Journal of Systems Engineering and Electronics
Vol. 34, No. 5, October 2023, pp.1191 – 1210

Combat situation suppression of multiple UAVs based on spatiotemporal cooperative path planning

HU Lei, YI Guoxing*, NAN Yi, and WANG Hao
School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

Abstract: Aiming at the suppression of enemy air defense (SEAD) task under the complex and complicated combat scenario, the spatiotemporal cooperative path planning methods are studied in this paper. The major research contents include optimal path points generation, path smoothing and cooperative rendezvous. In the path points generation part, the path points availability testing algorithm and the path segments availability testing algorithm are designed, on this foundation, the swarm intelligence-based path point generation algorithm is utilized to generate the optimal path. In the path smoothing part, taking terminal attack angle constraint and maneuverability constraint into consideration, the Dubins curve is introduced to smooth the path segments. In cooperative rendezvous part, we take estimated time of arrival requirement constraint and flight speed range constraint into consideration, the speed control strategy and flight path control strategy are introduced, further, the decoupling scheme of the circling maneuver and detouring maneuver is designed, in this case, the maneuver ways, maneuver point, maneuver times, maneuver path and flight speed are determined. Finally, the simulation experiments are conducted and the acquired results reveal that the time-space cooperation of multiple unmanned aeriel vehicles (UAVs) is effectively realized, in this way, the combat situation suppression against the enemy can be realized in SEAD scenarios.

Keywords: heterogeneous unmanned aeriel vehicles (UAVs), situation suppression, coope rative rendezvous, maneuver strategy, multiple constraints.

DOI: 10.23919/JSEE.2023.000119

1. Introduction

Unmanned aerial vehicles (UAVs) have the advantages of low cost, small size, reusability, zero casualty and strong adaptability, etc. [1−3]. With the development of UAV control, task planning, artificial intelligence [4], etc., UAVs swarm have been extensively applied to perform various tasks, including target search [5], transportation [6], and disaster relief [7]. Suppression of enemy air defense (SEAD) task [8] is a classic military operation with the purpose of destroying the enemy's valuable targets based on various air combat force. Various heterogeneous UAVs can replace human beings to perform dangerous tasks in complex environment [9], such as SEAD task. Planning the flyable and safe path can ensure that UAVs reach the predetermined position to realize the situation suppression against the enemy. Consequently, it is of great importance to research the path planning methods and maneuver strategies.

In essence, realizing the situation suppression against the enemy belongs to multi-UAVs cooperative path planning problems. Tsourdos et al. [10] gave the concrete definition of the multi-UAVs cooperative path planning problem, that is, the safe and reliable flight paths are planned with the considerations of various constraints and cooperative relationship, which ensure UAVs to reach the pre-designated task locations on schedule. Environment constraints include local climate, various obstacles and no-fly areas [11,12], etc. Target constraints include resource requirement [13], terminal angle [8], etc. UAVs constraints include effective load, endurance capability, flight speed range, safe radius [14], maneuverability [8], etc. Cooperative relationship includes time cooperation and space cooperation. For the SEAD task, the available paths of UAVs should be planned according to the considered constraints and the maneuver schemes of UAVs should be designed according to the cooperative relationship.

In SEAD task, avoiding these threats such as obstacles, no-fly areas and enemy's radars, and planning the safe and flyable path are the foundation for UAVs to realize the situation suppression against the enemy. The classical path planning methods include differential evolution algorithm [12], whales optimization algorithm [15], ant colony optimization (ACO) algorithm [11,16,17], particle swarm optimization (PSO) algorithm [18,19], genetic

Manuscript received August 25, 2021.
*Corresponding author.

algorithm (GA) [8,20], rapidly-exploring random trees (RRT) algorithms [21–23], and A-star algorithm [24,25]. The search space of A-star algorithm increases exponentially, when there are multiple extreme values, it cannot guarantee to get the optimal path. The search efficiency of the RRT algorithm is fantastic, but the path does not have asymptotic optimality. Consequently, many scholars paid much attention to research the RRT star algorithm [26] and improved RRT star algorithms [27]. ACO, PSO, GA, etc., belong to swarm intelligence optimization algorithms, these algorithms have the advantages of simple structure and easy implementation, consequently, these algorithms are widely applied to solve the path planning problem [8,11,16–20].

Many scholars paid much attention to researching the cooperative rendezvous of multi-UAVs, Oh et al. [28] gave the definition of the cooperative rendezvous, that is, UAVs can reach the pre-designated task locations simultaneously or continuously by adjusting the flight speed or flight path. Generally speaking, rendezvous strategies include speed control strategy and flight path control strategy. Keeping the flight path unchanged, the rendezvous task is realized by adjusting the flight speed, which is called speed control strategy. Keeping the flight speed unchanged, the rendezvous task is realized by adjusting the flight path, which is called flight path control strategy. The criterion of judgement is that the estimated time of arrival (ETA) of UAV satisfies with the anticipated scheme, here the ETA represents the time that UAV spends on the way to the designated task location [29]. Because the flight speed range of UAV is bounded, realizing complicated rendezvous task based on speed control is limited [30]. However, speed control is still widely utilized to solve rendezvous task, Shan et al. [31] firstly completed the multi-UAVs path planning by combining PSO and Hook-Jeeves search algorithm, secondly, the ETA rendezvous task was achieved based on speed control. For the rendezvous task of short-range UAV and long-range UAV, Duan et al. [32] realized the expected ETA rendezvous task by adjusting the flight path. Aim at the rendezvous task of homogeneous UAVs with the same flight speed, the detouring maneuver and circling maneuver are applied to make UAVs arriving the pre-designated task locations and attacking the targets simultaneously [33].

These algorithms mentioned above can solve the single UAV path planning and multi-UAVs cooperative path planning, however, there are few works which concentrate on the multiple heterogeneous UAVs cooperative situation suppression realization problem, especially considering the ETA constraint, terminal angle constraint, flight speed range constraint and maneuverability constraint, etc. In addition, threats existing in task scenarios are usually modeled by simple circle-shape [12,21], this modeling method is one-sided. Actually, it is appropriate to build threat models with different geometric figures like circle, triangle and ellipse according to real physical entity. As far as we know, few researchers pay attention to the cooperative rendezvous with the comprehensive consideration of speed control strategy and flight path control strategy. In view of the above problems, this article dedicates to realizing the cooperative rendezvous problem for multiple heterogeneous UAVs. The major contributions are presented as follows:

(i) The path planning problem of multi heterogeneous UAV in complex environment is modeled, and the swarm intelligence-based path point generation (SI-PPG) algorithm is designed.

(ii) The Dubins curve is applied to deal with the terminal angle constraint and maneuverability constraint, the acquired smoothed flight paths of UAVs ensure the space collaboration in SEAD tasks.

(iii) With the considerations of ETA requirement and flight speed range, the speed control strategy and flight path control strategy are designed to realize the cooperative rendezvous of multi-UAVs.

The rest of this paper is organized as follows: The SEAD task scenario is presented in Section 2. The path smoothing based on Dubins curve is introduced in Section 3. In Section 4, the proposed SI-PPG algorithm is given. In Section 5, the cooperative rendezvous strategy is introduced. In Section 6, the simulation experiments are conducted and the results are analyzed. At last, the conclusion is given in Section 7.

2. SEAD task model

2.1 SEAD task scenario

The SEAD task scenario is depicted in Fig. 1, supposing that there are N_U heterogeneous UAVs with different attributes, N_T targets with various requirements and M threats modeled by different shapes in the SEAD task scenario. The combat purpose of multi-UAVs is to realize the combat situation suppression oriented to these hostile targets. Each UAV must arrive at the pre-designated location point concurrently and respectively perform attack tasks simultaneously. Consequently, planning the effective and safe paths for the multi-UAVs is of great importance to realize the combat situation suppression. In this paper, this problem is divided into three parts including path points generation, path smoothing and cooperative rendezvous.

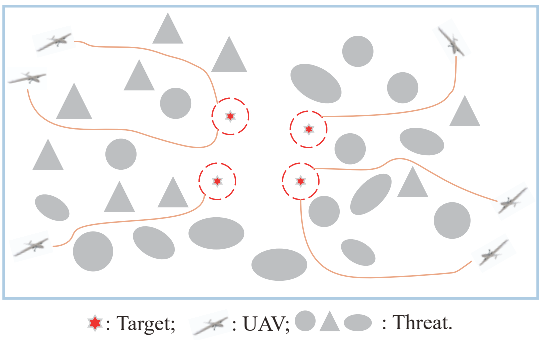


Fig. 1 SEAD task scenario

In the path points generation part, the expandable SI-PPG algorithms are designed to generate the broken-line path for each UAV, the path has no conflict with the various threats existing in SEAD scenarios. Considering the terminal angle constraint and maneuverability constraint, the Dubins curve is applied to smooth the paths of UAVs. In the cooperative rendezvous part, the cooperative rendezvous strategy is designed that UAVs can arrive at the location on schedule and perform the pre-designated tasks. In this paper, Dubins path length (DPL) depicts the length of the Dubins path from the starting point to terminal point. ETA depicts the flight time.

2.2 Target model

The symbol Tar represents the target, the attributes of target i can be depicted by the multivariate array <$\mathrm{Tar}_i, \boldsymbol{p}_{\mathrm{Tar}_i}, r_{\mathrm{Tar}_i}, \boldsymbol{p}_{\mathrm{Tar}_i,k}$>, where Tar_i depicts target i, $\boldsymbol{p}_{\mathrm{Tar}_i}$ depicts the location of target i and $\boldsymbol{p}_{\mathrm{Tar}_i} = (x_{\mathrm{Tar}_i}, y_{\mathrm{Tar}_i})$, r_{Tar_i} is the attack radius of target i, $\boldsymbol{p}_{\mathrm{Tar}_i,k}$ depicts the kth pre-designated rendezvous location of the target i and $\boldsymbol{p}_{\mathrm{Tar}_i,k} = (x_{\mathrm{Tar}_i,k}, y_{\mathrm{Tar}_i,k})$. The sketch map of the target is depicted in Fig. 2, here $\boldsymbol{p}_{\mathrm{Tar}_i,1}$ and $\boldsymbol{p}_{\mathrm{Tar}_i,2}$ denote that two UAVs are needed to attack the target.

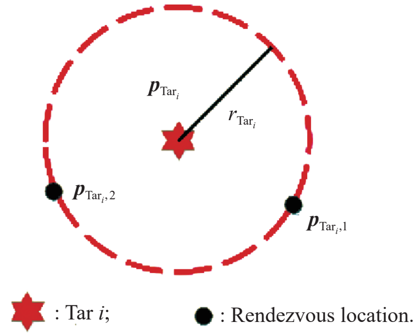
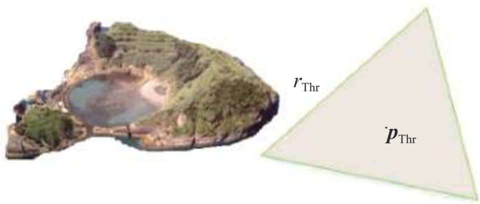


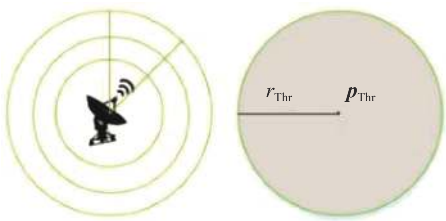
Fig. 2 Target model

2.3 Threat model

In this paper, no-fly area, the hostile detection radar and the offensive weapon, etc., are called threats, these threats are described as simple circle-shape in the traditional modeling method [21]. Here these threats are modeled into different shapes as depicted in Fig.3, including regular triangle, circle and ellipse, these threats with different azimuth angles maybe exist in the SEAD scenario. Let the symbol Thr represent the threat, then the attributes of threat i can be depicted by the multivariate array <$\mathrm{Thr}_i, s_{\mathrm{Thr}_i}, \boldsymbol{p}_{\mathrm{Thr}_i}, r_{\mathrm{Thr}_i}, \theta_{\mathrm{Thr}_i}$>, where Thr_i depicts threat i, s_{Thr_i} depicts the shape of threat i, $\boldsymbol{p}_{\mathrm{Thr}_i}$ depicts the center of threat i and $\boldsymbol{p}_{\mathrm{Thr}_i} = (x_{\mathrm{Thr}_i}, y_{\mathrm{Thr}_i})$, r_{Thr_i} is the size of threat i, θ_{Thr_i} depicts the azimuth angle. The parameter description is given in Table 1.



(a) Regular triangle threat



(b) Circle-shape threat

(c) Ellipse-shape threat

Fig. 3 Threats model

Table 1 Parameter description of threat

Type of threat s_{Thr}	Size of threat r_{Thr}
Regular triangle	Side length
Circle	Radius
Ellipse	Semi-minor axis，semimajor axis is $\sigma r_{\text{Thr}}(\sigma>1)$

2.4 Dubins car model

The multivariate array $<U_i, \boldsymbol{p}_{U_i}(t), r_{U_i}, \omega_{U_i}, [v_{U_i,\min}, v_{U_i,\max}]>$ is used to describe the attributes of UAV i, where the symbol U_i represents UAV i, the symbol $\boldsymbol{p}_{U_i}(t)$ depicts the location of UAV i and $\boldsymbol{p}_{U_i}(t)=(x_{U_i}(t), y_{U_i}(t))$, the symbol $\boldsymbol{p}_{U_i}(0)$ depicts the starting point of UAV i and $\boldsymbol{p}_{U_i}(0)=(x_{U_i}(0), y_{U_i}(0))$, $\boldsymbol{p}_{U_i}(t)$ depicts the location of UAV i at time t and $\boldsymbol{p}_{U_i}(t)=(x_{U_i}(t), y_{U_i}(t))$, the symbol r_{U_i} denotes the turning radius corresponding to the flight speed v_{U_i} and $v_{U_i}\in[v_{U_i,\min}, v_{U_i,\max}]$, ω_{U_i} is the angular velocity of UAV i.

There are lots of works which dedicate to studying the path planning, however these works did not give the flyable path [18,20]. In this paper, the UAV is regarded as Dubins car, the basic kinematic model is formulated [34,35].

$$\begin{cases} \dot{x}_{U_i} = v_{U_i}\cos\theta \\ \dot{y}_{U_i} = v_{U_i}\sin\theta \\ \dot{\theta}_{U_i} = u_{\theta_{U_i}} \end{cases} \tag{1}$$

where v_{U_i} is the flight speed，θ_{U_i} is the speed direction, and $u_{\theta_{U_i}}$ is the control variable.

The turning radius corresponding to flight speed v_{U_i} is calculated as follows:

$$r_{U_i} = \frac{v_{U_i}}{\omega_{U_i}}. \tag{2}$$

3. Dubins curve-based path smoothing

3.1 Dubins curve

Dubins path is the shortest path satisfying the curvature constraint in two-dimensional Euclidean plane [34,35]. Given the starting point and the terminal point, Dubins path between the two points is composed of arc and line-segments, here the arbitrary terminal angle of the terminal point can be handled to realize the combat situation suppression. S is the starting point and E is the terminal point, and the direction angles of both points are known, which depicts the speed direction of UAV at these path points. Then the shortest Dubins path from S to E can be denoted as

$$\text{dub}(S,E) = \min\{\text{dub}_{\text{RSR}}, \text{dub}_{\text{RSL}}, \text{dub}_{\text{LSR}}, \text{dub}_{\text{LSL}}, \text{dub}_{\text{RLR}}, \text{dub}_{\text{LRL}}\} \tag{3}$$

where $\text{dub}(S,E)$ depicts the shortest Dubins path. For the limited space of this paper, here we do not concentrate on the specific details corresponding to (3), interested readers can refer to [34] and [35].

As depicted in Fig. 4, there are L points which are utilized to connect the starting point and terminal point. It is obvious that the path cannot satisfy with the considered terminal angle constraint and maneuverability constraint, consequently, the Dubins cure is applied to smooth these path segments and get the flyable path. In this paper, we make the following agreements that the direction angle of UAV at current path point is to point to the next path point. As depicted in Fig. 4, the direction angle of the UAV at the ith path point is to point to the (i+1)th path point. According to the coordinates of path points and tangent function, it is easy to calculate the direction angles of the L middle path points.

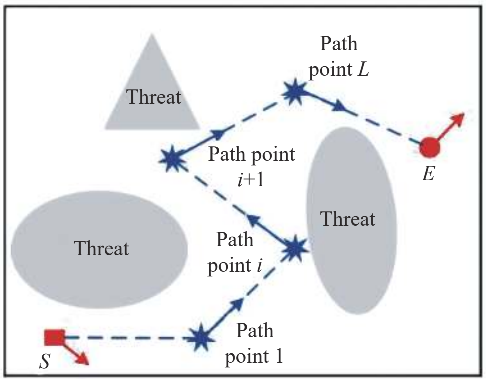


Fig. 4 Direction angle of middle path points

3.2 Path smoothing realization

Here the Dubins-curve-based case of choosing the shortest flyable path is given. Let the turning radius of UAV be 0.4 km, the coordinate of the starting point is (0,1), and the direction angle of the starting point is π/2. The coordinate of terminal point is (0.5,1.5), and the direction angle of terminal point is π/4 which is called terminal angle in this paper. Six Dubins paths with regard to RSR, RSL, LSR, LSL, RLR and LRL are presented in Fig. 5, and the corresponding distances of RSR, RSL, LSR, LSL, RLR and LRL are listed in Table 2.

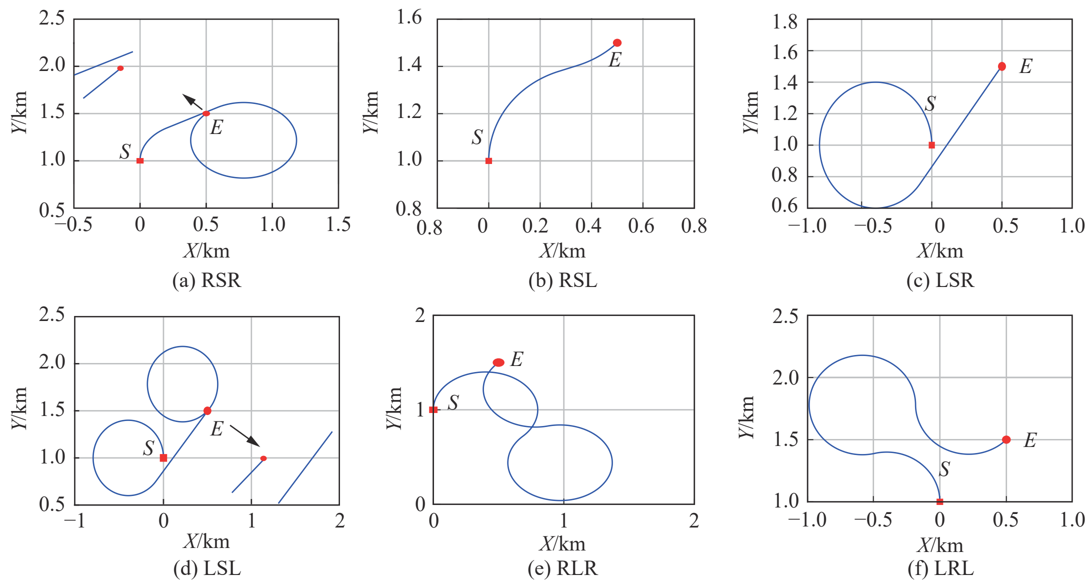


Fig. 5 Sketch map of Dubins path

Table 2 Length of the six Dubins paths

Path	Length/km
RSR	5.7092
RSL	0.75918
LSR	3.1962
LSL	5.7902
RLR	4.8948
LRL	3.6362

It is obvious that the shortest Dubins path from the starting point to the terminal point is depicted by RSL-path according to Fig. 5 and Table 2. Thus, the shortest flyable path from the starting point to the terminal point with the considerations of terminal angle constraint and maneuverability constraint can be acquired.

Here another case for generation of Dubins path with multi-middle path points is given, supposing that the turning radius of the UAV is 0.4 km, the coordinate of starting point is (1,1), and the direction angle of starting point is $\pi/2$. The coordinate of terminal point is (15,15), and the direction angle of terminal point is $-\pi/2$. There are four middle path points, their coordinates are (3,1), (2,5), (8,7), (12,14).

As depicted in Fig. 6, the path segments from the starting point to the terminal point through the four middle path points are depicted in Fig. 6(a), and the flyable Dubins path is drawn in Fig. 6(b). It is obvious that the flyable path satisfies the maneuverability constraint of UAV, and the terminal attack angle of UAV is $-\pi/2$ that meets the expected setting. In this way, the flyable path from starting point to terminal point through L middle path points with the consideration of terminal angle constraint and maneuverability constraint is acquired.

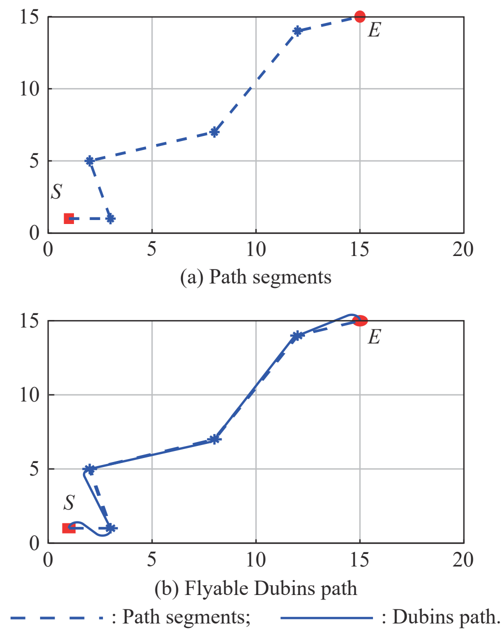


Fig. 6 Path of UAV

4. SI-PPG

Path points generation in the continuous complicated task scenario with multiple constraints is a classic non-linear optimization problem. The basic purpose of path points generation is to generate L points to connect the starting point and the terminal point, meanwhile the acquired path must avoid the threats and satisfy the shortest path requirement. A direct idea for generating L points in two-dimensions Euclidean axis is to build a $2L$-dimensional vector which can form the L points in a certain way. Swarm intelligence optimization algorithms like PSO and Bats algorithms are unavailable to deal with the complex constraints considered in this SEAD scenario, the penalty function method can turn the constrained optimization problem into unconstrained optimization problem [36]. In this part, the availability testing algorithms of path points and path segments are given at first. Then, the fitness function is built based on the penalty function method and search rules are designed to promote search efficiency. Finally, the detailed flow-chart of the SI-PPG algorithm is presented, which is the organic composition of the availability testing algorithm, the penalty function method and the swarm intelligence algorithm. It is worth noting that we do not pay much attention to the specific theory of swarm intelligence algorithms, but we focus on the realizable strategies for path points generation including the availability testing algorithms of path points and path segments, fitness function design and search rules design.

4.1 Availability testing algorithms for path point (ATPP)

The symbol A_i denotes path point i, if A_i has no conflict with all threats, then A_i is available, otherwise A_i is unavailable which means the path point A_i is overlapping with threats. The overlapping sketch-maps between path point A_i and threats with different threat models are depicted in Fig. 7.

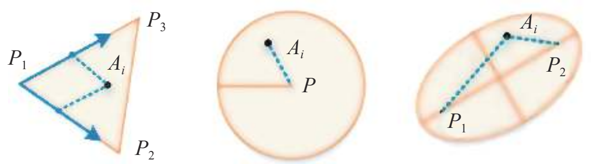


Fig. 7 Overlapping sketch-maps

Aiming at the regular triangle threat, if A_i is inside the triangle, the geometric relations of A_i and the three vertices can be denoted as follows:

$$A_i = P_1 + u(P_3 - P_1) + w(P_2 - P_1) \tag{4}$$

where $u \in [0,1]$, $w \in [0,1]$ and $0 \leqslant u+w \leqslant 1$, if $u=0$, $w=0$, A_i is the vertex P_1, if $u=0$, $w=1$, A_i is the vertex P_2, if $u=1$, $w=0$, A_i is the vertex P_3. Let $\upsilon_0 = P_3 - P_1$, $\upsilon_1 = P_2 - P_1$, $\upsilon_2 = A_i - P_1$, $\upsilon_2 = u\upsilon_0 + w\upsilon_1$, we can get the following equations by multiplying υ_0 and υ_1 on both sides respectively.

$$\upsilon_2\upsilon_0 = u\upsilon_0 \cdot \upsilon_0 + w\upsilon_1 \cdot \upsilon_0 \tag{5}$$

$$\upsilon_2\upsilon_1 = u\upsilon_0 \cdot \upsilon_1 + w\upsilon_1 \cdot \upsilon_1 \tag{6}$$

Then we can get a conclusion that if these three conditions are satisfied, $u \in [0,1]$, $w \in [0,1]$ and $0 \leqslant u+w \leqslant 1$, the path point A_i is unavailable which means that path point A_i is inside in the regular triangle threat, otherwise the path point A_i is available.

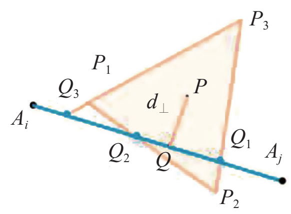
Aiming at circle-shape threat with radius r_{Thr}, the distance between A_i and the center of the circle can be denoted as d_{A_iP}, if $d_{A_iP} \leqslant r_{\text{Thr}}$, the path point A_i is unavailable, otherwise the path point A_i is available. Aiming at the ellipse-shape threat, the distances between path point A_i and the two ellipse foci can be denoted as $d_{A_iP_1}$ and $d_{A_iP_2}$, if $d_{A_iP_1} + d_{A_iP_2} \leqslant 2\sigma r_{\text{Thr}}$, σr_{Thr} is the semimajor axis, the path point A_i is unavailable, otherwise the path point A_i is available. The ATPP is presented. flag$A_i = 0$ represents the path point A_i is available and flag$A_i = 1$ represents the path point A_i is unavailable.

Algorithm 1 ATPP algorithm

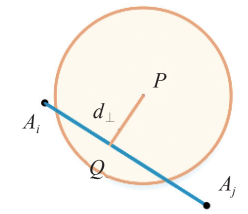
Input: s_{Thr}
Output: flagA_i
1) Let flag$A_i = 0$
2) **Switch** s_{Thr}
3) **Case** 1
4) **If** $u \in [0,1]$, $w \in [0,1]$ and $0 \leqslant u+w \leqslant 1$, flag$A_i = 1$. **End**
5) **Case** 2
6) **If** $d_{A_iP} \leqslant r_{\text{Thr}}$, flag$A_i = 1$. **End**
7) **Case** 3
8) **If** $d_{A_iP_1} + d_{A_iP_2} \leqslant 2\sigma r_{\text{Thr}}$, flag$A_i = 1$. **End**
9) **End**

4.2 Availability testing algorithm for the path segment (ATPS)

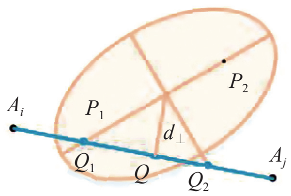
Let A_i and A_j represent path point i and path point j respectively. $A_i \to A_j$ represents the path segment from A_i to A_j, here the ATPS $A_i \to A_j$ is introduced. At first, A_i and A_j should be checked based on the ATPP algorithm, if these two path points are available, then we should check the availability of the path segment $A_i \to A_j$. The sketch-maps between path segment $A_i \to A_j$ and threats are depicted in Fig. 8. The distance between A_i and A_j is calculated and denoted as $d_{A_iA_j}$, and the straight-line equation constituted by A_i and A_j is determined.



(a) Regular triangle threat



(b) Circle threat



(c) Ellipse threat

Fig. 8 Sketch-maps for path segment and threats

Aiming at the regular triangle threat, the circumcircle radius is denoted as r_w, it is easy to find the perpendicular-line from the triangle center to the line A_iA_j and determine the intersection Q between the perpendicular-line and line A_iA_j, the vertical distance between the intersection and the triangle center is denoted as $d_\perp$. If $d_\perp \leqslant r_w$, we can calculate the intersections between the three sides of triangle and line A_iA_j, and named Q_1, Q_2, Q_3, if the path segment $A_i \to A_j$ intersects with the triangle, at least two of the three intersections (Q_1, Q_2, Q_3) are in the circumcircle of the triangle. Let d_{QA_i} represent the distance between intersection Q and A_i, and d_{QA_j} represent the distance between intersection Q and A_j, if the path segment $A_i \to A_j$ intersects with the triangle, the equation $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$ is valid. Consequently, if the following three conditions are satisfied: $d_\perp \leqslant r_w$; at least two of the three intersections (Q_1, Q_2, Q_3) are in the circumcircle of the triangle; $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$; then the path segment $A_i \to A_j$ is unavailable.

Aiming at circle-shape threat, if the following two conditions are satisfied: $d_\perp \leqslant r_{\rm Thr}$; $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$; the path segment $A_i \to A_j$ is unavailable. Aiming at the ellipse-shape threat, if the following three conditions are satisfied: $d_\perp \leqslant \sigma r_{\rm Thr}$; at least one of the two intersections (Q_1, Q_2) is in the ellipse; $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$; the path segment $A_i \to A_j$ is unavailable. Here Q_1 is the intersection between line A_iA_j and major axis of ellipse, Q_2 is the intersection between line A_iA_j and minor axis of ellipse. The ATPS is depicted in Algorithm 2. flag$A_iA_j = 0$ represents that the path segment $A_i \to A_j$ is available, flag$A_iA_j = 1$ represents that the path segment $A_i \to A_j$ is unavailable.

Algorithm 2 ATPS algorithm

Input: $s_{\rm Thr}$

Output: flagA_iA_j

1) **Let** flag$A_iA_j = 0$.
2) Compute $d_{A_iA_j}$.
3) Compute line determined by A_i, A_j.
4) **Switch** $s_{\rm Thr}$
5) **Case** 1
6) Compute circumcircle radius r_w.
7) Compute the intersection point Q and the vertical distance $d_\perp$.
8) **If** $d_\perp \leqslant r_w$.
9) Find the intersection points Q_1, Q_2, Q_3.
10) Let flag1 = 1, flag2 = 1.
11) **If** Q_1, Q_2, Q_3 are not in circumcircle, flag1 = 0. **End**
12) Compute distance d_{QA_i} and d_{QA_j}.
13) **If** $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$ is false, flag2 = 0. **End**
14) **If** flag1 & flag2 is true, flag$A_iA_j = 1$. **End**
15) **End**
16) **Case** 2
17) Compute the intersection point Q and compute the vertical distance $d_\perp$.
18) **If** $d_\perp \leqslant r_{\rm Thr}$.
19) Compute distance d_{QA_i} and d_{QA_j}.
20) **If** $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$ is true, flag$A_iA_j = 1$. **End**
21) **End**
22) **Case** 3
23) Compute the intersection points Q_1, Q_2.
24) Compute the intersection point Q.
25) **If** $d_\perp \leqslant \sigma r_{\rm Thr}$.
26) Let flag1 = 1, flag2 = 1.
27) **If** Q_1, Q_2 are not in ellipse, flag1 = 0. **End**
28) Compute distance d_{QA_i} and d_{QA_j}.
29) **If** $d_{QA_i} + d_{QA_j} - d_{A_iA_j} = 0$ is false, flag2 = 0. **End**
30) **If** flag1 & flag2 is true, flag$A_iA_j = 1$. **End**
31) **End**
32) **End**

4.3 SI-PPG algorithm

4.3.1 Search rules design

Here several basic search rules are introduced to promote the exploration efficiency.

(i) Search range bounded rule

The search range of path point i depends on the starting point and terminal point, let (x_{A_S}, y_{A_S}) represent the coordinate of starting point, and (x_{A_E}, y_{A_E}) represent the coordinate of terminal point, the search range of path point i can be denoted as follows:

$$\min\{x_{A_S}, x_{A_E}\} - \alpha \leqslant x_{A_i} \leqslant \max\{x_{A_S}, x_{A_E}\} + \alpha, \tag{7}$$

$$\min\{y_{A_S}, y_{A_E}\} - \beta \leqslant y_{A_i} \leqslant \max\{y_{A_S}, y_{A_E}\} + \beta, \tag{8}$$

where α and β are constants.

(ii) Population initialization rule

The size of the population is N_P, ζ is a constant prefixed and $\zeta \in (0,1)$. In the process of population initialization, at least $\lfloor \zeta N_P \rfloor$ individuals are guaranteed to satisfy the considered constraints, here $\lfloor \cdot \rfloor$ represents the rounding down symbol.

(iii) Availability rule for path point and path segment

The path point i and path point $i+1$ satisfy with the following formulas:

$$\sum_{j=1}^{M} p(A_i, \mathrm{Thr}_j) = 0, \tag{9}$$

$$p(A_i, \mathrm{Thr}_j) = \begin{cases} 1, & A_i \text{ is in } \mathrm{Thr}_j \\ 0, & A_i \text{ is out of } \mathrm{Thr}_j \end{cases}, \tag{10}$$

$$\sum_{j=1}^{M} q(A_i \to A_{i+1}, \mathrm{Thr}_j) = 0, \tag{11}$$

$$q(A_i \to A_{i+1}, \mathrm{Thr}_j) = \begin{cases} 1, & A_i \to A_{i+1} \text{ intersects with } \mathrm{Thr}_j \\ 0, & A_i \to A_{i+1} \text{ does not intersect with } \mathrm{Thr}_j \end{cases}, \tag{12}$$

where A_i represents path point i, Thr_j represents threat j $(j = 1, 2, \cdots, M)$, M is the number of threats, $A_i \to A_{i+1}$ represents the path segment between the two path points. $p(A_i, \mathrm{Thr}_j) = 0$ indicates that the path point is available with regard to threat j, $q(A_i \to A_{i+1}, \mathrm{Thr}_j) = 0$ indicates that the path segment is available with regard to threat j.

4.3.2 Fitness function design

The general constrained optimization problem can be depicted as follows:

$$\begin{cases} \min f(x) \\ \text{s.t. } c_i(x) \leqslant 0, \ \forall i = 1, 2, \cdots, h \end{cases}. \tag{13}$$

The basic idea of penalty function method turns the constrained optimization problem (13) into unconstrained optimization problem (14).

$$F = \min f(x) + \delta \sum_{i=1}^{h} g[c_i(x)] \tag{14}$$

where F is the fitness function, $g[c_i(x)]$ is the external function and $g[c_i(x)] = \max(0, c_i(x))^2$, δ is a penalty factor and is a large positive number.

Given that A_S represents the starting point and A_E represents the terminal point, the number of middle path points is L, then the path segments from starting point to terminal point through L middle path points can be denoted as $A_S \to A_1 \to \cdots \to A_L \to A_E$, the fitness function based on the penalty function method is designed as follows:

$$\min f(A_1, A_2, \cdots, A_L) = \min \sum_{i=1}^{L-1} d_{A_iA_{i+1}} + d_{A_SA_1} + d_{A_LA_E} + J_1 + I(J_1)J_2, \tag{15}$$

$$J_1 = \delta \sum_{i=1}^{L} \sum_{j=1}^{M} p(A_i, \mathrm{Thr}_j), \tag{16}$$

$$J_2 = \delta \left[\sum_{i=1}^{L-1} \sum_{j=1}^{M} q(A_i \to A_{i+1}, \mathrm{Thr}_j) + \sum_{j=1}^{M} q(A_S \to A_1, \mathrm{Thr}_j) + \sum_{j=1}^{M} q(A_L \to A_E, \mathrm{Thr}_j) \right], \tag{17}$$

$$I(J_1) = \begin{cases} 1, & J_1 = 0 \\ 0, & J_1 \neq 0 \end{cases}, \tag{18}$$

where (x_i, y_i) represents the coordinate of path point i, $d_{A_iA_{i+1}} = \sqrt{(x_{A_i} - x_{A_{i+1}})^2 + (y_{A_i} - y_{A_{i+1}})^2}$ depicts the distance between path point i and path point $i+1$. J_1 represents the penalty item for the availability testing of generated path points, J_2 represents the penalty item for the availability testing of generated path segments, δ is the penalty factor and is equal to 1 000 in this paper. $I(J_1)$ is the indicator function, $I(J_1) = 1$ represents that the generated path points are unavailable, then there is no need to compute J_2, which can reduce the computation consumption. The solution must satisfy the condition $J_1 = J_2 = 0$, in this case, L path points and $L+1$ path segments are available.

Here the flow-chart of SI-PPG algorithm is introduced in Fig. 9. The SI-PPG algorithm proposed in this paper is composed by four algorithms, including the ATPP algorithm, ATPS algorithm, penalty function method and swarm intelligence optimization algorithm. The ATPP algorithm can be used to judge the availability of path points, the ATPS algorithm can be used to judge the availability of path segments, the ATPP algorithm and the ATPS algorithm are elaborately introduced above. It is worth noting that we define three threat regions, including equilateral triangles, circles and ellipses in this paper. To solve this problem that how to avoid any threat region with arbitrary shape, we have the following two schemes. Firstly, we can use several equilateral triangles, circles and ellipses to depict any threat region with arbitrary shape, in this case, the proposed SI-PPG algorithm is still effective. Secondly, we can abandon some available regions and choose a proper shape from the triangle, circle and ellipse to depict the threat region with arbitrary shape.

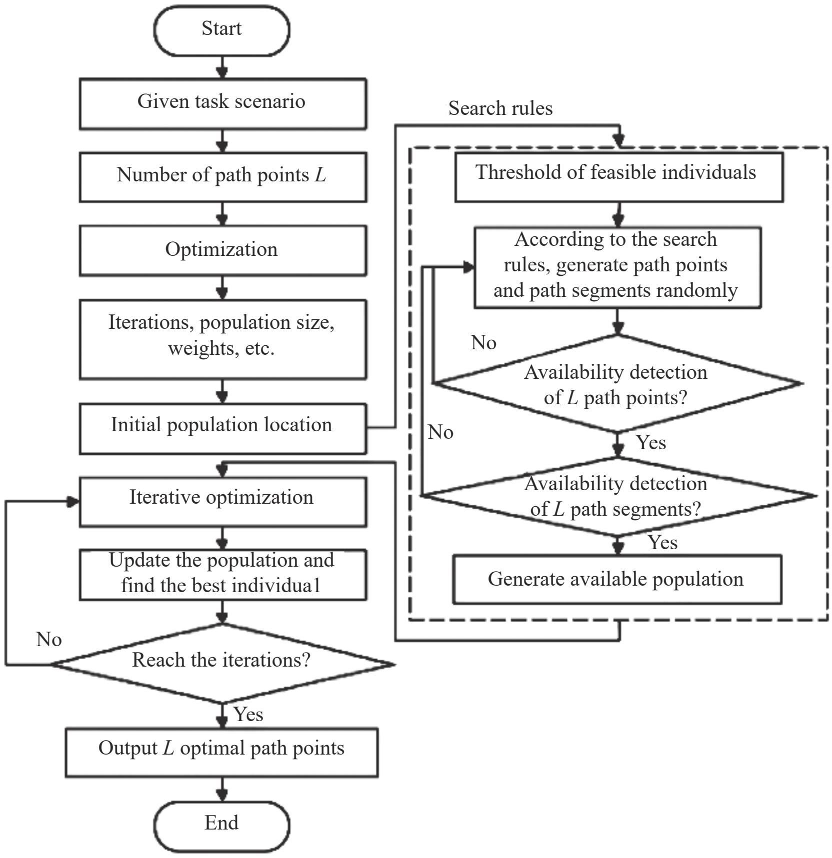


Fig. 9 Flow-chart of SI-PPG algorithm

The fitness function based on the penalty function method is designed above. However, we do not focus on swarm intelligence optimization algorithms in this paper. Here we provide an open field for readers who pay attention to the cooperation task planning, they can apply different swarm intelligence optimization algorithms to combine with the proposed ATPP algorithm, the ATPS algorithm and the fitness function to make a further study. In this paper, the improved PSO algorithm and improved Bats algorithm are respectively combined with the ATPP algorithm, the ATPS algorithm and the fitness function respectively in the experimental section, and relevant simulations will be introduced in Section 6.

5. Cooperative rendezvous strategy generation

5.1 Control strategy generation

The path segments can be obtained based on the proposed SI-PPG algorithm in Section 4, and the Dubins curve is utilized to smooth the flight path of the UAV with the terminal angle constraint and maneuverability constraint. Due to the difference of the Dubins path length and the divergence of flight speed of each heterogeneous UAV, it does not ensure all UAV to arrive at the pre-designated location on time, in this case, UAVs cannot perform the attack task simultaneously which maybe cause the failure of the combat situation suppression task. Consequently, we focus on the rendezvous strategies including speed control strategy and flight path control strategy.

Let the symbol v_{U_i} represent the flight speed of UAV i, and $v_{U_i} \in [v_{U_i,\min}, v_{U_i,\max}]$, $v_{U_i,,\min}$ is the minimal flight speed of UAV i and $v_{U_i,,\max}$ is the maximal flight speed of UAV i. According to the angular rate ω_{U_i}, the turning radius of UAV i can be computed as

$$r_{U_i} \in \left[\frac{v_{U_i,\min}}{\omega_{U_i}}, \frac{v_{U_i,\max}}{\omega_{U_i}}\right].$$

The symbol D_{U_i} represents the Dubins path length of

UAV i from the starting point to the terminal point through L middle path points, the flight time that UAV i reaches the pre-designated location at the maximum speed is recorded as $T_{U_i,\min}$, and $T_{U_i,\min} = D_{U_i}/v_{U_i,\max}$. Then the ETA of the multi-UAVs can be computed by

$$T_{\mathrm{ETA}} = \max\{T_{U_i,\min}\} \tag{19}$$

where T_{ETA} is the maximum of the set of flight time which depicts the ETA of UAVs-swarm.

For UAV i, the flight timeT_{U_i} satisfies the condition, $T_{U_i} \in [T_{U_i,\min}, T_{U_i,\max}]$, $T_{U_i,\min}$ is the minimal flight time and $T_{U_i,\min} = D_{U_i}/v_{U_i,\max}$, $T_{U_i,\max}$ is the maximal flight time and $T_{U_i,\max} = D_{U_i}/v_{U_i,\min}$. The UAV i can realize cooperation rendezvous based on speed control and the flight speed UAV i can be computed by

$$v_{U_i} = \frac{D_{U_i}}{T_{\mathrm{ETA}}},\quad T_{\mathrm{ETA}} \in [T_{U_i,\min}, T_{U_i,\max}]. \tag{20}$$

If $T_{\mathrm{ETA}} \notin [T_{U_i,\min}, T_{U_i,\max}]$, that is, $T_{\mathrm{ETA}} > T_{U_i,\max}$, the speed control is unable to undertake cooperation rendezvous.

5.2 Maneuver strategies realization

5.2.1 Maneuver ways and maneuver times

When the speed control cannot realize the cooperative rendezvous, the flight path control based on local path adjustment is introduced. Here the commonplace ways for local path adjustment include circling maneuver and detouring maneuver. According to the flight time $T_{U_i,\max}$ of UAV i and ETA T_{ETA}, we can get the maneuver time $\Delta t_{U_i} = T_{\mathrm{ETA}} - T_{U_i,\max}$ and maneuver distance $\Delta D_{U_i} = \Delta t_{U_i} v_{U_i}$. The maneuver ways and the maneuver times can be determined according to (21)−(23).

$$k_c = \begin{cases} \left\lceil \dfrac{\Delta D_{U_i}}{2\pi r_{U_i}} \right\rceil = \left\lceil \dfrac{\Delta D_{U_i}\omega_{U_i}}{2\pi v_{U_i}} \right\rceil, & \Delta D_{U_i} \geqslant \sigma \\ 0, & \text{otherwise} \end{cases}, \tag{21}$$

$$r_{U_i} = \begin{cases} \dfrac{\Delta D_{U_i}}{2\pi k_c}, & \Delta D_{U_i} \geqslant \sigma \\ r_{U_i,\min}, & \text{otherwise} \end{cases}, \tag{22}$$

$$k_d = \begin{cases} 1, & \Delta D_{U_i} < \sigma \\ 0, & \text{otherwise} \end{cases}, \tag{23}$$

where $\lceil\cdot\rceil$ is the rounding-up symbol, $v_{U_i} = v_{U_i,\min}$, $r_{U_i} = r_{U_i,\min}$, $r_{U_i} = v_{U_i,\min}/\omega_{U_i}$ is the maneuver radius at the flight speed $v_{U_i,\min}$, σ is a threshold and $\sigma = 4\pi r_{U_i,\max}/5$. According to maneuver distance ΔD_{U_i}, k_c times of circling maneuvers and k_d times of detouring maneuvers can be determined. In addition, (21) and (23) allow us to acquire the decoupling scheme of circling maneuver and detouring maneuver.

5.2.2 Circling maneuver based on ATPP algorithm

According to the maneuver times, the updated rules of flight speed and flight path are given. Let $A_{U_i,S}$ represent the starting point of UAV i, $A_{U_i,E}$ represents the terminal point of UAV i, the number of middle path points is L_{U_i}, the path segments can be denoted as $A_{U_i,S} \to A_{U_i,1} \to \cdots \to A_{U_i,L_{U_i}} \to A_{U_i,E}$, and we can get the Dubins path based on Dubins curve and path segments at the maximal speed of UAV i, which ensures the maneuverability constraint is satisfied during the process of circling maneuver. Here two steps are adopted to realize the circling maneuver.

Step 1 Compute the flight speed of UAV i according to (24).

$$v_{U_i} = \begin{cases} \dfrac{D_{U_i} + 2\pi k_c r_{U_i}}{T_{\mathrm{ETA}}}, & \Delta D_{U_i} \geqslant \sigma \\ v_{U_{i,\min}}, & \text{otherwise} \end{cases} \tag{24}$$

where D_{U_i} is the Dubins path length, ΔD_{U_i} is the maneuver distance.

Step 2 Choose the available circling maneuver point randomly on the Dubins path.

$$J_C = \delta \sum_{j=1}^{M} p(\mathrm{dub}_{A_C A_C}, \mathrm{Thr}_j) \tag{25}$$

where A_C is the circling point, $\mathrm{dub}_{A_C A_C}$ is the circling maneuver path, J_C is the indicator function. The value of J_C is computed based on the ATPP algorithm, if $J_C = 0$, the circling maneuver path is available, if $J_C \neq 0$, the circling maneuver path is unavailable, go back to Step 2, and choose a circling maneuver point repeatedly.

5.2.3 Detouring maneuver based on simplex search method

Aiming at detouring maneuver, two points denoted asB_1 and B_2 are selected randomly on the Dubins path, and a maneuver point which is denoted as B is randomly selected around the points B_1 and B_2. The coordinates and direction angle of maneuver point B can form a vector $\boldsymbol{x}_B$, and $\boldsymbol{x}_B = (x_B, y_B, \theta_B)$, the purpose of detouring maneuver is to minimize (26).

$$F_{\boldsymbol{x}_B} = \min_{\boldsymbol{x}_B} \left\| \mathrm{dub}_{B_1 B} + \mathrm{dub}_{B B_2} - \mathrm{dub}_{B_1 B_2} - \Delta D_{U_i} \right\|^2 \tag{26}$$

where $\mathrm{dub}_{B_1 B}$ depicts the Dubins path from path point B_1 to detouring maneuver point B, $\mathrm{dub}_{B B_2}$ denotes the Dubins path from detouring maneuver point B to path point B_2, $\mathrm{dub}_{B_1 B_2}$ is the original Dubins path from path point B_1 to path point B_2, ΔD_{U_i} represents the detouring maneuver distance.

Obviously, it is an unconstrained optimization problem, simplex search method is adapted to solve the problem (26), then the availability testing for the detouring maneuver path can be done based on the ATPP algorithm.

A simplex is defined as the (n+1)-dimensional convex polyhedron in the n-dimensional space, such as the triangle in two-dimensional space. The basic idea of the simplex search method is to find the function value corresponding to each vertex for the (n+1)-dimensional convex polyhedron in the n-dimensional space, the vertex with the maximal function value is called the highest point, the vertex with the minimal function value is called the lowest point. A new simplex can be acquired based on a better point generated by mathematical operations including reflection, extending and compression, the optimal point can be got by continuous iterations. The general optimization problem can be denoted as

$$F_x = \min_x f(\boldsymbol{x}) \tag{27}$$

where $\boldsymbol{x} \in \mathbf{R}^n$, we can get an ($n$+1)-dimensional simplex based on those points $\boldsymbol{x}^{(i)} \in \mathbf{R}^n$ $(i = 1,2,\cdots,n+1)$.

Let $f(x) = \left\|\mathrm{dub}_{B_1B} + \mathrm{dub}_{BB_2} - \mathrm{dub}_{B_1B_2} - \Delta D_{U_i}\right\|^2$, the search process can be concluded as follows.

Step 1 Select points B_1 and B_2 randomly, and compute the function value $f(\boldsymbol{x}^{(i)})$ $(i = 1,2,\cdots,n+1)$, and let the iteration times k=1.

Step 2 According to (28) and (29), we can find the highest point $\boldsymbol{x}^{(h)}$, secondary high point $\boldsymbol{x}^{(g)}$, and the lowest point$\boldsymbol{x}^{(l)}$, calculate the centroid $\bar{\boldsymbol{x}}$ of n-points besides $\boldsymbol{x}^{(h)}$ and calculate $f(\bar{\boldsymbol{x}})$.

$$\begin{cases} f\left(\boldsymbol{x}^{(h)}\right) = \max\left\{f\left(\boldsymbol{x}^{(1)}\right),\cdots,f\left(\boldsymbol{x}^{(n+1)}\right)\right\} \\ f\left(\boldsymbol{x}^{(l)}\right) = \min\left\{f\left(\boldsymbol{x}^{(1)}\right),\cdots,f\left(\boldsymbol{x}^{(n+1)}\right)\right\} \\ f\left(\boldsymbol{x}^{(g)}\right) = \max\left\{f\left(\boldsymbol{x}^{(i)}\right)\middle|\boldsymbol{x}^{(i)} \neq \boldsymbol{x}^{(l)}\right\} \end{cases} \tag{28}$$

$$\bar{\boldsymbol{x}} = \frac{1}{n}\left[\sum_{i=1}^{n+1}\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h)}\right] \tag{29}$$

Step 3 Compute $\boldsymbol{x}^{(n+2)}$ and $f(\boldsymbol{x}^{(n+2)})$ according to the reflection operation formula (30).

$$\boldsymbol{x}^{(n+2)} = \bar{\boldsymbol{x}} + \alpha\left(\bar{\boldsymbol{x}} - \boldsymbol{x}^{(h)}\right) \tag{30}$$

Step 4 Compute $\boldsymbol{x}^{(n+3)}$ and $f(\boldsymbol{x}^{(n+3)})$ according to the extending operation formula (31).

$$\begin{cases} \boldsymbol{x}^{(n+3)} = \bar{\boldsymbol{x}} + \gamma\left(\boldsymbol{x}^{(n+2)} - \bar{\boldsymbol{x}}\right) \\ \boldsymbol{x}^{(n+4)} = \bar{\boldsymbol{x}} + \beta\left(\boldsymbol{x}^{(h')} - \bar{\boldsymbol{x}}\right) \end{cases} \tag{31}$$

If $f(\boldsymbol{x}^{(n+2)}) < f(\boldsymbol{x}^{(l)})$, compute $f(\boldsymbol{x}^{(n+3)})$ and go to Step 5. If $f(\boldsymbol{x}^{(l)}) \leqslant f(\boldsymbol{x}^{(n+2)}) \leqslant f(\boldsymbol{x}^{(g)})$, let $\boldsymbol{x}^{(h)} = \boldsymbol{x}^{(n+2)}, f(\boldsymbol{x}^{(h)}) = f(\boldsymbol{x}^{(n+2)})$ and go to Step 7. If $f(\boldsymbol{x}^{(n+2)}) > f(\boldsymbol{x}^{(g)})$, let $f(\boldsymbol{x}^{(h')}) = \min\{f(\boldsymbol{x}^{(h)}), f(\boldsymbol{x}^{(n+2)})\}$, $h' \in \{h, n+2\}$, compute $\boldsymbol{x}^{(n+4)}$ and $f(\boldsymbol{x}^{(n+4)})$, and go to Step 6.

Step 5 If $f(\boldsymbol{x}^{(n+3)}) < f(\boldsymbol{x}^{(n+2)})$, let $\boldsymbol{x}^{(h)} = \boldsymbol{x}^{(n+3)}, f(\boldsymbol{x}^{(h)}) = f(\boldsymbol{x}^{(n+3)})$ and go to Step 7, otherwise, let $\boldsymbol{x}^{(h)} = \boldsymbol{x}^{(n+2)}$, $f(\boldsymbol{x}^{(h)}) = f(\boldsymbol{x}^{(n+2)})$ and go to Step 7.

Step 6 If $f(\boldsymbol{x}^{(n+4)}) < f(\boldsymbol{x}^{(h')})$, let $\boldsymbol{x}^{(h)} = \boldsymbol{x}^{(n+4)}$, $f(\boldsymbol{x}^{(h)}) = f(\boldsymbol{x}^{(n+4)})$ and go to Step 7, otherwise, let $\boldsymbol{x}^{(i)} := \boldsymbol{x}^{(i)} + \frac{1}{2}(\boldsymbol{x}^{(l)} - \boldsymbol{x}^{(i)})$ $(i = 1,2,\cdots,n+1)$ and compute $f(\boldsymbol{x}^{(i)})$ $(i = 1,2,\cdots,n+1)$ and go to Step 7.

Step 7 Test whether the convergence criterion is satisfied.

$$\left\{\frac{1}{n+1}\sum_{i=1}^{n+1}\left[f\left(\boldsymbol{x}^{(i)}\right) - f(\bar{\boldsymbol{x}})\right]^2\right\}^{\frac{1}{2}} < \varepsilon \tag{32}$$

If convergence criterion is satisfied, then stop iteration, the optimal solution can be got according to the lowest point, otherwise, let k=k+1， go back to Step 1. In this paper, $\varepsilon = 0.001$.

Step 8 Test whether the detouring maneuver path is satisfied with the availability.

$$J_D = \delta\sum_{j=1}^{M} p(\mathrm{dub}_{B_1B_2}, \mathrm{Thr}_j) \tag{33}$$

Here $\mathrm{dub}_{B_1B_2}$ is the detouring maneuver path, J_D is the indicator function. If $J_D = 0$, the detouring maneuver path is available, if $J_D \neq 0$, the detouring maneuver path is unavailable, go back to Step 1. It is worth noting that the availability testing for the detouring maneuver path can be done based on the ATPP algorithm.

5.3 Cooperative rendezvous strategy generation algorithm

The cooperative rendezvous strategy generation (CRSG) algorithm for multi-UAVs is depicted in Algorithm 3.

Algorithm 3 CRSG algorithm

Input: $D_{U_i}, T_{U_i}, \omega_{U_i}, v_{U_i,\min}, v_{U_i,\max}$

Output: Cooperation rendezvous strategy

1) Compute the ETA of multi-UAVs

2) **If** $T_{\mathrm{ETA}} \in [T_{U_i,\min}, T_{U_i,\max}]$, choose speed control and compute v_{U_i}. **End**

3) **If** $T_{\mathrm{ETA}} > T_{U_i,\max}$, choose flight path control

4) Compute maneuver time Δt_{U_i} and maneuver distance ΔD_{U_i}.

5) Compute the circling maneuver times and detouring maneuver times.

6) **If** $k_c \neq 0$

7) Select the circling maneuver point A_C randomly and generate circling maneuver path.

According to ATPP, test the availability of the circling maneuver path, if $J_C \neq 0$, go to 7), if $J_C = 0$, go to 8).

8) **End**

9) **If** $k_d \neq 0$

10) Select points B_1 and B_2 randomly, and generate detouring maneuver path based on simplex search method.

11) According to ATPP, test the availability of the detouring maneuver path, if $J_D \neq 0$, go to 9), if $J_D = 0$, go to 12).

12) **End**

13) **End**

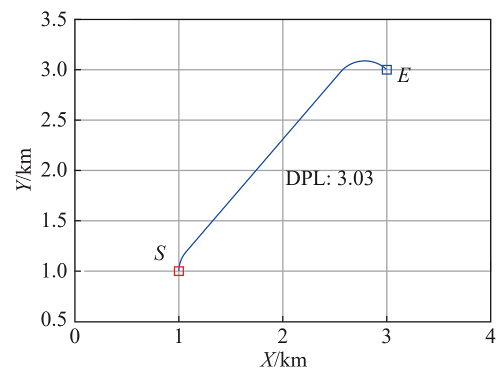
The proposed CRSG algorithm includes three parts, the first part is to determine the control strategies and maneuver ways, the second part is to realize the circling maneuver, and the third part is to realize the detouring maneuver. To be specific, the ETA of multi-UAVs ($T_{\rm ETA}$) is calculated, and the ways to realize rendezvous of each UAV is determined. If the speed control is available, then the flight speed is updated according to (20). If the speed control is unavailable, then the maneuver ways and maneuver times are determined, then circling maneuver and detouring maneuver are respectively designed. The ATPP algorithm is utilized to verify the availability of the generated maneuver path, stop running the maneuver path generation operation until the maneuver path is available.

As for circling maneuver, it is worth noting that the availability testing for the circling maneuver path can be done based on the ATPP algorithm. In the implementation process, the maneuver point should satisfy two conditions: the point is in the Dubins path; there is no threats at this point within the circling maneuver radius. Because the Dubins path is known and the ATPP algorithm can be applied to judge the availability of the circling maneuver path, in this case, we can get the circling maneuver path rapidly. As for detouring maneuver, in order to ensure the efficiency of path planning, when we choose point B_1 and point B_2, the Dubins path length between the two points is far greater than detouring maneuver distance, in this case, maneuver purpose is realized by adjusting the path slightly and wide range maneuvering is avoided, consequently, we can effectively reduce the probability of conflict with threat region.

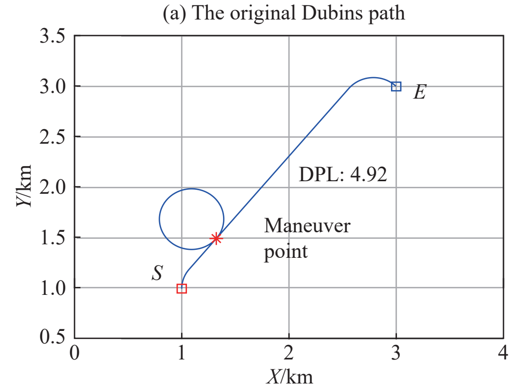
5.4 Maneuver strategy realization

A case is given in this section in order to demonstrate the effectiveness of the proposed circling maneuver strategy and detouring maneuver strategy. Let the turning radius of UAV is 0.3 km, that is, $r_{U_i} = 0.3$ km, which denotes the maneuverability constraint. This UAV takes off from the starting point (1,1) to the terminal point (3,3), the starting angle is $\pi/2$ and the terminal angle is $-\pi/4$. The basic purpose is to realize once circling maneuver and once detouring maneuver respectively, the increasing distance after once circling maneuver is equal to $2\pi r_{U_i}$ km, the increasing distance after once detouring maneuver is 0.8 km which is pre-designated.

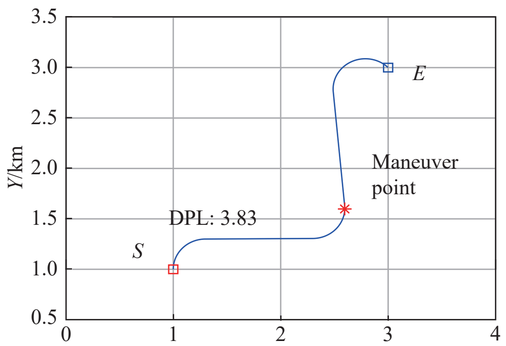
The results without threat existing in the scenario are depicted in Fig. 10, to be specific, the original Dubins path can be depicted in Fig. 10(a), it is obvious that the Dubins path from starting point to terminal point satisfies the maneuverability constraint and terminal angle constraint. The Dubins path after circling maneuver is depicted in Fig. 10(b). The DPL after circling maneuver is 4.92 km, and the circling maneuver distance is 1.89 km which is almost equal to $2\pi r_{U_i}$. The Dubins path after detouring maneuver is depicted in Fig. 10(c), the DPL after detouring maneuver is 3.83 km which meets the predesignated requirement.



(a) The original Dubins path



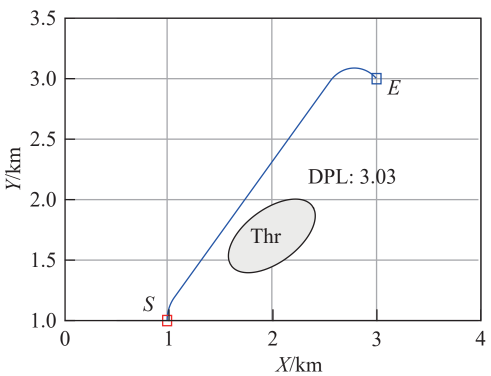
(b) Dubins path after circling



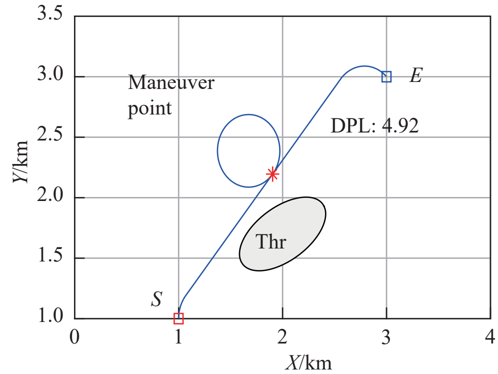
(c) Dubins path after detouring

Fig. 10 Dubnis path without threat

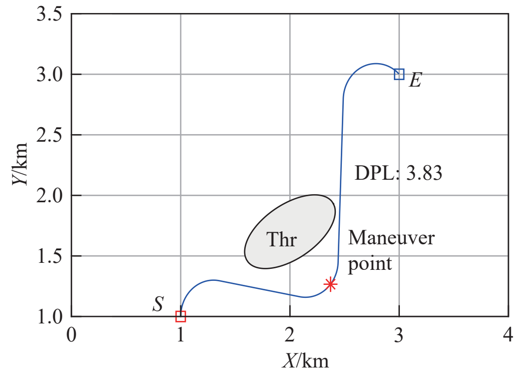
The results with a threat existing in the scenario are depicted in Fig. 11, the threat is an ellipse-shape whose azimuth angle is $\pi/6$. Comparing Fig. 10(b) and Fig. 11(b), the maneuver points for circling are different due to the random selection rule given in Subsection 5.2.2, and the DPL depicted in the two figures are the same, which proves the effectiveness of the proposed circling maneuver strategy. Observing Fig. 10(c) and Fig. 11(c), it is noticeable that the maneuver results for the scenario with threat and the scenario without threat are dissimilar. As depicted in Fig. 10(c), the detouring path is not available for the next scenario with threat according to the ATPP algorithm, the maneuver point will be generated again based on the simplex search method until the detouring maneuver path is satisfied with the availability requirement. From the result depicted in Fig. 11(c), the detouring path can meet the maneuverability constraint, terminal angle constraint, availability constraint and maneuver distance constraint.



(a) The original Dubins path



(b) Dubins path after circling



(c) Dubins path after detouring

Fig. 11 Dubnis path with threat

6. Experiments and results analysis

6.1 Simulation experiment of algorithm effectiveness

The proposed IBats-based PPG and IPSO-based PPG algorithms, the RRT algorithm [22], the bidirectional extended RRT (BERRT) algorithm [37], the self-adaptive step-RRT (SAS-RRT) algorithm [38] and the RRT star algorithm [26] are utilized to solve the path planning problem, the scenario size is 40 km × 40 km. The population size of IBats and IPSO algorithms is 200, the maximum iteration times is 100, the search step-size of RRT, BERRT and RRT star is 0.15 km, that is, 15 TM (Ten meter), the minimal search step-size of SAS-RRT is 0.15 km, the maximal search step-size of SAS-RRT is 0.3 km, the maximum iteration of RRT, BERRT, SAS-RRT and RRT star algorithm is 500, the search accuracy is 0.2 km.

Thirty simulation experiments are carried out, and the statistical results are shown in Table 3. In terms of path length and number of nodes, the proposed IBats-based PPG algorithm and the IPSO-based PPG algorithm have the advantages over the RRT, BERRT, SAS-RRT and RRT star algorithms. In terms of simulation time, the proposed IBats-based PPG algorithm has the advantages over the RRT, RRT star and IPSO-based PPG algorithms. Although the search efficiency of BERRT and SAS-RRT algorithms are better than IBats-based PPG and IPSO-based PPG algorithm, but the path length acquired by BERRT and SAS-RRT algorithms are not the shortest. The path planning results are shown in Fig. 12, it is obvious that the proposed IBats-based PPG algorithm and the IPSO-based PPG algorithm can get the almost same planning results, compared with the classical RRT, BERRT, SAS-RRT and RRT star algorithm, the proposed algorithms can get the shortest path.

Table 3 Statistical results

Algorithm	Path length/km	Number of nodes	Simulation time/s
IBats-based PPG	43.54	1	3.16
IPSO-based PPG	43.22	1	4.87
RRT	61.58	38	3.24
BERRT	61.11	41	2.77
SAS-RRT	62.53	36	2.86
RRT star	45.93	8	5.19

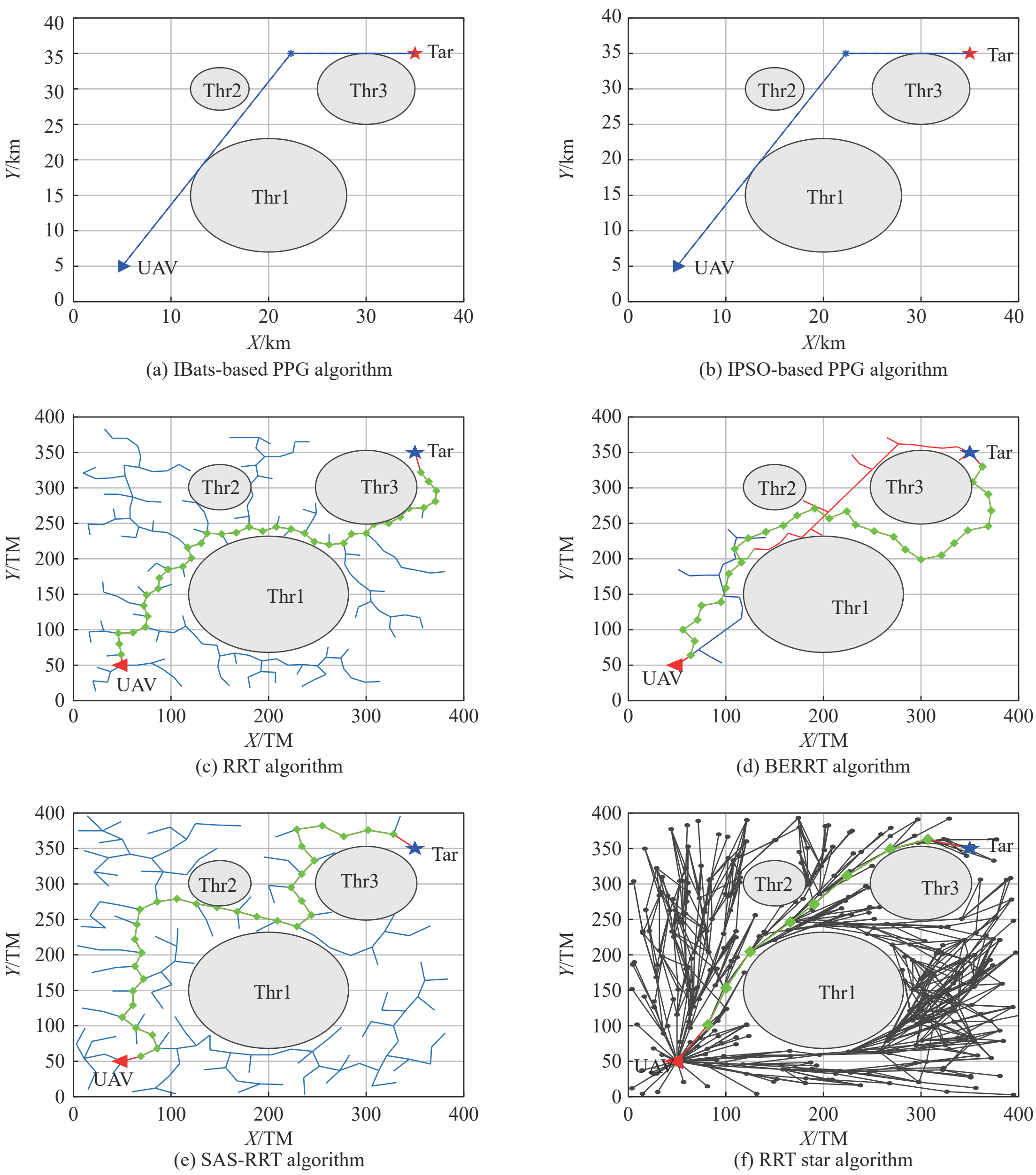


Fig. 12 Path generation

As a conclusion, the swarm intelligence-based path point generation idea is effective, here both IBats algorithm and IPSO algorithm are applied to solve the path planning problem, and the shortest path is obtained within acceptable computation time. It is worth noting that this paper provides a scalable platform, interested readers can apply other swarm intelligence algorithms to generate the path of UAV.

6.2 Simulation for SEAD task

6.2.1 Parameters description of SEAD task

As depicted in Fig. 13, the SEAD task scenario is a square area whose size is 40 km × 40 km. There are four targets, eight UAVs and 20 threats scattered in this scenario. The basic attributes of the four targets are given in Table 4. The attributes of the eight-UAVs are given in Table 5. The fundamental parameters of the 20 threats are given in Table 6. The SEAD task assignment solution is pre-fixed in Table 7, U1 and U2 are used to attack Tar1, U3 and U4 are used to attack Tar2, U5, U6 and U7 are used to attack Tar3, U8 are used to attack Tar 4.

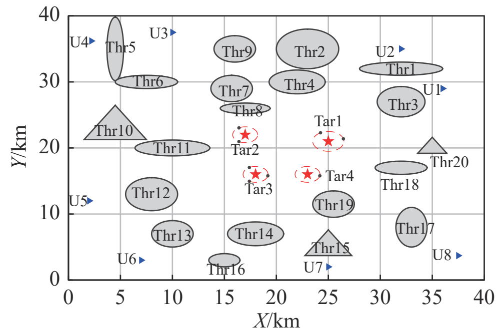


Fig. 13 SEAD task scenario

Table 4 Parameters of targets

Parameter	Target code			
	Target1	Target2	Target3	Target4
Location/km	(25,21)	(17,22)	(17.9,16.1)	(23,16)
Attack radius/km	1.5	1.2	1.2	1.2
Requirement	2	2	3	1

Table 5 Parameters of UAVs

Parameter	UAV code							
	U1	U2	U3	U4	U5	U6	U7	U8
Initial location/km	(36,29)	(32,35)	(10,37.5)	(2.2,36.2)	(2,12)	(7,3)	(25,2)	(37.5,3.7)
Turning rate/(rad/s)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Speed range/(m/s)	[50,80]	[40,70]	[65,85]	[40,80]	[45,75]	[45,75]	[35,70]	[45,75]

Table 6 Parameters of threats

Parameter	Threat code						
	Thr1	Thr2	Thr3	Thr4	Thr5	Thr6	Thr7
Location/km	(32,32)	(23,35)	(32,27)	(22,30)	(4.5,35)	(7.5,30)	(15.7,29)
Shape code	3	2	2	2	3	3	2
Size/km	1/4	3	2.3	1.8/2.7	0.8/4.8	1/3	2
Azimuth/(°)	0	90	0	0	90	0	60
Parameter	Threat code						
	Thr8	Thr9	Thr10	Thr11	Thr12	Thr13	Thr14
Location/km	(17,26)	(16,35)	(4.5,23)	(10,20)	(8,13)	(10,7)	(18,7)
Shape code	3	2	1	3	2	2	3
Size/km	0.8/2.4	2	6	1.2/3.6	2.5	2	1.8/2.7
Azimuth/(°)	0	0	0	0	45	0	0
Parameter	Threat code						
	Thr15	Thr16	Thr17	Thr18	Thr19	Thr20	
Location/km	(25,5)	(15,3)	(33,8)	(32,17)	(25.5,11.5)	(25,20)	
Shape code	1	3	3	3	2	1	
Size/km	4.5	1/1.5	1.5/3	1/2.5	2	2.8	
Azimuth/(°)	0	0	90	0	−40	120	

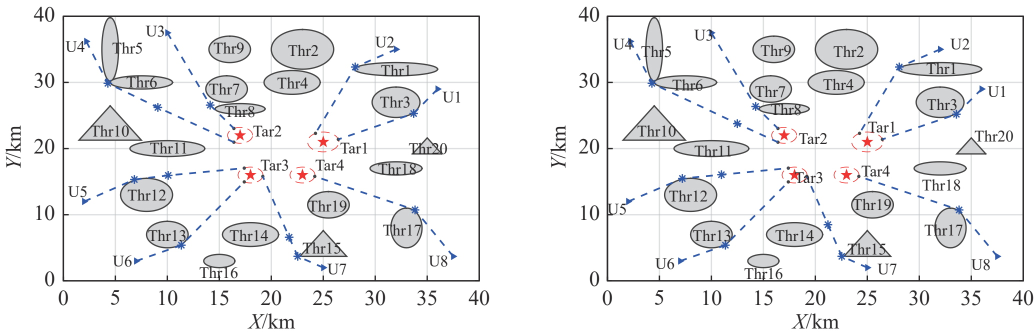
Table 7 Task assignment

UAV code	Execution target	Location/km	Terminal angel/(°)
U1	Tar1	(26.4,21.4)	15
U2	Tar1	(24.3,22.3)	120
U3	Tar2	(16.4,23)	120
U4	Tar2	(16.4,21)	−120
U5	Tar3	(17.4,17)	120
U6	Tar3	(17.4,15)	−120
U7	Tar3	(19.2,15.8)	−10
U8	Tar4	(24.2,15.8)	−10

The combat purpose of the 8-UAVs is to realize the combat situation suppression against the hostile 4-targets based on the cooperative rendezvous, in other words, the common goal of UAVs swarm is to arrive the respective pre-designated locations at the same time in order to perform synchronous attack task. The next experiments including path points generation, path smoothing and cooperative rendezvous are conducted and relevant results are analyzed in detail to prove the effectiveness of our works.

6.2.2 PPG simulation

In order to prove the effectiveness of the proposed SI-PPG algorithm, here IBats-PPG algorithm and IPSO-PPG algorithm are implemented to generate the shortest paths for the 8-UAVs respectively. The experimental result based on the IBats-PPG algorithm is depicted in Fig. 14(a), and the experimental result based on the IPSO-PPG algorithm is depicted in Fig. 14(b), it is clear that the planned paths for the 8-UAVs are available, consequently, we can get a conclusion that both IBats-PPG algorithm and IPSO-PPG algorithm can be used to solve the path planning problem.



(a) IBats-PPG algorithm (b) IPSO-PPG algorithm

Fig. 14 Generated path segments

Thirty experiments are conducted to calculate the statistical average results of fitness value and simulation time, the results are depicted in Fig. 15 and Fig. 16. The fitness value stands for the path length, according to Fig. 15, we can get the optimal search results based on the IBats-PPG algorithm and IPSO-PPG algorithm. Obviously, the mean simulation time of IBats-PPG algorithm and IPSO-PPG algorithm are distinctive according to Fig. 16, the simulation time of IBats-PPG algorithm is less than that of IPSO-PPG algorithm, consequently the IBats-PPG algorithm has a wider application especially in the high real-time occasion compared with the IPSO-PPG algorithm. In addition, IBats and IPSO algorithm are applied successfully, which indicates that the proposed SI-PPG algorithm is extendable. Interested researchers can try different swarm intelligence optimization algorithms, such as ACO and GA.

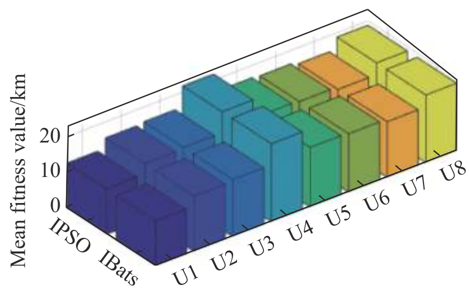


Fig. 15 Mean fitness value

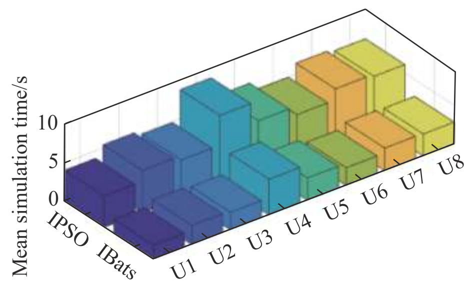


Fig. 16 Mean simulation time

6.2.3 Path smoothing simulation

According to the path planning results generated by IBats-PPG algorithm and IPSO-PPG algorithm, and the terminal angle constraint and maneuverability constraint are taken into consideration, the smoothed paths based on Dubins curve are presented in Fig. 17 and Fig. 18.

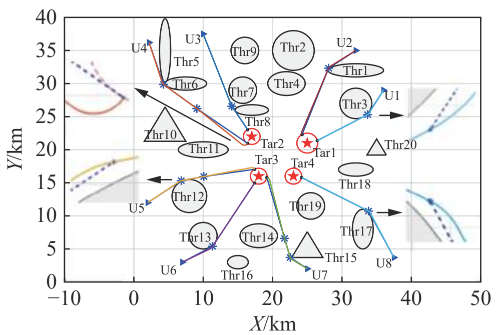


Fig. 17 Smoothed path for the path segments generated by IBats-PPG algorithm

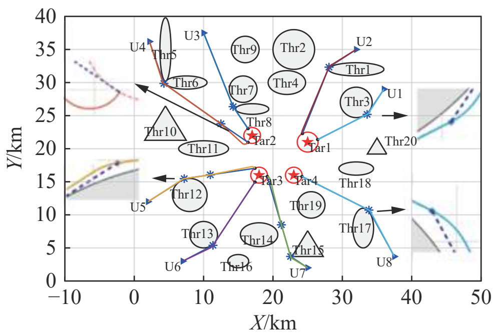


Fig. 18 Smoothed path for the path segments generated by IPSO-PPG algorithm

Focusing on U1, U8 and U5, the Dubins paths of these three UAVs are enlarged in the vicinity of threats (Thr3, Thr12, Thr17), it is distinct that the smoothed flyable paths satisfy threat avoidance constraint. The enlarged figure of U4's flight path nearby the Tar2 depicts that, based on Dubins curve, we can effectively deal with the terminal angle constraint and maneuverability constraint and get the flyable flight path.

6.2.4 Cooperative rendezvous simulation

Here the paths generated by IBats-PPG algorithm are used for cooperative rendezvous simulation in order to reduce the unnecessary repetition. The cooperative rendezvous messages including maneuver ways and maneuver times are calculated and depicted in Table 8. The DPL of each UAV listed in Table 8 is computed at the maximal flight speed, which ensures the maneuverability constraint can be satisfied.

Table 8 Cooperation rendezvous information

Item	U1	U2	U3	U4	U5	U6	U7	U8
DPL/km	12.615	15.591	15.885	22.328	16.797	16.302	15.744	18.855
Control way	Path	Speed	Path	Speed	Speed	Speed	Speed	Speed
Maneuver distance/km	1.340	–	2.26	–	–	–	–	–
Maneuver strategy	Detouring	–	Circling	–	–	–	–	–
Maneuver times	1	–	1	–	–	–	–	–

According to the DPL and flight speed ranges of UAVs, the flight time region, the maximal flight time, the minimal flight time and ETA of UAVs swarm can be calculated, which are shown in Fig. 19. It is obvious that the minimal flight time of U4 is regarded as the ETA of UAVs swarm, which is 279.101 s. In addition, the flight time regions of U2, U5, U6, U7 and U8 contain the ETA, consequently, the cooperative rendezvous of these UAVs can be realized based on speed control. Even though U1 and U3 fly at the minimal speed, their flight time is less than the ETA, so the cooperative rendezvous of U1 and U3 cannot be realized based on speed control, which indicates that realizing complex and complicated rendezvous task based on speed control is limited because the flight speed range of UAV is bounded. This is the reason that why we study both speed control and flight path control to achieve cooperative rendezvous in this paper.

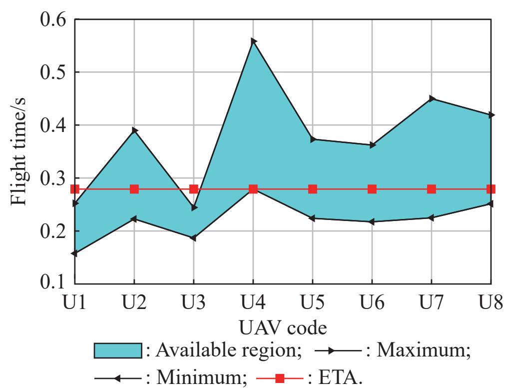


Fig. 19 Flight time and ETA

Here U1 and U3 should adopt flight path control strategy to realize the rendezvous goal. To be specific, the maneuver distance of U1 is 1.34 km and the fixed parameter σ is easy to get, $\sigma = 4\pi r_{U_i,\max}/5 = 2.01$ km, the detouring maneuver times of U1 can be determined, that is to say, U1 can realize the rendezvous purpose based on once detouring maneuver. In terms of U3, the maneuver distance of U3 is 2.26 km and $\sigma = 4\pi r_{U_i,\max}/5 = 2.14$ km, the circling maneuver times is acquired according to (21) and (22), U3 can realize the rendezvous purpose based on once circling maneuver.

After the cooperation rendezvous, the flight speed of each UAV is shown in Fig. 20, the DLPs before the rendezvous and after the rendezvous are depicted in Fig. 21. It is clear that the flight speed of each UAV meets the flight speed range constraint, to be specific, the flight speed of U4 is 80 m/s while the DLP of U4 is 22.328 km, so the flight time of U4 is 279.101 s which is regarded as the ETA of multi-UAVs. The flight speed of U1 keeping the minimum and detouring maneuver is applied to increase the flight distance in order to realize the cooperative rendezvous. The actual increased flight distance of U3 is 5.341 km, which is the product of k_d and $2\pi r_{U_i,\max}$, consequently, the flight speed of U3 is updated after circling maneuver according to (24), and the flight speed of U3 satisfies the flight speed constraint.

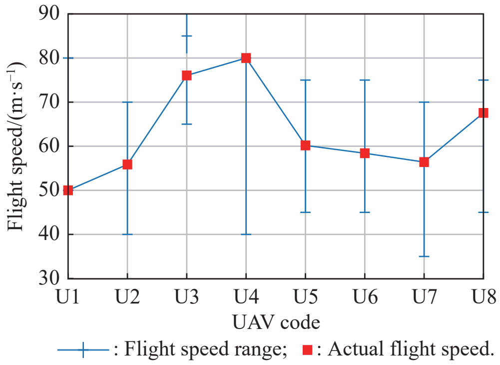


Fig. 20 Actual flight speed of each UAV

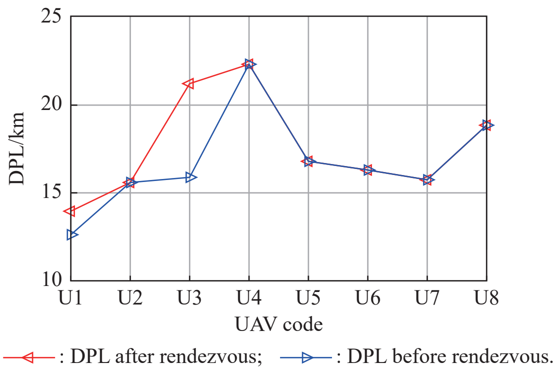


Fig. 21 DPLs before rendezvous design and after rendezvous design

The flight path of each UAV after the design of cooperative rendezvous is shown in Fig. 22, the flight paths of U2, U4, U5, U6, U7 and U8 depicted in Fig. 22 are exactly the same with that depicted in Fig. 17, the reason is that those UAVs can realize cooperation rendezvous based on speed control. The flight path of U1 depicted in Fig. 22 is obviously different with that depicted in Fig. 17 due to the detouring maneuver. Similarly, the flight path of U3 depicted in Fig. 22 is obviously different with that depicted in Fig. 17 due to the circling maneuver.

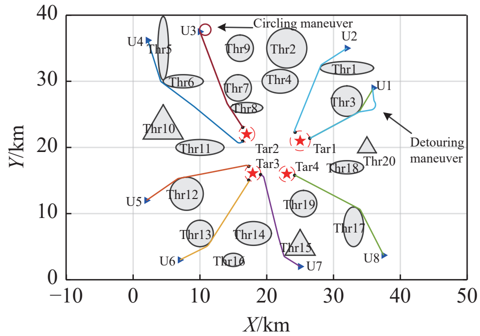


Fig. 22 Flight path after maneuver

The absolute value of the difference between the ETA of each UAV before rendezvous and that after rendezvous is called absolute ETA error, which is shown in Fig. 23. The maximal value of the absolute ETA error is 0.0025 s, which is negligible. In other words, all UAVs can arrive at the pre-designated locations almost at the same time, both speed control strategy and flight path control strategy can support the purpose of realizing combat situation suppression based on the cooperative rendezvous. In addition, the proposed decoupling scheme of circling maneuver and detouring maneuver is proved to be practicable.

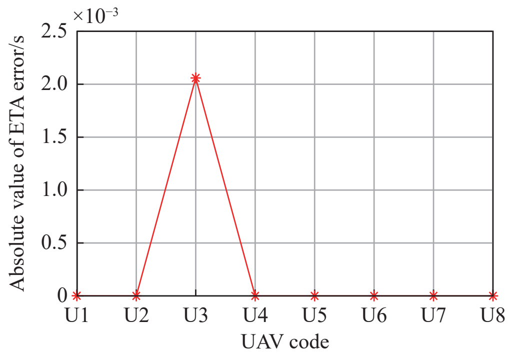


Fig. 23 Absolute ETA error

7. Conclusions

To realize situation suppression against the enemy, the multi-UAVs cooperative path planning problems are studied in this paper, the major conclusions can be summarized as follows.

(i) The designed SI-PPG algorithm can effectively solve the path planning problem in continuous scenarios, to be specific, the IBats-PPG algorithm and IPSO-PPG algorithm are designed to generate the path points for SEAD tasks. That is to say, the proposed SI-PPG algorithm has wide scalability, interested scholars can combine different swarm intelligence optimization algorithms with the proposed algorithm to deal with path points generation problem.

(ii) The flyable flight path based on Dubins curve can be acquired, which can be effectively utilized to deal with the terminal angle constraint and maneuverability constraint. The acquired paths allow UAVs to fly in actual combat scenarios, which increases the realizability of the planned scheme.

(iii) The flight speed control strategy and flight path control strategy are introduced, and the decoupling scheme of circling maneuver and detouring maneuver is designed, the simulation results for SEAD tasks indicate that the cooperative rendezvous strategy designed in this paper can realize the situation suppression against the enemy.

This paper defines three threat regions, including equilateral triangle, circle and ellipse, which have some limitations. In the follow-up research, we should consider how to avoid the threat region of arbitrary shape.

References

[1] HUANG H, WU K, WANG H F, et al. Path planning of UAV low altitude penetration based on improved moth-flame optimization. Journal of Chinese Inertial Technology, 2021, 29(2): 256–263. (in Chinese)

[2] MA C Y, FENG Z Q, ZHENG X M. Development of bionic UAVs cluster technology. Transactions of Nanjing University of Aeronautics and Astronautics, 2018, 35(1): 1–8.

[3] KHAN A, AFTAB F, ZHANG Z S. UAPM: an urgency aware packet management for disaster management using flying ad-hoc networks. China Communications, 2019, 16(11): 167–182.

[4] YANG Q Q, GAO Y Y, GUO Y, et al. Target search path planning for naval battle field based on deep reinforcement learning. Systems Engineering and Electronics, 2022, 44(11): 3486−3495. (in Chinese)

[5] YAN Z P, BAI R, CHI D N, et al. Formation coordination control of multi-UUV for object searching. Computer Measurement & Control, 2013, 21(6): 1532–1536. (in Chinese)

[6] JAVIER A M, BAKER S, RUS D. Multi-robot formation control and object transport in dynamic environments via constrained optimization. The International Journal of Robotics Research, 2017, 36(9): 1000–1021.

[7] MAZA I, CABALLERO F, CAPITAN J, et al. Experimental results in multi-UAV coordination for disaster management and civil security applications. Journal of Intelligent and Robotic Systems, 2011, 61(1/4): 563–585.

[8] WU W N, GUAN Y Z, GUO J F, et al. Research on cooperative task assignment method used to the mission SEAD with real constraints. Control and Decision, 2017, 32(9): 1574–1582. (in Chinese)

[9] XIANG X J, YAN C, WANG C, et al. Coordination control method for fixed-wing UAV formation through deep reinforcement learning. Acta Aeronautica et Astronautica Sinica, 2021, 42(4): 420–433. (in Chinese)

[10] TSOURDOS A, WHITE B, SHANMUGAVEL M, et al. Cooperative path planning of unmanned aerial vehicles. ZHOU Z, WANG X. trans. Beijing: National Defence Industry Press, 2013. (in Chinese)

[11] LI X Q, MA R, ZHANG S, et al. Improved design of ant colony algorithm and its application in path planning. Acta Aeronautica et Astronautica Sinica, 2020, 41(S2): 213–219. (in Chinese)

[12] WU W H, GUO X F, ZHOU S Y. Dynamic route planning based on improved constrained differential evolution algorithm. Control and Decision, 2020, 35(10): 2381–2390. (in Chinese)

[13] HU L, YI G X, HUANG C, et al. Research on dynamic weapon target assignment based on cross-entropy. Mathematical Problems in Engineering, 2020, 2020: 1–13.

[14] SUN X L. Research on mission planning for unmanned aerial vehicles based on multi-stage path prediction. Harbin: Harbin Institute of Technology, 2015.

[15] WU K, TAN S C. Path planning of UAVs based on improved whale optimization algorithm. Acta Aeronautica et Astronautica Sinica, 2020, 41(2): 107–114. (in Chinese)

[16] ZHEN Z Y, ZHU P, XUE Y X, et al. Distributed intelligent self-organized mission planning of multi-UAV for dynamic targets cooperative search-attack. Chinese Journal of Aeronautics, 2019, 32(12): 2706–2716.

[17] ZHEN Z Y, XING D J, GAO C. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. Aerospace Science & Technology, 2018, 76: 402–411.

[18] PANG Q W, HU Y J, LI W G. Path planning algorithm for multi-UAVs cooperative reconnaissance on multiple targets. Journal of Chinese Inertial Technology, 2019, 27(3): 340–348. (in Chinese)

[19] CHEN Y B, YU J Q, MEI Y, et al. Modified central force optimization (MCFO) algorithm for 3D UAV path planning. Neurocomputing, 2016, 171(1): 878–888.

[20] LI W G, HU Y J, PANG Q W, et al. Track planning of multi-UAV cooperative reconnaissance based on improved genetic algorithm. Journal of Chinese Inertial Technology, 2020, 28(2): 248–255. (in Chinese)

[21] LI W G, SUN S Y, LI J Z, et al. UAV dynamic path planning algorithm based on segmentated optimization RRT. Systems Engineering and Electronics, 2018, 40(8): 1786–1793. (in Chinese)

[22] LAVALLE S M, KUFFNER J J. Randomized kino-dynamic planning. International Journal of Robotics Research, 1999, 15(5): 378−400.

[23] KOTHARI M, POSTLETHWAITE I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. Journal of Intelligent and Robotic Systems, 2013, 71(2): 231–253.

[24] PERSSON S M, SHARF I. Sampling-based A* algorithm for robot path-planning. The International Journal of Robotics Research, 2014, 33(13): 1683−1708.

[25] SINGH Y, SHARMA S, SUTTON R, et al. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering, 2018, 168(1): 187−201.

[26] KARAMEN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 2011, 30(7): 846–894.

[27] JEONG I B, LEE S J, KIM J H. Quick-RRT*: triangular inequality-based implementation of RRT* with improved ini-

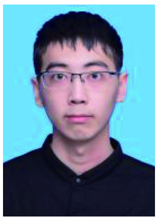
tial solution and convergence rate. Expert Systems with Applications, 2019, 123: 82–90.
[28] OH H, KIM S, SHIN H S, et al. Rendezvous and standoff target tracking guidance using differential geometry. Journal of Intelligent and Robotic Systems, 2013, 69(4): 389–405.
[29] MANATHARA J G, GHOSE D. Rendezvous of multiple UAVs with collision avoidance using consensus. Journal of Aerospace Engineering, 2012, 25(4): 480–489.
[30] MCLAIN T W, BEARD R W. Coordination variables, coordination functions, and cooperative timing missions. Journal of Guidance, Control & Dynamics, 2005, 28(1): 150–161.
[31] SHAN W Z, CUI N G, HUANG B, et al. Multiple UAV cooperative path planning based on PSO-HJ method. Journal of Chinese Inertial Technology, 2020, 28(1): 122–128. (in Chinese)
[32] DUAN H B, ZHANG X Y, WU J. Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments. Journal of Bionic Engineering, 2009, 6(2): 161–173.
[33] SUN X L, MENG Y L, QI N M, et al. Cooperative path planning for rendezvous of unmanned aerial vehicles. Robot, 2015, 37(5): 621–627.
[34] FOSSEN T I, PETTERSEN K Y, GALEAZZI R. Line-of-sight path following for Dubins paths with adaptive sideslip compensation of drift forces. IEEE Trans. on Control Systems Technology, 2015, 23(2): 820–827.
[35] DUBINS L E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics, 1957, 79: 497–516.
[36] HU L. Research on combat effectiveness algorithm of flying weapon under dynamic condition. Harbin: Harbin Institute of Technology, 2018. (in Chinese)
[37] KUFFNER J J, LAVALLE S M. RRT-connect: an efficient approach to single-query path planning. Proc. of the IEEE International Conference on Robotics and Automation, 2000: 995−1001.
[38] LI Y, XU D, ZHOU C. Cooperation path planning of dual-robot based on self-adaptive stepsize RRT. Transactions of the Chinese Society of Agricultural Machinery, 2019, 50(3): 358–367. (in Chinese)

Biographies

HU Lei was born in 1993. He received his M.S. degree in control science and engineering, School of Astronautics, Harbin Institute of Technology in 2018. Currently, he is pursuing his Ph.D. degree in Harbin Institute of Technology. His research interests include artificial intelligence, UAV cluster, weapon system combat effectiveness, and decision-making.
E-mail: maple_hsjz@163.com

YI Guoxing was born in 1974. He received his Ph.D. degree in control science and engineering, School of Astronautics, Harbin Institute of Technology. His research interests include UAV system and application technology, the mechanism and application of hemispherical resonance gyro, and the research of inertial and integrated navigation.
E-mail: ygx@hit.edu.cn

NAN Yi was born in 1989. She received her M.S. degree in control science and engineering, School of Astronautics, Harbin Institute of Technology. Her research interests include operational effectiveness, decision-making, and reinforcement learning.
E-mail: nanyi11@163.com



WANG Hao was born in 1996. He received his M.S. degree in control science and engineering, School of Astronautics, Harbin Institute of Technology. Currently, he is pursuing his Ph.D. degree in Harbin Institute of Technology. His research interests include artificial intelligence, missile swarm, combat effectiveness evaluation, decision-making and reinforcement learning.
E-mail: 21B904060@stu.hit.edu.cn