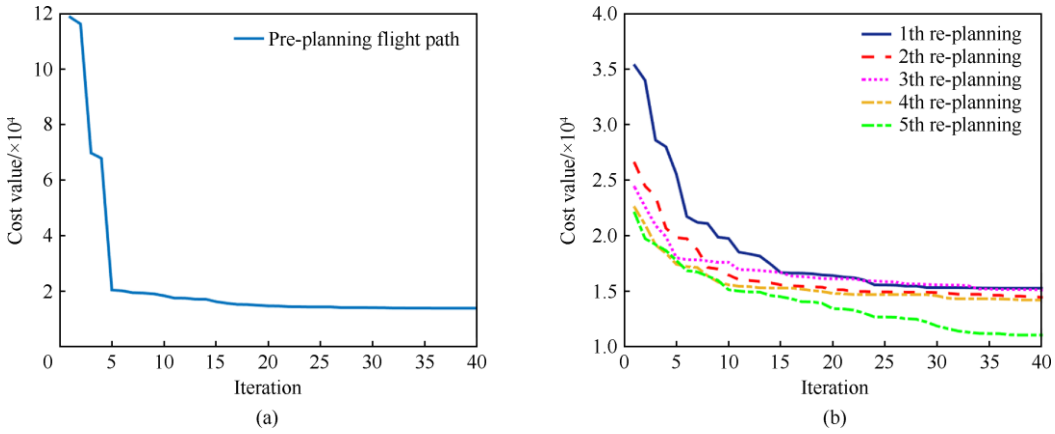


Fig. 23. The convergence results in Map 1.

Fig. 23 shows the convergence curves of the CTS-ABC algorithm in Map 1. Fig. 23(a) depicts the iterative process of the flight path pre-planned by the CTS-ABC algorithm. It can be intuitively seen that the iterations for the CTS-ABC algorithm to converge to the global optimum is 27. The curves in Fig. 23(b) indicate the iterative process of the five replanned flight paths using the CTS-ABC algorithm. Form the convergence curve, it is clear that the cost values for the five replanned paths are 1.71×10^4, 1.65×10^4, 1.59×10^4, 1.51×0^4, and 1.28×10^4, respectively. Since the waypoints of the five replans are closer, the cost values of these replanned paths are also closer. The statistical results of the path planning can be shown in Table 5.

Table 5

Statistical results of the CTS-ABC algorithm in Map 1。

Performances	f_1^{oc} / f_2^{oc} / f_3^{oc} (×10⁴)			Iterations/times	Optimization time/s	Collision warning points
Pre-planning	0.77	0.65	0.44	27	7.92	-
1th re-planning	0.64	0.59	0.48	29	6.40	[48,26,61]
2th re-planning	0.52	0.51	0.62	35	6.23	[64,34,80]
3th re-planning	0.47	0.49	0.63	38	6.64	[73,36,79]
4th re-planning	0.46	0.48	0.57	31	6.05	[82,39,79]
5th re-planning	0.41	0.39	0.48	36	6.11	[90,41,78]

5.2. Scenario 2

In the second scenario, a more complex planning environment is used to verify the effectiveness of the proposed method. Fig. 24 shows the diffusion process of dynamic threats in Map 2. Due to the more complex terrain, the dynamic threat spreads more widely across the map. The simulation of the threat diffusion process took 25 s, in other words, the simulated flight time of the UAH following the pre-planned path was also 25 s.

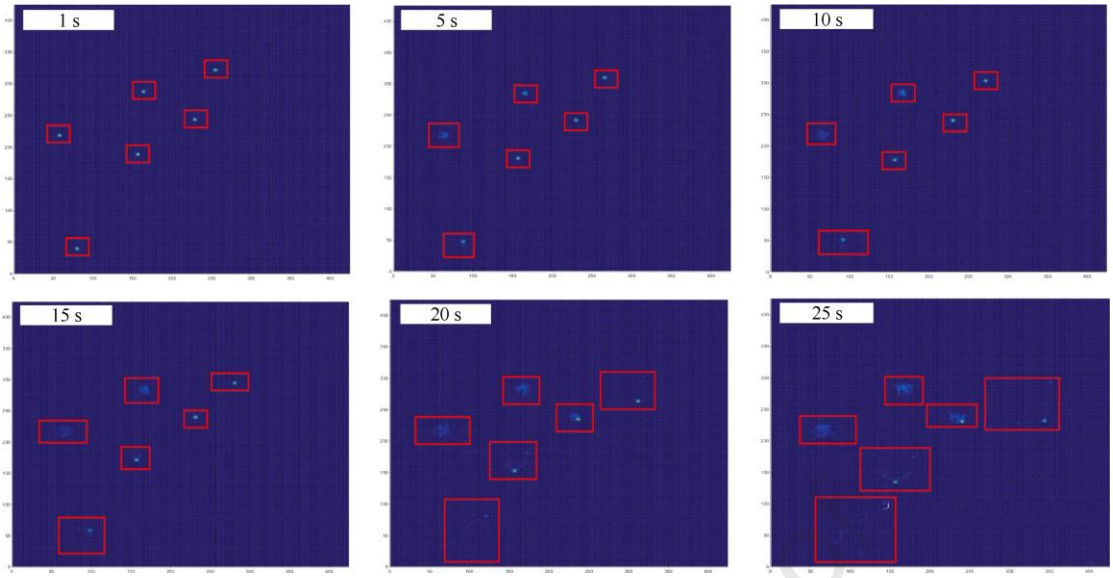


Fig. 24. The diffusion process of dynamic threat in Map 2.

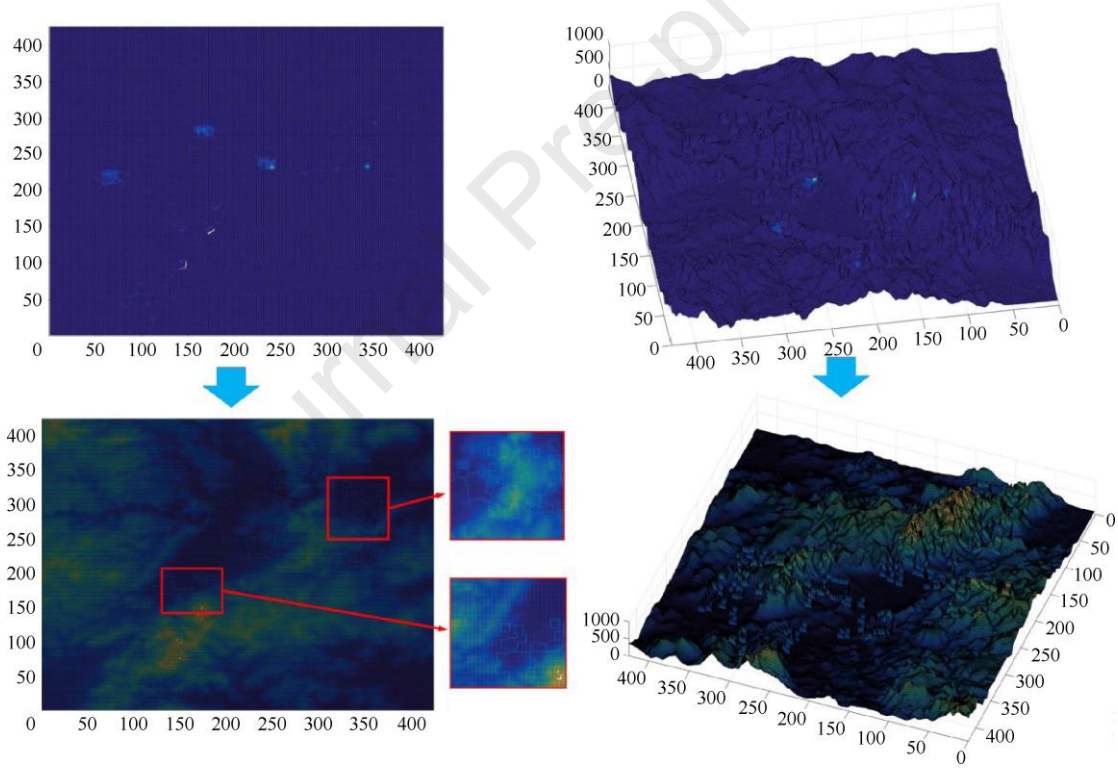


Fig. 25. The final shape of the dynamic no-fly zone Map 2.

Fig. 25 shows the final shape of the dynamic no-fly zone in Map 2. Compared to Map 1, the dynamic no-fly zones in Map 2 are more discrete in their distribution and more irregular in shape. The increase in the size and number of no-fly zones will directly increase the difficulty of UAH predictive path planning. Fig. 26 shows the part of the process of UAH predictive path planning in Map 2. Figs. 26(5)–Fig. 26(17) show the part of the process of UAH path replanning.

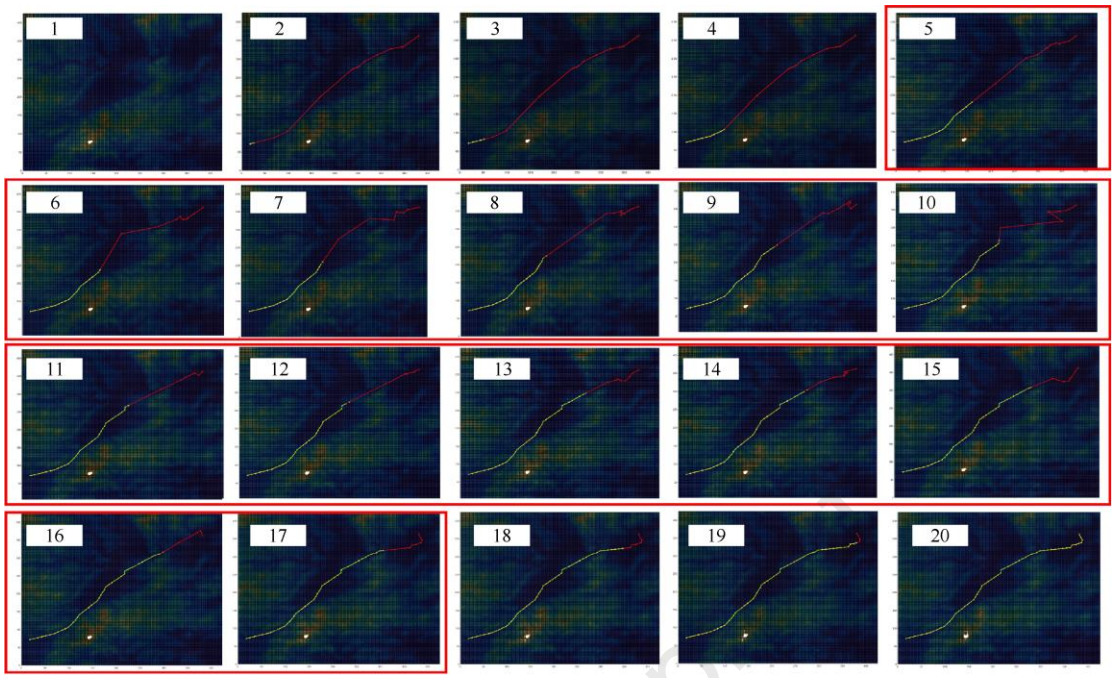


Fig. 26. The processes of pre-planning and re-planning Map 2.

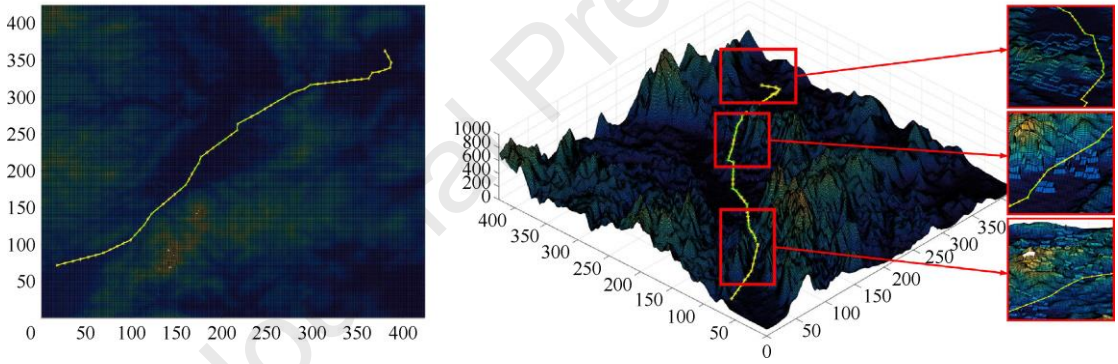
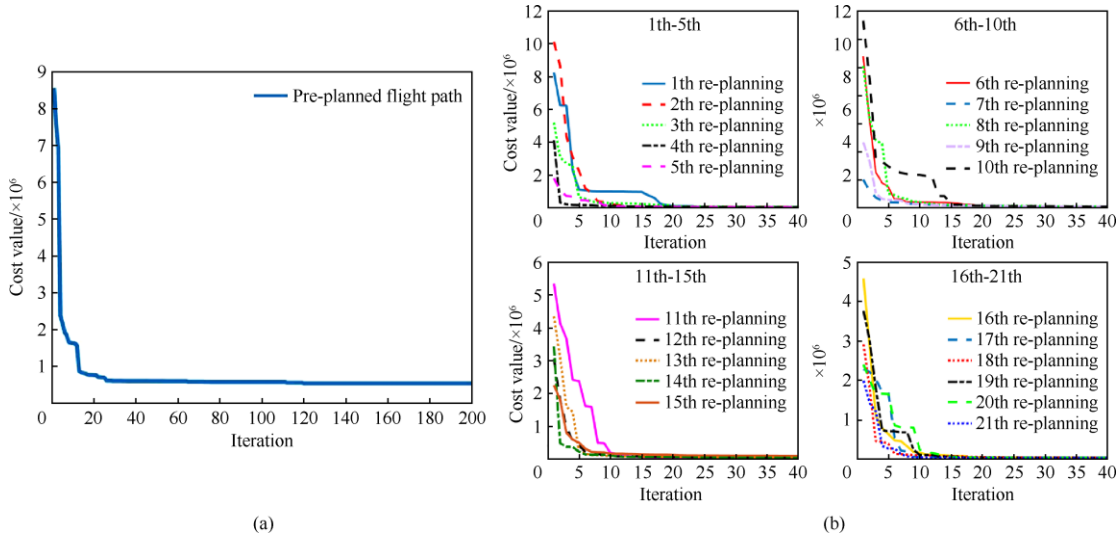


Fig. 27. The final flight path by the CTS-ABC algorithm in Map 2.

Fig. 27 shows the final path planning results of the CTS-ABC algorithm in Map 2. Due to the constantly changing dynamic no-fly zone, the predictive path planning was replanned 21 times.



(a) (b)

Fig. 28. The convergence results in Map 2.

Fig. 28 is the convergence curves of the CTS-ABC algorithm in the complex environment model. Form the simulation convergence curve, we can see that the CTS-ABC algorithm can find the optimal solution with fewer iterations in both path pre-planning and path re-planning. In Fig. 28(a), the CTS-ABC algorithm requires only 120 iterations to plan an initial flight path. The cost value of the initial flight path is 0.62×10^6. The curves in Fig. 28(b) indicate the iterative process of flight path re-planning using the CTS-ABC algorithm. The intuitive quantitative statistical results are shown in Table 6.

Table 6

Statistical results of the CTS-ABC algorithm in Map 2。

Performances	f_1^{oc} / f_2^{oc} / f_3^{oc} (×10⁶)			Iterations/times	Optimization time/s	Collision warning points
Pre-planning	0.347	0.109	0.164	120	16.3	-
1th re-planning	0.164	0.042	0.104	24	9.5	[105,113,345]
2th re-planning	0.132	0.064	0.108	23	9.3	[114,127,327]
3th re-planning	0.149	0.057	0.095	37	11.2	[123,142,309]
4th re-planning	0.117	0.035	0.144	35	10.6	[129,149,299]
5th re-planning	0.128	0.048	0.113	37	10.9	[154,175,284]
6th re-planning	0.136	0.039	0.099	38	11.4	[161,181,281]
7th re-planning	0.104	0.041	0.126	38	11.1	[171,203,271]
8th re-planning	0.111	0.032	0.138	37	10.5	[175,211,266]
9th re-planning	0.109	0.051	0.109	39	11.7	[177,219,250]
10th re-planning	0.123	0.032	0.107	37	10.2	[197,237,266]
11th re-planning	0.127	0.027	0.142	36	9.6	[204,243,265]
12th re-planning	0.093	0.029	0.078	35	10.3	[217,255,264]
13th re-planning	0.074	0.024	0.101	32	8.9	[218,264,250]
14th re-planning	0.065	0.019	0.101	29	8.6	[226,269,251]
15th re-planning	0.043	0.021	0.070	39	10.4	[240,279,241]
16th re-planning	0.086	0.017	0.110	30	9.7	[262,295,245]
17th re-planning	0.071	0.018	0.076	24	8.8	[269,299,246]
18th re-planning	0.077	0.019	0.075	33	9.3	[276,304,247]
19th re-planning	0.062	0.014	0.091	21	6.9	[283,309,247]
20th re-planning	0.049	0.016	0.094	25	7.7	[292,311,246]
21th re-planning	0.037	0.012	0.095	29	7.4	[299,317,254]

Simulation results show that the ground threat prediction path planning method of UAH can plan a safe flight path in the case of incomplete battlefield information. In addition, the collaborative thinking strategy enhances the optimization performance of the traditional ABC algorithm. Effective threat information prediction with hybrid enhanced CTS-ABC algorithm provides a strong guarantee for the UAH path planner.

6. Conclusions

In this paper, a ground threat prediction-based path planning method for UAH has been proposed. Firstly, a dynamic ground threat distribution probability model was developed based on the characteristics of typical air defense threats. The movement state of dynamic threats was fuzzy inferred from the distribution probability, based on which the possible distribution area of dynamic ground threats was simulated. Then, based on the probability of distribution of the ground threat, we have modelled dynamic no-fly zones for UAH path planning. The boundaries of the dynamic no-fly zone were determined and updated in real time by extracting the edge of the probability spread area. Furthermore, for improving the efficiency and quality of the UAH predictive path planner, a hybrid enhanced ABC algorithm based on collaborative thinking strategy has been proposed. By introducing the leader-member thinking mechanism and the collaborative learning update strategy, the optimization performance of ABC algorithm was released and the solution efficiency of UAH path planning system was improved. Finally, simulation results show that the proposed path planning method can provide a safe flight path for UAH in the presence of non-complete dynamic ground threats.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Joelianto E, Sumarjono QF, Budiyono A, et al. Model predictive control for autonomous unmanned helicopters. *Aircraft Engineering and Aerospace Technology* 2011; 6(83): 375-387.

[2] Ma HX, Chen M, Wu WX, et al. Switched safe tracking control design for unmanned autonomous helicopter with disturbances. *Nonlinear Analysis-hybrid System* 2021; 39: 100979.

[3] Liao SL, Zhu RM, Wu NQ, et al. Path planning for moving target tracking by fixed-wing UAV. *Defence Technology* 2020; 16(4): 811-824.

[4] Lv XX, Li WH, Wang JF. Safety-field-based path planning algorithm of lane changing for autonomous vehicles. *International Journal of Control, Automation, and Systems* 2022; 2(20): 563-576.

[5] Wang XW, Peng HJ, Liu J, et al. Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck. *Defence Technology* 2022; 18(2): 238-248.

[6] Botzheim J, Toda Y, Kubota N. Bacterial memetic algorithm for offline path planning of mobile robots. *Memetic Computing* 2012; 1(4): 73-86.

[7] Jain J, Kolbe U, Breuel G, et al. Collision avoidance for multiple static obstacles using path-velocity decomposition. *IFAC Papersonline* 2019; 52(8): 356-270.

[8] Li LN, Guo YQ, Zhang XD, et al. Path planning algorithm for robot based on firefly algorithm combined with

artificial potential field method. *Computer Engineering and Application* 2018; 54(20): 104-109.

[9] Kothari M, Postlethwaite I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. *Journal of Intelligent and Robotic System* 2013; 71(2): 231-253.

[10] Le AV, Prabakaran V, Sivanantham V, et al. Modified A-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors* 2018; 18(8): 2585.

[11] Nie JL, Zhang QJ, Wang YF. UAV path planning based on weighted-Voronoi diagram. *Flight Dynamics* 2015; 33(4): 339-343.

[12] Rantanen MT, Juhola M. Using probabilistic roadmaps in changing environments. *Computer Animation and Virtual Worlds* 2014; 25(1): 17-31.

[13] Primiero G. Algorithmic iteration for computational intelligence. *Minds and Machines* 2017; 27(3): 521-543.

[14] Castellano G, Fanelli AF, Torsello MA. Computational intelligence techniques for web personalization. *Web Intelligence and Agent Systems* 2008; 6(3): 253-272.

[15] Phung MD, Ha QP. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Applied Soft Computing* 2021; 107: 107376.

[16] Zhang XY, Xia S, Zhang T, et al. Hybrid FWPS cooperation algorithm based unmanned aerial vehicle constrained path planning. *Aerospcae Science and Technology* 2021; 118: 107004.

[17] Han ZL, Chen M, Shao SY, et al. Improved artificial bee colony algorithm-based path planning of unmanned autonomous helicopter using multi-strategy evolutionary learning. *Aerospcae Science and Technology* 2022; 122: 107374.

[18] Sun Z, Shao ZF, Li H. An eikonal equation based path planning method using polygon decomposition and curve evolution. *Defence Technology* 2020; 16(5): 1011-1018.

[19] Mannarini G, Subramani DN, Lermusiaux PFJ. Graph-Search and differential equations for time-optimal vessel route planning in dynamic ocean waves. *IEEE Transactions on Intelligent Transportation Systems* 2020; 21(8): 3581-3593.

[20] Hesham O, Wainer G. Advanced models for centroidal particle dynamics: short-range collision avoidance in dense crowds. *Simulation-transactions of the Society for Modeling and Simulation International* 2021; 97(8): 529-543.

[21] Yuan Y, Wang HJ, Lin CJ, et al. A novel GRU-RNN network model for dynamic path planning of mobile robot. *IEEE Access* 2019; 7: 15140-15151.

[22] Yu KL, Ping JX, Cao YH, et al. Local path planning based on mixed arithmetic. *Journal of Academy of Armored Force Engineering* 2008; 22(2): 43-45.

[23] Hajiloo R, Abroshan M, Khajepour A. Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems* 2020; 22(5): 3167-3178.

[24] Peng Y, Li SW, Hu ZZ. A self-learning dynamic path planning method for evacuation in large public buildings based on neural networks. *Neurocomputing* 2019; 365: 71-85.

[25] Kashyap AK, Parhi DR, Muni MK, et al. A hybrid technique for path planning of humanoid robot NAO in static and dynamic terrains. *Applied Soft Computing* 2020; 95: 106581.

[26] Jeong S, You K, Seok D. Hazardous flight region prediction for a small UAV operated in an urban area using a deep neural network. *Aerospcae Science and Technology* 2021; 118: 107060.

[27] Li SY, Chen M, Wang YH, et al. Air combat decision-making of multiple UCAVs based on constraint strategy games. *Defence Technology* 2022; 18(3): 368-383.

[28] Ostrowski R, Cywinsli A, Strzelec M. Electronic warfare in the optical band: Main features, examples and selected measurement data. *Defence Technology* 2021; 17(5): 1636-1649.

[29] Yoon YJ, Kim MG, Kim YD. Three-dimensional path planning for aerial refueling between one tanker and multiple UAVs. *International Journal of Aeronautical and Space Sciences* 2018; 19(4): 1027-1040.

[30] Maathuis C, Pieters W, van den Berg J. Decision support model for effects estimation and proportionality assessment for targeting in cyber operations. *Defence Technology* 2021; 17(2): 352-374.

[31] Guo X, Zhu DH, Zou XD, et al. Dynamic inertia evaluation for type-3 wind turbines based on inertia function. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2021; 11(1): 28-38.

[32] Bouchard F, Mian A, Jialun Z, et al. Riemannian geometry for compound gaussian distributions: application to recursive change detection. *Signal Processing* 2020; 176: 107716.

[33] Han ZL, Chen M, Zhou TL, et al. Path planning of unmanned autonomous helicopter based on human-computer hybrid augmented intelligence. *Neural Plasticity* 2021; 2021: 6639664.

[34] Lin YC, Saripalli S. Sampling-based path planning for UAV collision avoidance. *IEEE Transactions on Intelligent Transportation Systems* 2017; 18(11): 3179-3192.

[35] Xu CF, Duan HB, Liu F. Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning. *Aerospcae Science and Technology* 2010; 14(8): 535-541.

[36] Yu S, Ding L, Wu HT. Path planning for UAVs based on improved artificial bee colony algorithm. *Electronics Optics and Control* 2017; 24(1): 19-23.

[37] Lin C, Han HJ, Du JX, et al. Adaptive traffic engineering based on active network measurement towards software defined internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems* 2021; 22(6): 3697-3706.

[38] Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences* 2012; 192: 120-142.

[39] Chen X, Xu B, Mei CL, et al. Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation. *Applied Energy* 2018; 212: 1578-1588.

[40] Patil SD, Ragha L. Adaptive fuzzy-based message dissemination and micro-artificial bee colony algorithm

optimised routing scheme for vehicular ad hoc network. *IET Communications* 2020; 14(6): 994-1004.

[41] Arana-Jimenez M, Antczak T. The minimal criterion for the equivalence between local and global optimal solutions in nondifferentiable optimization problem. *Mathematical Methods in The Applied Science*s 2017; 40(18): 6556-6564.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: