Journal of Intelligent & Robotic Systems (2020) 100:1383–1400
https://doi.org/10.1007/s10846-020-01241-w

Cooperative Task Assignment and Track Planning For Multi-UAV Attack Mobile Targets

Chen Xia[1] · **Liu Yongtai**[1] · **Yin Liyuan**[1] · **Qi Lijie**[1]

Received: 10 July 2019 / Accepted: 22 July 2020 / Published online: 8 August 2020

Abstract

This paper proposes a system framework for solving the problem of multi-UAV cooperative task assignment and track planning for ground moving targets. For the combinatorial optimization model, it is solved by a new particle swarm optimization algorithm based on guidance mechanism. In order to plan an effective track for the target more rapidly, a new ant colony optimization algorithm based on adaptive parameter adjustment and bidirectional search is proposed. Furthermore, in the case of target movement, a method of the predicted meeting point is proposed to solve the problem that the moving point cannot be used as the target point of the track planning algorithm. In addition, the track planning problem in the UAV tracking mode is also considered. An online re-planning method is proposed for time-sensitive uncertainties. Finally,the simulation results show that compared with other algorithms, the proposed method can not only effectively plan a reasonable track, but also solve the uncertainty problem, and obtain the optimal task allocation plan, which improves the multi-UAV cooperative combat capability.

Keywords Moving target · Cooperative task assignment · Track planning · Ant colony optimization · Particle swarm optimization

1 Introduction

Multiple unmanned aerial vehicles(Multi-UAV) cooperative task assignment and track planning helps to make a mission execution plan for the targets within a certain decision-making time, so as to obtain the maximum mission benefit at the minimum system cost. UAVs are gradually replacing manned aerial vehicles to perform various complex tasks, including response, tracking and attack.For the multi-UAV cooperative track planning and task assignment, the optimal feasible track of each UAV should be considered, as well as space and time coordination, so that the final assignment scheme meets the requirements of global optimization.

The multi-UAV cooperative task assignment problem can be reduced to a complex combinatorial optimization problem [1] considering task sequence, time constraints, environmental changes and track feasibility. Multi-UAV collaborative task assignment does not have a fixed formation structure, but aims at achieving optimal efficiency, which is different from formation control [2]. Different combinations of UAVs will affect the final mission completion effect. Therefore, the increase of the number of targets and UAVs will expand the scale of the problem. When the target is in a moving state at all times, the uncertainty of this problem will make the battlefield situation more complex. It is very difficult to establish an accurate mathematical model and find an appropriate algorithm for this problem, which is one of the most challenging problems in multi-UAV cooperative task assignment.

Mixed integer linear programming model(MILP) [4] and Dynamic network flow optimization model [5] are widely used to solve UAV task assignment problem. In addition, Alighanbari and How described this problem as a dynamic programming problem, which is more simple in calculation than the MILP model [6]. There is no absolute distinction of advantages and disadvantages between the different models, but the MILP model adopted in this paper has certain versatility.From the perspective of model solving, it is difficult to find the optimal solution for

✉ Liu Yongtai
416599803@qq.com

Chen Xia
xiachen1108@163.com

[1] Shenyang Aerospace University, Shenyang, China

the discrete combinatorial optimization problem. However, some scholars have proposed some algorithms to solve such problems.The main solving algorithms are divided into two categories, distributed algorithms and centralized algorithms. The distributed algorithm has good robustness. When there is a failure with a UAV, it does not affect the actions of other UAVs.This kind of algorithms include Negotiation scheme based on the contract net [7] and Market mechanism [8]. However, the optimization of the results cannot be guaranteed while avoiding conflicts between UAVs. Classical centralized algorithms include Hungarian algorithm[3], genetic algorithm [9], particle swarm optimization algorithm [10], ant colony algorithm [11]and state space priority algorithm [12]. This kind of algorithm has the ability to find the global optimal solution, it is suitable for solving small and medium-sized problems.

Track planning is an important part of task assignment.The least cost path between UAVs and targets can be planned, which affects the result of task assignment. It is possible to quickly plan a reasonable track in a complex environment or even a target movement, which determines the quality of the task. The track planning algorithm for UAVs has also been extensively studied [13–15]. Literature [13] proposed a two-stage optimization iterative algorithm to solve the problem of decentralized cooperative track planning for multiple UAVs. Literature [14] introduces the concepts of “virtual velocity rigid body” and “virtual target point” for path planning problems of multiple UAV formations in known and realistic environments. Literature [15] proposed a two-layer planning scheme, which divides the UAV collaborative attack path planning into a path planning layer and a collaborative planning layer, and determined the candidate path through the ant colony optimization algorithm (ACO). The UAV is also used to perform tracking tasks [16–18]. Literature [16] proposed a method of planning the tracking path according to the target moving state, and finally keeping the UAV in the circle centered on the target. In [17], a UAV online track planning algorithm based on the Pythagoras Hodograph curve was proposed. The position and cross heading error equations are established to improve the tracking accuracy. In [18], the task assignment and the track planning are considered comprehensively, and a cross-based path generation algorithm and a negotiation-based task assignment algorithm were proposed. These algorithms are capable of regenerating a viable path online at a small computational cost to enable the UAV to perform update tasks in literature [18].

Inspired by all mentioned above, this paper concentrates on the cooperative task assignment and track planning problem of multi-UAV attack ground moving targets. Firstly, a multi-UAV cooperative task assignment optimization model for moving targets is established. Then, a new ant colony optimization algorithm based on adaptive parameter adjustment and bidirectional search (BSAP-ACO) is proposed, which can quickly plan the feasible track of UAV. On this basis, two methods for calculating the predicted meeting point and on-line track re-planning are proposed. The methods solve the problem that the moving point cannot be directly used as the target point of the track planning algorithm. In addition, the flight path in the tracking mode is also planned according to the speed ratio. An online task reassignment method is proposed to solve the problem of time-sensitive uncertainty. Finally, an improved particle swarm optimization algorithm based on guidance mechanism (GMPSO) is proposed to solve the task assignment optimization model. The algorithm adds a guiding mechanism in the iterative process, which greatly improves the ability to obtain an optimal solution.

The main contributions of this paper include: A proposed new BSAP-ACO algorithm, predicted meeting point and online track re-planning method, online task reassignment method based on time-sensitive uncertainty problem, a new GMPSO algorithm. Compare with the previous literature, this paper proposes a new, effective, systematic framework for solving cooperative task assignment and track planning problems.

Inspired by all the above factors, this paper focuses on the problem of multiple UAVs attacking ground moving targets. The main contributions of this paper include:

1) The problem of multi-UAVs cooperative task assignment and path planning based on ground moving target is solved in this paper, which includes multi-UAVs cooperative path planning, multi-UAVs cooperative task assignment and online re-planning under uncertainty.
2) A new BSAP-ACO algorithm is proposed for cooperative path planning. On this basis, the method for calculating the predicted meeting point is proposed. It solves the problem that the moving point cannot be directly used as the end point of the track planning algorithm. In addition, the flight path in the tracking mode is also planned according to the speed ratio.
3) The optimization model of multi-UAVs attacking moving targets is established according to the cost and benefit of UAV in different modes, and considering that the maximum flight range, the maximum number of attack tasks, and the continuity of all UAV paths as constraints and so on. The improved GMPSO algorithm is used to solve the problem.
4) An online task re-planning method for multi-UAVs in uncertain environment is proposed. This method can ensure that when the moving state of the target changes, the track of multi-UAVs can be re-planned and the reasonable task assignment scheme can be achieved.

As far as we know, there is no research on attacking moving targets on the ground for the cooperative task assignment and track planning of multi-UAVs.

2 Problem Description and Formulation

The problem considered in this paper is to assign M_v UAVs to complete the task of response, tracking and attack to M_t moving targets, where $M_v > M_t$. There are sequential requirements for M_t targets, first response, then tracking, and finally attack. The response task is the process of the UAV flying to the target position. The tracking mode is that when multiple UAVs cooperate to perform the task, the first arriving UAV needs to track the moving target and wait for other UAVs to arrive. When the UAV performs the mission independently, the range cost of tracking mode is 0. In attack mode, the UAV will destroy the moving target. Referring to the target mirror method [19], each target is mirrored into three points. The UAV flies from the initial position to the first mirror point of the target to complete the response task, and the first mirror point to the second mirror point to complete the tracking task, and the second mirror point to the third mirror point to perform the attack task. In this way, the problem of UAV performing three tasks at one target point can be transformed into a problem of performing only one task at one target point. That is, each UAV performs different tasks at three positions of each target. The set of all target mission points and UAV location points is $V = \{1, 2, 3, ..., M_v + 3M_t\}$. The first M_v elements in the set V are the position points of the M_v UAVs, and the other $3M_t$ elements are the mirror points of all the targets. In order to avoid the collision problem of the unmanned aerial vehicle, the unmanned aerial vehicle is caused to fly at different heights. The completion effect of the task is related to two factors, the first is the total range of all UAVs. Assume that fixed-wing UAVs flying at a constant speed are used to perform the task, so the shorter total range also reduces the average time to complete the task. The second is the gain from attacking enemy targets. It determines, to a certain extent, the necessity of performing tasks. Therefore, before task assignment, it is necessary to carry out track planning for the UAVs, obtain the flight cost between each target and the UAVs, and use the flight cost and attack income as inputs to establish the task assignment model. The cooperative task assignment problem of the UAV can be reduced to combinatorial optimization problem.

$$\min_{x_{i,j}^k} \sum_{i=1}^{M_v+3M_t} \sum_{j=1}^{M_v+3M_t} \sum_{k=1}^{M_v} (L_{i,j}^k \cdot P_{1,l}^k + l_{i,j}^k(\{x_{i,j}^k\}) \cdot P_{2,l}^k - G_{i,j}^k \cdot P_{0,l}^k) \cdot x_{i,j}^k \tag{1}$$

$$s.t. \quad \begin{aligned} &\sum_{j=1}^{M_v+3M_t} \sum_{k=1}^{M_v} x_{i,j}^k = 1 \quad \forall i \\ &\sum_{i=1}^{M_v+3M_t} \sum_{k=1}^{M_v} x_{i,j}^k = 1 \quad \forall j \end{aligned} \tag{2}$$

$$\sum_{j=1}^{M_v+3M_t} x_{i,j}^k = 1.\ if \sum_{j_1=1}^{M_v+3M_t} x_{j_1,i}^k = 1 \quad \forall i, j \tag{3}$$

$$\begin{aligned} x_{i,j}^k &= 0.if (i \le M_v \cap i \ne k) or (j \le M_v \cap j \ne k) or \\ &(i \le M_v \cap j \le M_v) \ \forall i, j, k \end{aligned} \tag{4}$$

$$\sum_{i=1}^{M_v+3M_t} \sum_{j=1}^{M_v+3M_t} (L_{i,j}^k + l_{i,j}^k(\{x_{i,j}^k\}) < L_{Max}^k \quad \forall k \tag{5}$$

$$\sum_{i=1}^{M_v+3M_t} \sum_{j=1}^{M_v+3M_t} x_{i,j}^k < N_{Max}^k \quad \forall k \tag{6}$$

where i and j are two points in the set V, $x_{i,j}^k$ is a decision variable of 0 / 1, indicating whether the k-th UAV flies from point i to point j;$L_{i,j}^k$ is the range from point i to point j in the response mode of UAV. $l_{i,j}^k(\{x_{i,j}^k\})$ is a function, it is the range of the k-th UAV in tracking mode, and $G_{i,j}^k$ is the benefit of the k-th UAV in attack model,The calculation methods of $l_{i,j}^k(\{x_{i,j}^k\})$ and $G_{i,j}^k$ are respectively given in Formulas 7 and 8. $P_{s,l}^k$ is a 0-1 mode selection variable, where $s = \{1, 2, 0\}$ corresponds to the three modes of response, tracking and attack of the UAV. When $s = l$, $P_{s,l}^k = 1$, when $s \ne l$, $P_{s,l}^k = 0$. $l = mod((j - M_v), 3)$, where $mod(\star, a)$ is a function, it represents the remainder after $\star$ removes a. For example, suppose that $(j - M_v) = 8$, $l = mod(8, 3) = 2$. And when $s = 2$, $P_{s,2}^k = 1$, that is, the UAV performs the tracking task at the $(M_v + 8)$ point. L_{Max}^k is the largest range limit for the UAV, and N_{Max}^k is the maximum number of tasks that UAVs can do. Equation 1 consists of two parts, namely the range cost and the attack gain. Constraint (2) ensures that each task point is accessed at least once. Constraint (3) limits the UAV to reach one point and leave from the same point. Constraint (4) ensures that the k-th UAV can only take off and land in its own position, and limits that the UAV can not fly from the i point to the same point. In the constraint, when $j \le M_v$, the UAV is used for landing at the take-off point. Constraints (5) and (6) are respectively flight distance and mission number limits.

$$l_{i,j}^k(\{x_{i,j}^k\}) = \begin{cases} v_k \cdot (t_{i_1,j}^{k_1} - t_{i,j}^k), & t_{i_1,j}^{k_1} > t_{i,j}^k \\ 0, & t_{i_1,j}^{k_1} < t_{i,j}^k \end{cases} \tag{7}$$

$$G_{i,j}^k = Prob_{k,m} \cdot Val_m \quad m \in (1, 2, ..., M_t) \tag{8}$$

where v_k is the speed of the UAV, $t_{i,j}^k = L_{i,j}^k / v_k$ represents the time when the UAV flies from point i to point j.

Specifically, when two UAVs cooperate to carry out the task, the UAV that reaching the mission point first executes the tracking mode, so as to wait for the another UAV to arrive at the same time to execute the attack task. $Prob_{k,m}$ represents the effectiveness of the k-th UAV attack the target m. Val_m is the economic value of the m-th target.

3 New Modified BSAP-ACO Algorithm

In order to speed up the solving time of track planning and ensure that a feasible track can be planned, the ACO algorithm is used to carry out track planning for UAVs. In the case of target movement, the battlefield environment becomes more complex than hitting a fixed target. The traditional algorithm often solves slowly. Although the convergence speed of the method can be accelerated by giving a good initial value, there is still the phenomenon of falling into the local extreme value [20], and there may even be no solution, which does not meet the needs of the actual air combat.

3.1 Establishment of The Fitness Function

Due to the diversity of ant colony algorithms, more feasible solutions are generated.UAVs are not only constrained by the cost of burning oil, but also affected by some obstacles. Therefore, this paper proposes an fitness function to satisfy the flight track of UAVs:

$$J_k = w_1 L_k + w_2 d_k \tag{9}$$

where L_k is the cost of the voyage consumed by the k-th ant under the current iteration. d_k is the safe replacement price of the k-th ant under the current iteration.

$$d_k = K/N \tag{10}$$

$$N = \sum_{p=1}^{n} r_p \tag{11}$$

where r_p is the distance between the p-th track point and its nearest obstacle, K is a constant and n is the number of track points.

3.2 Adaptive Parameter Adjustment

There are three ways to select the path points of the traditional ant colony algorithm:

(1) For a given parameter q_0, if the random parameter Q_0 satisfies $Q_0 > q_0$,the pheromone between path point i and path point j is computed to make the track point corresponding to the maximum heuristic product as the next path point.
(2) If $Q_0 < q_0$, a new random number Q_1 is generated, and if $Q_1 > q_0$, the path point is selected by roulette.
(3) If $Q_0 < q_0$ and$Q_1 < q_0$, the next path point is randomly selected from the feasible path point.

Because the fixed parameters are easy to make the path selection fall into the local extreme value, especially in the face of the more complex moving target problem.Therefore, in this paper, the shortcomings of ant colony algorithm in solving dynamic problems are improved, and the UAV track planning based on BSAP-ACO algorithm is given. After each ant searches, the current ant's search state is utilized, and the current ant's path fitness value is compared with the current best fitness value. Then the parameters are dynamically adjusted according to their differences. The parameter q_0 is dynamically adjusted as shown in the formula:

$$q_0^{t+1} = q_0^t - \frac{q_0^t(l_t - l_{best})}{l_{best}} \tag{12}$$

where l_{best} is the adaptability value of the optimal path searched globally at t time. l_t is the fitness value of the current ant path.When $l_t > l_{best}$, it is proved that the adaptability value of the current ant path is greater than the optimal path, then the search intensity should be reduced in this area, that is, $q_0^{t+1} < q_0^t$. When$l_t < l_{best}$,it is proved that the fitness value of the current ant's path is smaller than the optimal path, and that the current path is a better solution, then the search force should be increased in this area, that is, $q_0^{t+1} > q_0^t$. In order to prevent ants from falling into local extremum, or the diversity of solutions is too large, the range of q_0 adjustment should be limited:

$$\begin{cases} q_0^{t+1} = q_{max} & if \;\; q_0^{t+1} > q_{max} \\ q_0^{t+1} = q_{min} & if \;\; q_0^{t+1} < q_{min} \end{cases} \tag{13}$$

When the ant is lost in the search process, that is, the solution can not be found, the parameter value should be quickly adjusted to $q_0^{t+1} = q_{min}$, to ensure that the next ant is in the region where the current optimal solution is located.

The initial pheromone of ant colony algorithm is random, the first search process is greatly affected by the heuristic value, and the update of pheromone also has a great influence on the ants behind it, so it is easy to fall into the local extreme value.In order to avoid falling into the local extreme value, the jump-out mechanism of the algorithm is set as follows:

when an ant finds the current optimal solution, if there is a subsequent ant's search result equal to the current optimal solution, in order to avoid the search result being locally optimal instead of global optimal, set the enhancement coefficient σ to the subsequent ants. Make the parameters satisfy

$$q_0^{t+1} = q_0(1 + \sigma) \tag{14}$$

Use this method to update the q_0 value, increase the diversity of feasible solutions, and avoid the algorithm falling into the suboptimal solution.

3.3 Bidirectional Search Mechanism

With the increase of search randomness, ant colony algorithm will also lead to an extension of the optimization time while getting a better solution. In order to speed up the search for the optimal solution, the bidirectional search mechanism is used to optimize the algorithm. The traditional ant colony algorithm search mechanism starts from the starting point to the end of the target point, and the communication ability between ant colony individuals has not been fully developed, which affects the convergence speed of the algorithm. The bidirectional search mechanism can solve this problem very well.

The $GroupA$ and $GroupB$ of the two groups of ants are placed at the starting point and the ending point respectively. When the algorithm begins, the two groups of ants start at the same time, and take each other as the starting point to search for the shortest path.The two groups of ants carry the own path information, and when each travels to the next node, it is necessary to judge whether to meet ants of another group, if it is determined that it is not, proceed to the next node until the other point of departure is reached.$\forall p_1 p_2$, $p_1 \in GroupA$, $p_2 \in GroupB$. $p_1 p_2$ are represented by coordinates (x_1, y_1) and (x_2, y_2) respectively in the map.The end condition of the path is

(1) if $p_1(x_1, y_1) = targetpoint$, it can be judged that the ant in the GroupA group reaches the target point.
(2) if $p_2(x_2, y_2) = startpoint$, it can be judged that the ant in the GroupB group reaches the starting point.
(3) if $p_1(x_1, y_1) = p_2(x_2, y_2)$, it can be judged that the two groups of ants meet and integrate into a new path.

In the early stage of BSAP-ACO algorithm, by dynamically adjusting the adaptive parameter q_0, the ants can quickly approach to the optimal solution and affect the search speed of the subsequent ants. The increase of the speed of finding the optimal solution in the initial stage increases the convergence speed of the whole algorithm. Through the given enhancement coefficient method, the development of the algorithm is increased when the algorithm converges, the diversity of the solution is increased, and the condition of falling into the local extreme value is avoided. By adding the bidirectional search mechanism, the optimization speed is reduced due to the jump out mechanism, and the optimal path is quickly found, which effectively accelerates the speed of finding the optimal solution.

4 Track Planning Method Based on Target Movement

4.1 The Predicted Meeting Point

Before the task is assigned to the mobile target, it can be seen from the Formula 1 that it is necessary to determine the flight distance of each UAV to each target. Because the target points of ACO algorithm must be fixed, the first problem to be solved is how to plan the track of moving target.In this part, a simple and effective pursuit method of calculating the predicted meeting point is proposed to solve the problem of the moving target.

When the target is in a moving state, the target position is constantly changing. The optimal track can not only greatly shorten the additional range caused by the uncertainty of the moving target, but also saving fuel, and can speed up the completion time of the whole task. The idea of this method is to calculate the predicted meeting point through the speed of the UAV and the target in the heading direction of the target. This predicted meeting point is a fixed point, so BSAP-ACO can take the predicted meeting point as the target point for track planning. This problem can be solved with a simple mathematical method, and then the predicted meeting point is taken as the target point of the ant colony algorithm. The part of the code for the geometric model is shown in Fig. 1. F is a function, which can calculate the coordinate point where the UAV meets the target according to the coordinate and speed of the UAV, the coordinate, moving direction and speed of the target.

The UAV determines whether the target direction changes by a time step. When the target direction changes, the predicted meeting point is recalculated according to the current UAV and target information. The principle of UAV chasing moving target in two cases is given.

In Fig. 2, the moving target travels along a straight line. After calculating the coordinates of the predicted meeting point, the UAV flies directly from the starting point to the predicted meeting point to perform the task.In Fig. 3, the driving direction of the target continuously changes. The UAV obtains the moving target state with a uniform time step. When the driving direction of the target changes, the UAV recalculates the predicted meeting point and adjusts the course and continues to chase the target along the re-planned track until it meets the target. It is important to note that the speed of the UAVs must be faster than the target speed in order to enable the UAV to successfully pursue the target.

4.2 On-line Track Replanning

In fact, in a real battlefield environment, due to the need to avoid threats such as obstacles and radar, and the

Fig. 1 Codes for obtaining predicted encounter points

Algorithm: For Predicting The Meeting Point Between UAV And Moving Target.

```
Input: Position coordinates , velocity and direction information of the UAV and
the target.
Output: The position coordinates and time of meeting.
z1=1; %% The moving target is upward in the y-axis direction. (z1=0 or 1).
if z1>0
    x= fsolve(@(x)fun(x,x1,x2,y1,y2,a,b,k1),
    [100,100,100]);
else
    x= fsolve(@(x)fun(x,x1,x2,y1,y2,a,b,k1),
    [-100,-100,-100]);
    X = x(1);   % Abscissa of the meeting point.
    Y = x(2);   % Ordinates of the meeting point.
    T = x(3);   % Time cost.
End
%%Call to the fun function
Function F= fun(x,x1,x2,y1,y2,a,b,k1)
    F1 = (x(1)-x2)^2+(x(2)-y2)^2-(b*x(3))^2;
    F2 = (x(1)-x1)^2+(x(2)-y1)^2-(a*x(3))^2;
    F3 = x(2)-y2-k1*(x(1)-x2);
    F = [F1;F2;F3];
end
Obtain the position coordinates of the predicted meeting points and the time cost.
```

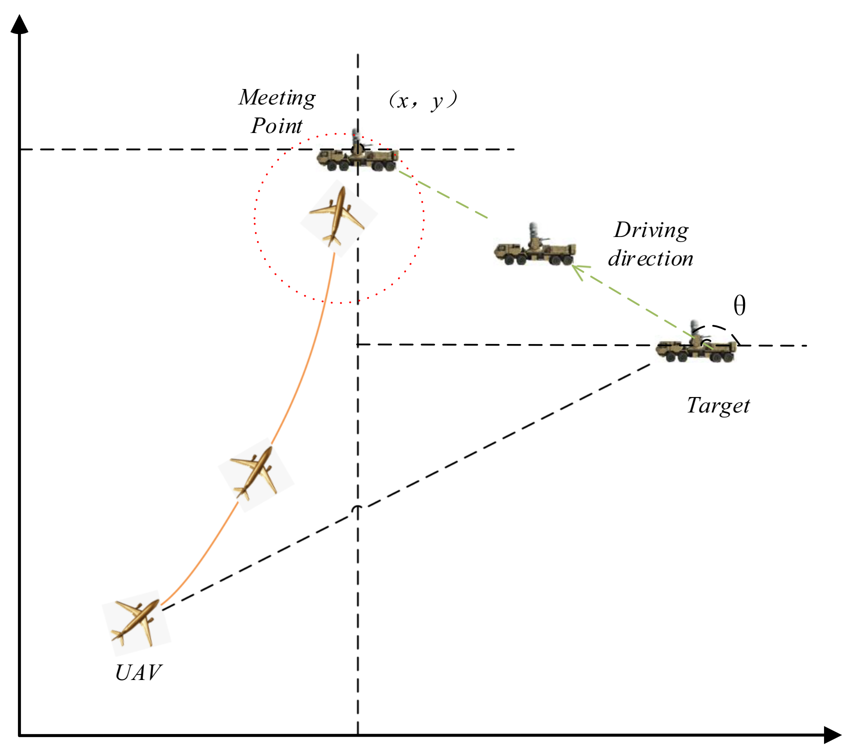


Fig. 2 The target moves along a straight line

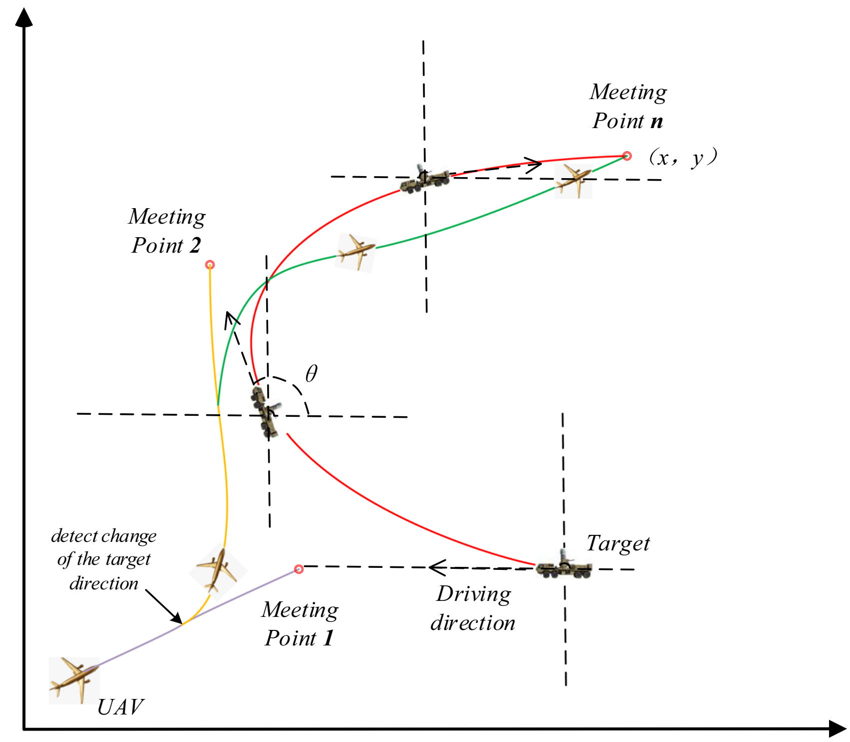


Fig. 3 The target moves along a non-linear line

shortcomings of the track planning algorithm itself, the track between the UAV and the target will not always be straight. This means that when the target travels to the forecast point, the UAVs do not arrive at the same time. In order to solve this problem, we propose an online re-planning method in this part. The UAV and the target are synchronized, and when the target reaches the predicted meeting point after the t-hour, the UAV also flies at t-hour. Since the flight path of the UAV not be a straight line, it does not reach the predicted meeting point. The flight distance of the UAV is $v_k \cdot t$, which is the straight-line distance between the UAV and the predicted meeting point. Therefore, the UAV needs to recalculate the predicted encounter point with the current position coordinates and the target dynamically, and carry on the secondary planning track. The UAV flies to the new predicted meeting point and repeat the process until the target enters the attack range of UAV.

Figure 4 is the principle of online track planning. In the figure, the $Meeting\ point$ 1 is the first predicted meet point of the UAV and the target, and when the target line is set to the $Meeting\ point$ 1, the UAV only flies to the S position due to the need to avoid the obstacle, and then calculates the $Meeting\ point$ 2 with the current UAVs and the target position as the start coordinate, and the UAV carries out track planning on the target point with the new predicted meeting point as the target point, until the target enters the UAV attack range and begins to perform the next task. Figure 5 shows the chase moving target flow based on BSAPA-ACO algorithm for track planning and online re-planning.

5 The Method of UAV Cooperative Attack

5.1 Track Planning in UAV Tracking Mode

If two UAVs cooperate to perform an attack mission, one UAV always waits for another.Therefore, the time cost of a UAV tracking mode is not independent, but based on the time when other UAVs arrive at the mission point. The key to calculating the tracking time and distance cost is to put all UAVs on a unified time line. In order to complete the tracking task, the UAV and the target motion must be synchronized, which means that the speed of the UAV must be guaranteed not less than the target speed, which is similar to the constraint of the pursuit problem in the previous chapter. The difference is that when the UAV is at the same speed as the tracked target, the UAV can fly above the target and track it. In order to satisfy all the constraints of the problem, we assume that the speed of the UAV is faster than the target speed. In this part, a tracking path planning method based on Dubin's paths [21] is proposed.

Assume that the target moves along a straight path or an approximate straight path in a certain time interval.In order to maintain the synchronous movement between the UAV and the target, a path planning method based on the velocity ratio between UAV and target is proposed in this paper. Using v_k andv_m to represent the speed of the UAV and the target respectively, we can define the speed ratio $r_0 = v_k/v_m$, where r_0 is greater than 1.

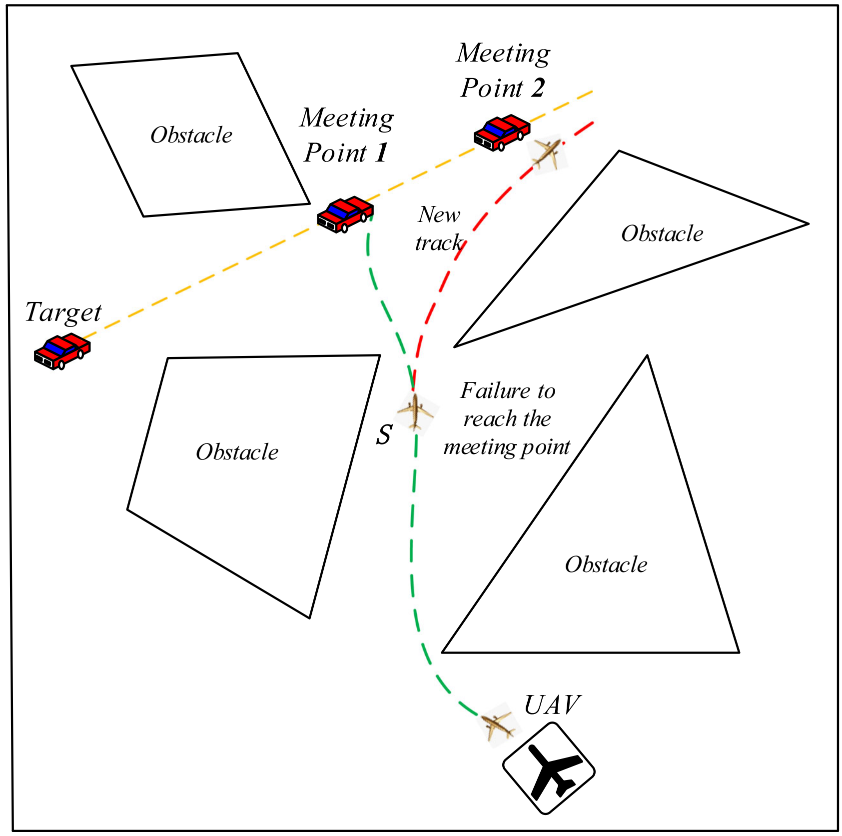


Fig. 4 The process of online re-tracking planning

Assuming r_0 is not equal to positive infinity, that is, the target is not static relative to the UAV, the horizontal distance between UAV and target in moving direction of the target should be the same at all times. Considering that the speed of the UAV is greater than the speed of the target, the motion of the UAV should be the curve motion with the minimum turning radius. The combination of these curves around the target driving path constitutes the tracking flight path of the UAV. The more common and easily calculated curve is the arc, and the arc-based tracking path method is given in Fig. 6.

$S1$: Fig. 6a shows the case of $1 < r_0 < pi/2$. $Path$ is the target moving route, the UAV and the target both

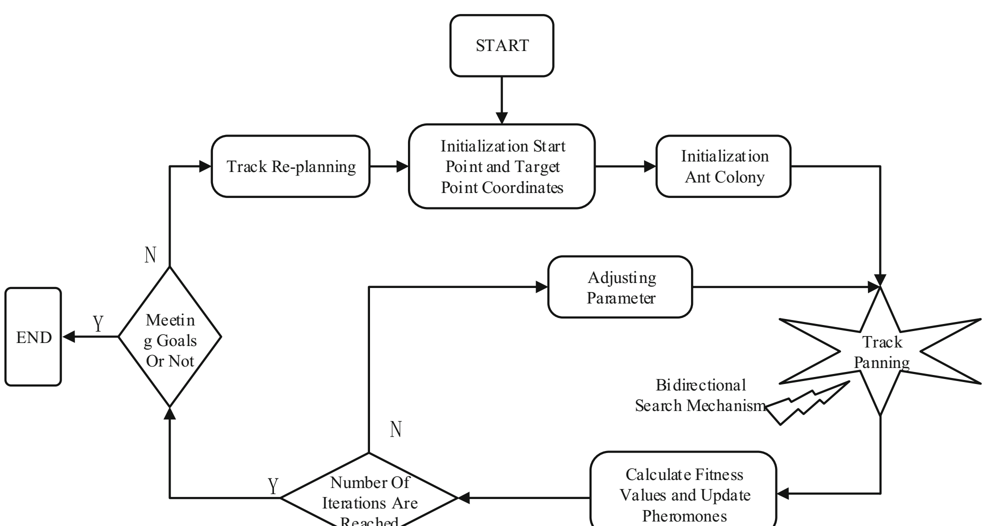


Fig. 5 Online track re-planning flow chart based on BSAP-ACO algorithm

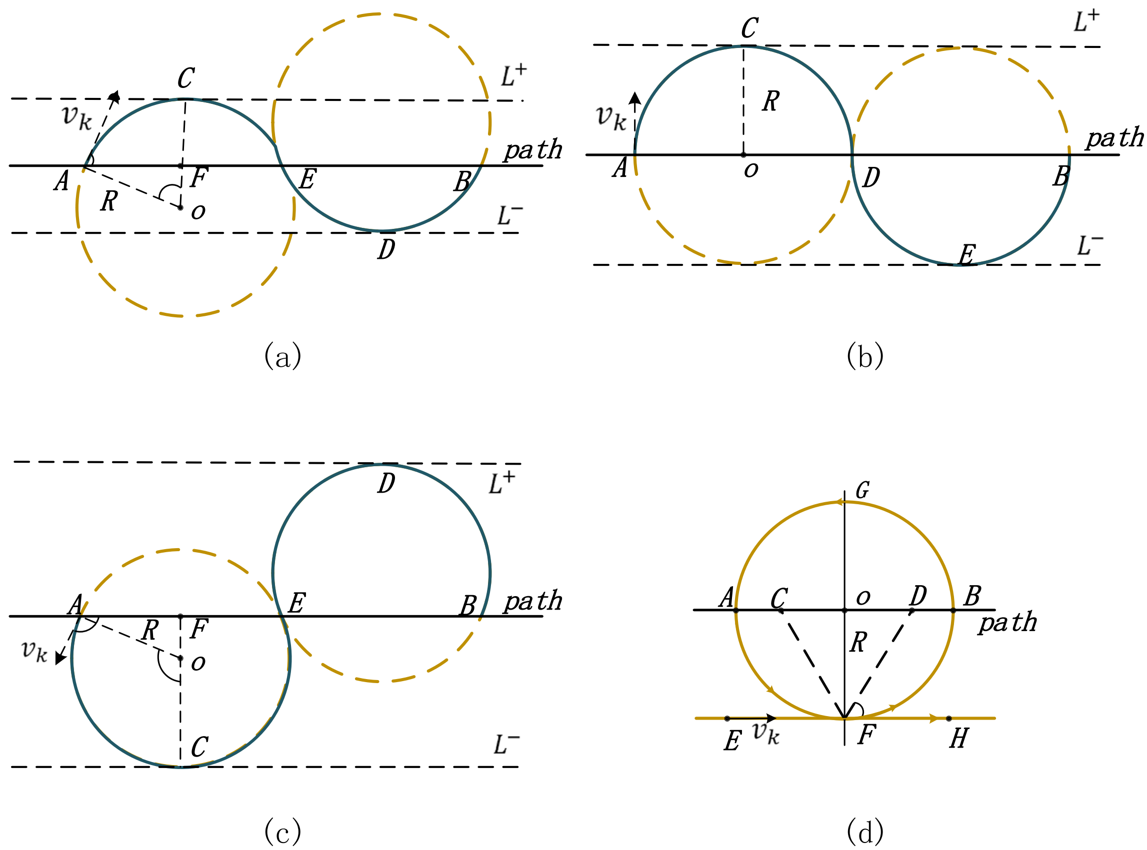


Fig. 6 Flight path planning in tracking mode

start from point A, and the target moves to point B. In order to ensure the smoothing of the tracking path of the UAV, a part of the circumference of radius R is selected as the flight path of the UAV, where R represents the minimum turning radius of the fixed-wing UAV. The tracking track of the UAV is $\widehat{ACEDB}$, which is composed of two circular arcs of a circle. The two arcs are symmetrical, which means that during the same time interval, the distance that the UAV flies on these two tracks is the same in the target moving path. φ is the angle between the target and the initial speed direction of the UAV at the starting point. Above and below the target moving path, there are two parallel lines L^+ and L^-. Their distance from the path is $d = CF$. These two lines are the tangent of two circles, which is also the turning tangent of the UAV. The simulation diagram in this case is shown in Fig. 7.

$S2$: Fig. 6b shows the case of $r_0 = pi/2$. In this special case, the tracking flight path of the UAV is $\widehat{ACDEB}$, which consists of two semicircular $\widehat{ACD}$ and $\widehat{DEB}$. The angle between the UAV and the initial motion direction of the target is a right angle.

$S3$: Fig. 6c shows the case of $pi/2 < r_0 \leq r_*$. Contrary to the case of S1, the angle between the UAV and the initial motion direction of the target is an obtuse angle. The UAV tracking flight path is $\widehat{ACEDB}$, and the distance between the circumferential tangent and the target moving path is $d = CF > R$.

$S4$: $r_* < r_0$. When the speed ratio r_0 is a large value, the path of Fig. 6a, b and c no longer meets the requirements. First consider an extreme situation: $r_0 \longrightarrow \infty$. Compared with the UAV, the target is stationary at the center o position, and the UAV only needs to perform circular motion around the center with any radius greater than or equal to R to successfully track the target. After extending this idea to the general situation, a new tracking path can be designed.

In Fig. 6d, the target moves along the path from point A to point B at a relatively slow speed, and the tracking path of the UAV is $\widehat{EFGFH}$. The movement of the UAV in each cycle can consist of three parts: a complete circle, two segments parallel to the $Path$. The UAV and the target respectively start from point A and point E

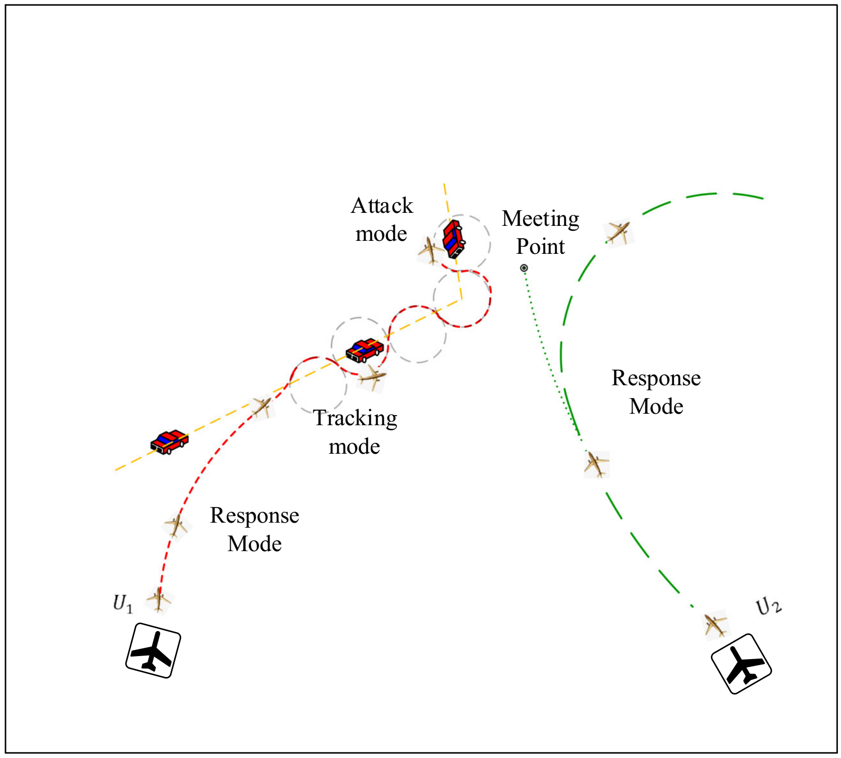


Fig. 7 Processing in sudden situations

with the same abscissa value, and arrive at point B and point H respectively. During the tracking process, the positions corresponding to the target and the UAV are $\{A, E\}, \{C, F\}, \{O, G\}\{D, F\}\{B, H\}$.

The critical value r_* is given in the literature [22]. Because of this paper only considers tracking a moving target with uniform velocity, when using the track planning method of this paper to track non-uniform targets, the relevant calculations and proofs can be found in the literature [22].

5.2 UAV Coordinated Attack Under Uncertainty

In Chapter 4, we solve the problem of track planning for UAV pursuing non-linear moving targets. In this part, we consider a special case: According to the initial moving state of the target, the ideal distribution solution is obtained, that is, the scheme with the minimum value of the function in Formula 1, the target suddenly changes direction at time t. In this case, according to the previous moving state, the task assignment is no longer the optimal distribution solution. The UAV needs to re-determine the task through the current new state of each target. Assign the plan and restart the task. This process is similar to the previous chapter in dealing with non-linear moving targets and solving online redistribution problems. In Fig. 7, a schematic diagram is given in combination with the track planning in the tracking mode of Fig. 6a.

In the Fig. 7, U_1 and U_2 cooperate to perform tasks according to the allocated scheme. The UAV U_1 pursues the target and enters the tracking mode to wait for U_2. The waiting time depends on the distance between U_2 and the target. At a certain moment, the target suddenly changes the moving direction, it means that the target will not appear at the predicted meeting point between U_2 and the target. After recalculating the distance information between UAV and target and using it as the input of Formula 1, a new task assignment scheme is obtained. U_2 performs a new task, and U_1 switches from the tracking mode to the attack mode.

6 A New Algorithm of Particle Swarm Optimization Based on Guidance Mechanism

In this part, we try to propose a new particle swarm optimization algorithm. The algorithm ensures that the system can effectively solve the task assignment problem of moving targets.

6.1 Coding Strategy

Suppose we have M_v VAUs to cooperate attack the M_t moving targets. Our fleet is set to $M_v = \{V_k,\ k = 1, 2, ..., K\}$, where V_k represents k-th UAV. The set of enemy targets is $M_t = \{T_j,\ j = 1, 2, ..., T\}$, and T_j represents the j-th moving target of the enemy. Cooperative operation scheme can be described as finding a suitable target attack scheme S for our UAVs to maximize the cooperative operation efficiency of our fleet. Because it is a multi-UAV coordination problem, this paper uses matrix form to represent an alternative target decision scheme. In order to better solve the problem of multi-UAV cooperative task assignment, binary coding is adopted. And in order to make each particle can represent all the blow information, the dimension of the particle matrix should be $M_v M_t$, and its coding form is as shown in formula (15) and (16).

$$S_{m\times n} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \quad s_{kj} = 0 \; or \; 1 \tag{15}$$

When $s_{kj} = 0$, it means that the k-th UAV does not attacks the j-th moving target. When $s_{kj} = 1$, it represents that the k-th UAV attacks the j-th moving target alone or cooperates with other UAVs. For example, the decoding of a particle is

$$S_f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{16}$$

The particles indicate that the first target is coordinated attack by the UAV_1 and the UAV_3, the second target is struck by the UAV_4, and the third target is attacked by the UAV_2 alone.

6.2 Mutation Operation

Particles undergo mutation operations with a certain probability of variation P_m. The specific implementation method is to randomly exchange any two columns in the particle information matrix S. The changed particle information matrix still satisfies the coding rules. It can supplement and repair part of the information missing in the evolution process of the agent. The method enhances the diversity of the population, and improve the probability that the algorithm jumps out of the local optimum. Take the particle S_f as an example, after the information of the columns 1 and 2 is exchanged, a new particle S_{fc} is generated.

$$S_{fc} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{17}$$

In the new particle S_{fc}, since the original particle conforms to the encoding rule, the mutated particle still satisfies the encoding rule.

6.3 Guiding Mechanism

Set the i-th particle velocity to $v_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$, The individual extremum and the population extremum are respectively $P_{best_i} = (p_{i1}, p_{i2}, ..., p_{iD})^T$ and $G_{best} = (g_{i1}, g_{i2}, ..., g_{iD})^T$. The particle swarm optimization algorithm first initializes n particles in the feasible solution space. Each particle updates its speed and position through evolutionary process of individual extremum and group extremum [23]:

$$v_{id}^{k+1} = \omega * v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (g_{id}^k - x_{id}^k) \tag{18}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{19}$$

$$\omega = \omega_{max} - k/k_{max} \cdot (\omega_{max} - \omega_{min}) \tag{20}$$

where $d = 1, 2, ..., D$; $i = 1, 2, ..., n$; k is the current iteration number, $k = 1, 2, ..., k_{max}$, c_1 c_2 is the acceleration factor, which is a non-negative constant;r_1, r_2 is a random number distributed in the interval [0,1]. ω is the inertia weight, which linearly decrements from the maximum weight ω_{max} to the minimum weight ω_{min} as the evolutionary algebra increases.

It can be seen from the iterative formula of the PSO algorithm that the position and velocity updates of all particles are guided by G_{best}, which is the best particle in the current iteration. That is, at the beginning of each iteration of the algorithm, the current optimal particle of the population is consistent with the previous iteration. The particles in the population quickly move forward to the optimal particle position of the population in the previous iteration. In the process of approaching, if any particle finds a better position than G_{best}, G_{best} will be updated to the particle at the current better solution position, which means that at the end of each iteration process, the G_{best} in this iteration process is guaranteed to be consistent with the previous iteration, or the particle at a better position will be found, thus making the algorithm has fast convergence performance. But at the same time, if G_{best} falls into local optimum, the whole population will appear "premature" phenomenon and fall into the trap of local minimum solution. It's unable to find the global optimal solution about the problem. In order to solve the problem, propose a guiding mechanism to introduce some powerful competitors when the particle is close to the current G_{best}, or the guiding ability of G_{best} is insufficient.The mechanism increases the diversity of the population and improves the global search ability of the algorithm.

Assume that the k-th generation group extremum and the i-th particle's individual extremum are $G_{best}(k)$ and $P_{best_i}(k)$. The algorithm principle is

$$\sum_{i=1}^{Q_{pop}} \Delta P_{best_i}(k) = \sum_{i=1}^{Q_{pop}} f(P_{best_i}(k)) - \sum_{i=1}^{Q_{pop}} f(P_{best_i}(k-1)) \leq 0 \tag{21}$$

$$\Delta G_{best}(k) = f(G_{best}(k)) - f(G_{best}(k-1)) \leq 0 \tag{22}$$

where $\sum_{i=1}^{Q_{pop}} \Delta P_{best_i}(k)$ characterizes the degree of joint optimization of individual extremum, $G_{best}(k)$ characterizes the degree of optimization of population extremum. The larger the value, the stronger the guiding effect of $G_{best}(k)$, and the problem solution develops in a good direction. On the contrary, the guiding effect of $G_{best}(k)$ is weakened, and the problem solution is close to the global optimum, or the local optimum. Therefore, it is necessary to introduce competitive new elite individuals, to increase the guiding ability. The method increases individual diversity and improves the "premature" phenomenon of the algorithm. The specific method is as follows:

Setting individual error threshold Thr. If the following formula is satisfied, the boot mechanism is enabled.

$$|\sum_{i=1}^{Q_{pop}} \Delta P_{best_i}(k)| < Q_{pop} * Thr \wedge G_{best}(k)| < Thr \tag{23}$$

In the formula, $\sum_{i=1}^{Q_{pop}} \Delta P_{best_i}(k)$ is the sum of the optimization degree of all individual extremum in Formula (21), Q_{pop} is the quantity of particles in the population, and $G_{best}(k)$ is the optimization degree of group extremum in Formula (22). When Formula (23) is satisfied, it is proved that the current particle has insufficient ability to find a better position, and the particle needs to be guided.The steps are as follows:

(1) f_1 new individuals are cloned through the current $G_{best}(k)$, and then each individual is mutated according to Formula (24).
(2) Re-initialize f_2 individuals in the feasible domain of the problem solution.
(3) Fitness function values of f_1 new individuals after mutations and f_2 new individuals were calculated, and the individual S_{new} with the best fitness value is selected for comparison with G_{best}.
(4) If $S_{new} < G_{best}$, G_{best} is still the population extremum, otherwise S_{new} becomes the new population extremum.

Using the polynomial mutation strategy in literature [24], a new individual z is generated for the individual q in the replicated f_1 $G_{best}(k)$ in the following formula:

$$z_c = q_c + (x_c^u - x_c^l) * \Delta c \tag{24}$$

Among them, z_c and q_c are the c-th component of z and q, and x_c^u is the upper limit of the c-th component of the decision variable, where the value is 1. x_c^l is the lower limit of the c-th component of the decision variable, where it is 0. The polynomial equation for calculating Δc is

$$\Delta c = (2r_c)^{\frac{1}{\eta_n+1}} - 1, 0 \leq r_c < 0.5 \tag{25}$$

$$\Delta c = 1 - [2(1-r_c)]^{\frac{1}{\eta_n+1}}, 0.5 \leq r_c \leq 1 \tag{26}$$

In the formula, r_c is a random number distributed in the interval [0,1], and η_n is a variation parameter to control the degree of variation.

7 Simulation Results and Analysis

7.1 Ant Colony Algorithm Simulation

In this part, 10 simulation experiments are carried out according to the same environment, and the convergence speed of ACO algorithm, BSAP-ACO algorithm and Dynamic feedback ant colony optimization algorithm (DFACO) in reference [20] are compared. Table 1 is the experiment of 10 simulations.

Before the ant colony algorithm searches, all the pheromones are fixed values, so the ant colony algorithm is greatly affected by the heuristic value when searching for the first time, and the update of pheromones has a better influence on the subsequent ants. In order to avoid falling into local extremum, the "Jump out" mechanism of the algorithm is increased, but the convergence speed is affected. However, the BSAP-ACO algorithm increases the dual search mechanism, accelerates the path optimization speed of ant colony.

Table 1 Comparison between three algorithms

	TIMES	1	2	3	4	5	6	7	8	9	10
ACO	Number of iterations to reach convergence	14	16	13	13	14	15	13	14	14	15
DFACO	Number of iterations to reach convergence	11	12	10	9	12	10	12	9	11	10
BSAP-ACO	Number of iterations to reach convergence	9	7	7	9	8	7	7	9	9	8

Figure 8 shows the relationship between the number of iterations and the minimum track average of the three algorithms in 10 simulation experiments. The ACO algorithm converges when the average number of iterations is 14, and the average track cost is 48.05. The DFACO converges at the 11th time, and the track cost is 45.51 when it converges, while the BSAP-ACO algorithm proposed in this paper converges when the average number of iterations is 8, and the average range cost is 43.96.

As can be seen from Table 1 and Fig. 8, compared with ACO algorithm and DFACO algorithm, the BSAP-ACO algorithm can speed up the convergence of the algorithm by improving the speed of initial search optimization through adaptive parameter adjustment. Through the given enhancement coefficient method, the condition of falling into the local extreme value is avoided. By adding a two-way search mechanism, the speed of finding the optimal solution is effectively accelerated. Therefore, the convergence speed of BSAP-ACO algorithm is faster and the convergence result is better.

The diversity of the three algorithms in this paper is compared by using the diversity comparison scheme in reference [20].

$$DIV(n) = \sqrt{\frac{\sum_{k}^{m}(J_k(n) - avg(J(n)))^2}{m}} \tag{27}$$

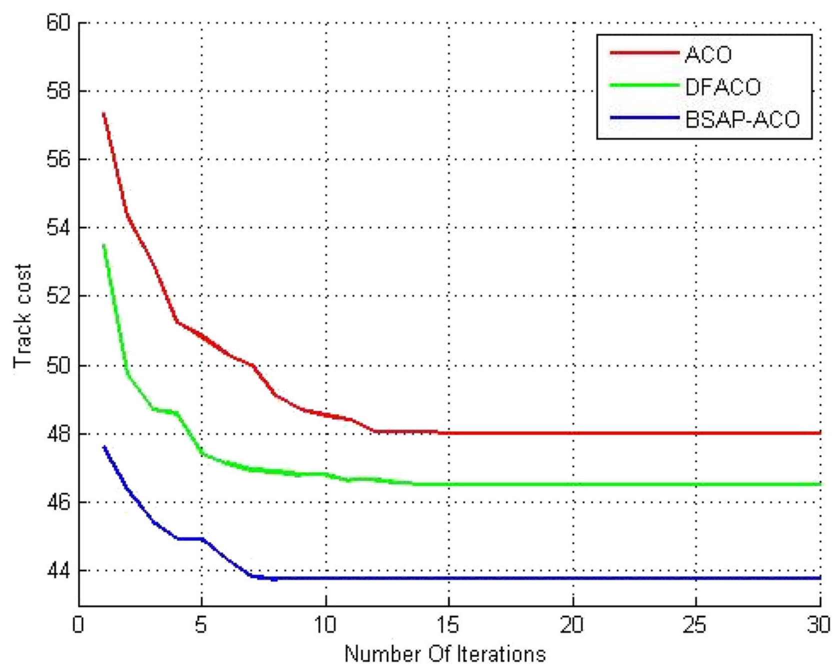
where $J_k(n)$ represents the fitness value of the k-th ant under the n-th iteration. $avg(J(n))$ represents the fitness of all ants under the n-th iteration, m represents an ant that has found a feasible solution.

Figure 9 shows the comparison of the diversity results of the three algorithms, and the red line is the diversity of the traditional ACO algorithm. It can be seen from the graph that the diversity of the traditional ACO algorithm is decreasing in the later iterative period, and it is difficult for ants to get rid of the guiding effect of pheromones on ants. Green line is the diversity of DFACO algorithm. The diversity of this algorithm is the most abundant, but because of the randomness of particle search position increases, the algorithm cannot quickly converge to the optimal solution, which affects the convergence speed. Blue line is the diversity of BSAP-ACO algorithm in this paper.The algorithm adjusts the parameters after each ant search, and the enhancement coefficient is set. When the diversity of the algorithm is reduced, the partial ant is allowed to break away from the guide of the pheromone, while the randomness of the algorithm is increased, the quality of the information is ensured. The variation of the diversity is relatively stable.

7.2 Simulation of Task Assignment

Since the UAV range is a very important input in the fitness function of task assignment, the study of the task assignment algorithm should be carried out after the study of the track planning algorithm. In this section, we first compare several algorithms, and then apply them to the deterministic task allocation problem in two cases.

Fig. 8 Convergence comparison of three algorithms



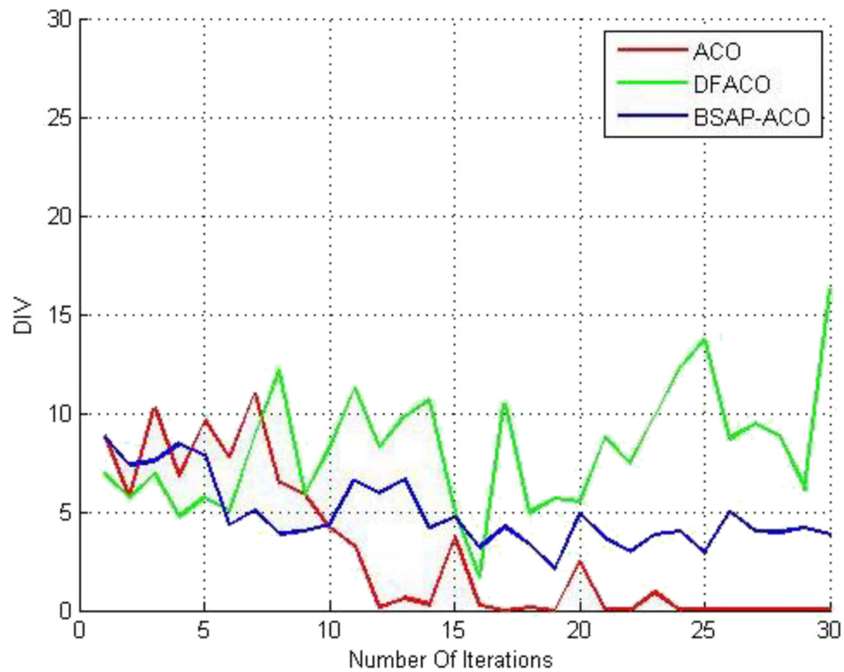


Fig. 9 Diversity comparison of three algorithms

7.2.1 Comparison of Several Algorithms

Set the battlefield in the first. Assume that six UAVs are assigned to four moving targets. UAVs enter the battlefield area at the same time from six different locations. The targets position and moving direction are shown in Fig. 10, and the attack probability is in Table 2.

For simplicity, we set speed of UAVs as $100km/h$ without affecting the performance of the algorithm, the minimum turning radius is $1km$. The speed of targets is $40km/h$, and the economic value Val_m is 10.

When two UAVs work together to attack a target, the UAV that arrives first executes the tracking mode, the track cost is the sum of the range of the response mode and the range of the tracking mode. The range of the tracking mode depends on the arrival time of another UAV.

Based on this battlefield setting, GA algorithm, PSO algorithm and the Multi-objective Particle Swarm Optimization (MOPSO) algorithm are introduced to solve the task assignment problem. In these algorithms, the first solution is random, and then the root algorithm continues to find the optimal solution according to their respective optimization methods, directly to the final number of iterations. The parameters of the GMPSO algorithm are set as follows:

The initial population is set as 50, the number of iterations is set as 100, algorithm acceleration factor is set as $c_1 = c_2 = 1.3$, the number of individuals with replication variation in the guidance mechanism is set as $l_1 = 30$, the number of individuals re-initialization that is set as $l_2 =$

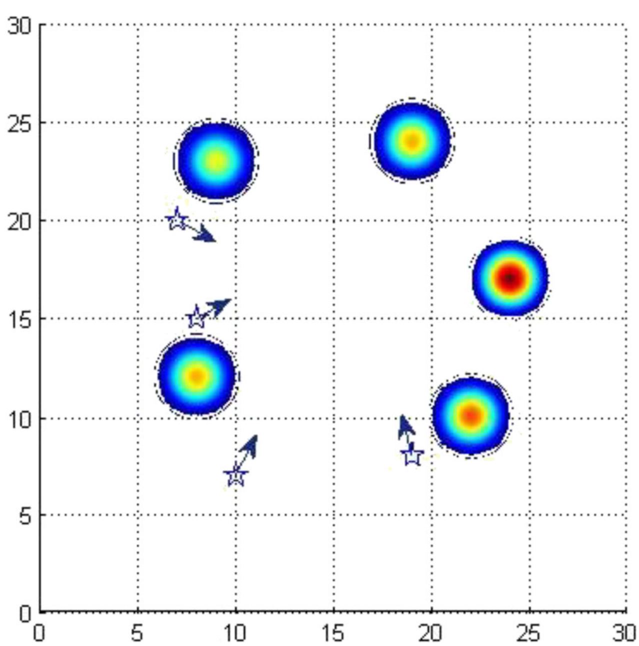


Fig. 10 Position and direction of the targets before the task begins

Table 2 The probability of the UAV attacking the target

	T_1	T_2	T_3	T_4
U_1	0.71	0.85	0.63	0.65
U_2	0.79	0.90	0.75	0.81
U_3	0.82	0.78	0.83	0.90
U_4	0.63	0.76	0.81	0.84
U_5	0.81	0.73	0.77	0.86
U_6	0.90	0.84	0.87	0.73

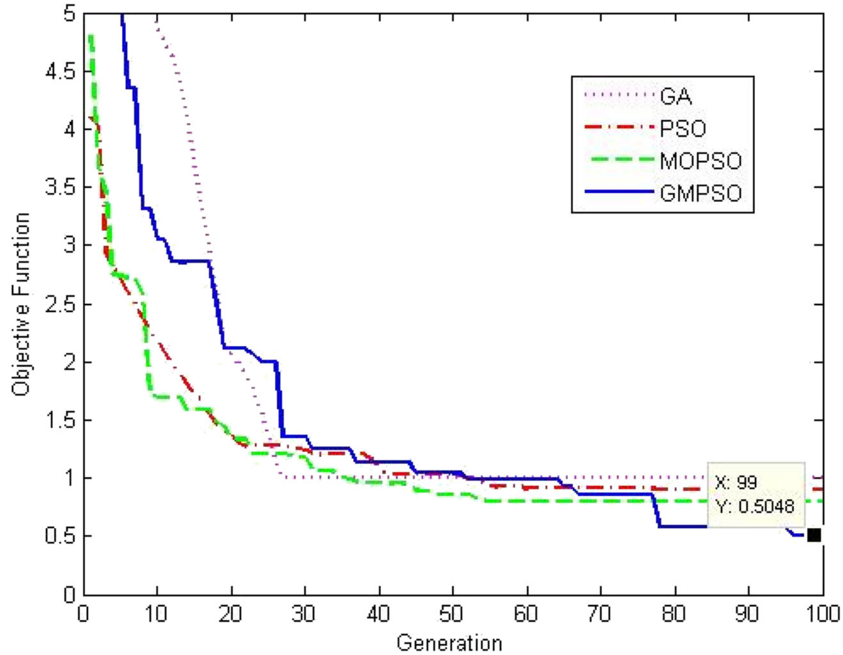


Fig. 11 Best objective function curves of these algorithms

30, mutation operator is set as nm=5 (The GA algorithm intersects and the mutation operator are both set as 20) , error threshold is set as Thr=0.03, the weight range of the algorithm are $\omega_{min} = 0.35$ and $\omega_{max} = 0.9$.

In Fig. 11, due to the difference of the initial particles and the influence of the constraint when calculating the fitness value, the convergence rate of each algorithm is quite different in the initial stage. It can be seen that when the iteration is to the thirty times, the convergence of the algorithms is similar. Because of the guiding mechanism of GMPSO algorithm, compared with the other three algorithms, the ability to find the optimal solution is better. Due to space constraints, this paper selects MOPSO and GMPSO, the two best performing algorithms, to compare the distribution case and the value of fitness function.

Table 3 is the task assignment scheme of MOPSO algorithm. In the table, the task allocation scheme is that U_2 and U_3 attack T_1, U_1 attacks T_2, U_6 attacks T_3, U_4 and U_5 attack T_4. The output final fitness function value is 0.7204.

Table 4 is the task assignment scheme of GMPSO algorithm. In the table, the task allocation scheme is that U_1 attacks T_1, U_2 and U_3 attack T_2, U_4 and U_5 attack T_3, U_6 attacks T_4. According to Formula (1), the final fitness function output from this task allocation scheme is 0.5048.

7.2.2 Task Assignment With Time-sensitive Uncertainty Problem

In Chapter 4 we propose a method to solve the assignment problem with time-sensitive uncertainty. In order to verify the effectiveness of the method, a comparative simulation experiment is carried out in this paper. Compare according to the following two cases.

(a) **Deterministic off-line problem**

UAVs enter the battlefield at the same time from (1, 30) (1, 20) (1, 10) (10, 1) (20, 1) (30, 1). The task assignment scheme in Table 4 is the best task scheme in this case, and the track diagram is given in Fig. 12.

The position of hollow triangle in the picture is the initial position of the UAV. After the complete mission, the UAV returns to its respective starting point to land,

Table 3 Task assignment scheme of MOPSO algorithm

	T_1	T_2	T_3	T_4
U_1	0	1	0	0
U_2	1	0	0	0
U_3	1	0	0	0
U_4	0	0	0	1
U_5	0	0	0	1
U_6	0	0	1	0

Table 4 Task assignment scheme of GMPSO algorithm

	T_1	T_2	T_3	T_4
U_1	1	0	0	0
U_2	0	1	0	0
U_3	0	1	0	0
U_4	0	0	1	0
U_5	0	0	1	0
U_6	0	0	0	1

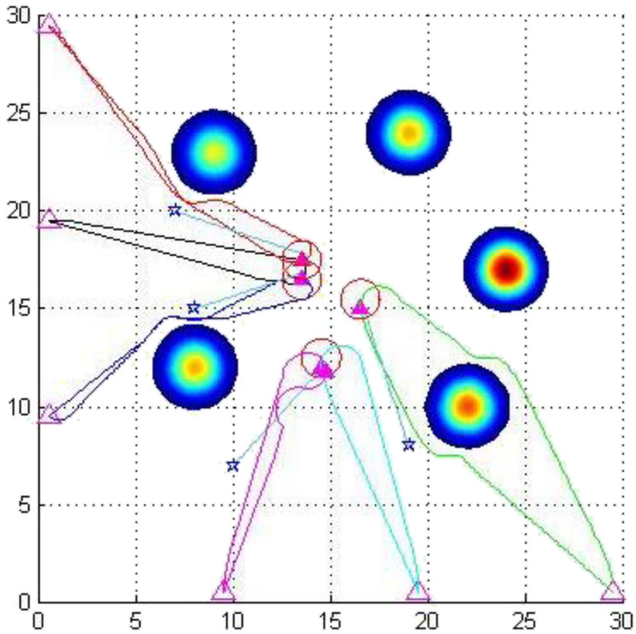


Fig. 12 Planning track of the best task assignment result

the blue circular areas are obstacles, and the solid triangle position is the coordinates of the UAV carrying out the strike mission. From the task allocation scheme in Fig. 12, we can get the following characteristics:

$C1$: When the attack probability is not the decisive factor, the farther target is often carried out by a separate UAV, which can avoid the two UAVs from cooperating with the extra-long range mission.

$C2$: When a target is attacked cooperatively, two UAVs with a short interval of time to reach the same target often team up to carry out the task, which is because the tracking distance depends on the time difference between the two UAVs.

These characteristics are the same as our prejudgment, which also proves the rationality of the distribution scheme.

(b) On-line problem of time-sensitive uncertainty problem

When the target is on the way, the direction of the target will be changed, which will cause the final adaptability of the initial allocation scheme to become unpredictable. The UAV must be adjusted the task allocation scheme to obtain the optimal fitness function again. Under this condition, we set up some targets to change the direction at different time, and compare the final task allocation scheme with Table 3. Make the following changes to the driving situation of the target:

Table 5 Task allocation scheme based on time-sensitive uncertainty problem

	T_1	T_2	T_3	T_4
U_1	0	1	0	0
U_2	0	1	0	0
U_3	1	0	0	0
U_4	0	0	0	1
U_5	0	0	0	1
U_6	0	0	1	0

When the target T_2 is driven to coordinate (11, 16), the direction of the target T_2 will be changed to 70^o. When the target T_4 reaches the coordinate (18.2, 10.4), change the direction of the drive to 200^o.

The task allocation scheme is shown in Table 5.

In Table 5, the task allocation scheme is that U_3 attacks T_1, U_1 and U_2 attack T_2, U_6 attacks T_3 and U_4 and U_5 attack T_4. Great changes have taken place between this task allocation scheme and the straight line driving of the target.The value of fitness function is 0.5389, which value is slightly larger than the straight line driving condition of the target.The track diagram is given in Fig. 13.

Compared with the relatively smooth track in Fig. 12, the UAV flight track in Fig. 13 has obvious orbit change trace, which indicates that the UAV reallocates the task after judging the change of the moving state of the target, and when the target of the mission changes, the UAV flies to the new target. It is important to note that when the

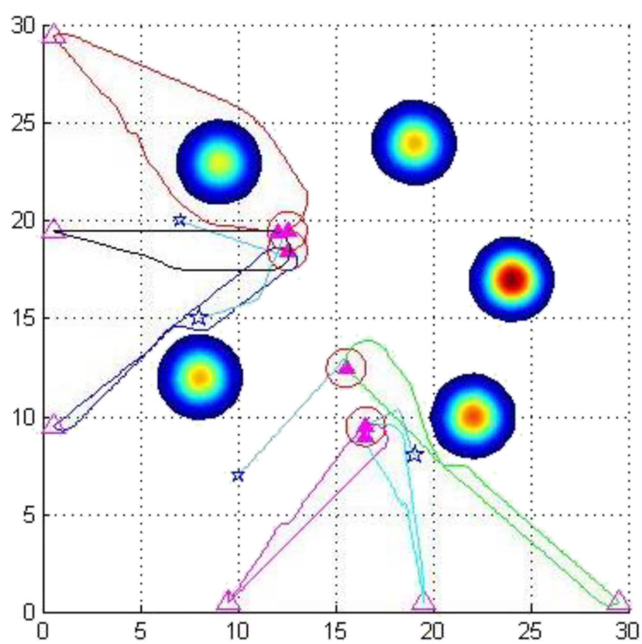


Fig. 13 Planning track of the best task assignment result under time-sensitive uncertainty

target movement state changes, the fitness value of the recalculated task allocation scheme is not sure whether it will increase or decrease. The uncertainty of target movement makes it more possible to change the numerical value. If the direction of target movement is more beneficial to the current state of the UAV, the final fitness value may be smaller than that of the original scheme.

The simulation results show that the task allocation and track planning method not only reasonably selects the route to avoid obstacles, but also obtains the minimum fitness value, determines the global optimal task assignment scheme, and achieves the best combat effect.

8 Conclusion

In this paper, the problem of multi-UAV cooperative task allocation and track planning considering multi-ground moving targets is fully simulated and solved, including the problem of track planning for moving targets, the problem of off-line task assignment and the problem of online task re-assignment with uncertainty. A new BSAP-ACO algorithm based on adaptive parameter adjustment and bidirectional search is proposed, which can effectively plan the optimal track. In the case of target movement, a method of the predicted meeting point is proposed to solve the problem that the moving point cannot be used as the target point of the track planning algorithm. An effective method to solve the problem of track planning in UAV tracking mode is presented. A new GMPSO algorithm is proposed, which can solve the optimization problem under multiple constraints. Finally, the simulation experiment is carried out to verify the effectiveness of the method.

Although some research on the cooperative task assignment and track planning of ground moving targets has done in this paper, there are still some problems that need to be further studied. For example, in the online re-planning phase, how to ensure the smoothness of the track switching part, and how to plan the track of moving target when the speed of the target is not constant. These problems need to be further explored in future work.

Funding Information National Natural Science Foundation of China Funding Project(61503255)

References

1. Cristian, R.A., David, C.: Constrained multi-objective optimization for multi-UAV planning[J]. J. Ambient. Intell. Humaniz. Comput. **10**(6), 2467–2484 (2018)
2. Xing, W., Zhao, Y., Karimi, H.R.: Convergence analysis on multi-AUV systems with leader-follower architecture[J]. IEEE Access **5**(1), 853–868 (2017)
3. Kuncheva, L.I.: Full-class set classification using the Hungarian algorithm[J]. Int. J. Mach. Learn. Cybern. **1**(1-4), 53–61 (2010)
4. Forsmo, E.J., Grotli, E.I., Fossen, T.I., et al.: Optimal search mission with Unmanned Aerial Vehicles using Mixed Integer Linear Programming[C]. International Conference on Unmanned Aircraft Systems, pp. 253–259, IEEE (2013)
5. Nygard, K.E., Chandler, P.R., Pachter, M.: Dynamic network flow optimization models for air vehicle resource allocation[C]. In: American Control Conference, pp. 1853–1858, IEEE (2001)
6. Lemaire, T., Lacroix, A.R.: A Distributed Tasks Allocation Scheme in multi-UAV Context[C]. In: IEEE International Conference on Robotics and Automation, pp. 3622–3627, IEEE (2004)
7. Chen, J., Sun, D.: Coalition-based approach to task allocation of multiple robots with resource constraints[J]. IEEE Trans. Autom. Sci. Eng. **9**(3), 516–528 (2012)
8. Shima, T., Rasmussen, S.J., Sparks, A.G.: UAV Cooperative multiple task assignments using genetic algorithms[C]. In: American Control Conference, pp. 2989–2994, IEEE (2005)
9. Zhu, Z.X., Tang, B.W., et al.: Multi-robot task allocation based on an improved particle swarm optimization approach[J]. Int. J. Adv. Robot. Syst. **14**(3), 1–22 (2017)
10. Wang, H.P., Liu, C.G., Li, W.J.: On optimizing UAV (unmanned aerial vehicle) mission planning with ANT algorithm[J]. Journal of Northwestern Polytechnical University **23**(1), 98–101 (2005)
11. Rasmussen, S.J., Shima, T., Mitchell, J.W., et al.: State-space search for improved autonomous UAVs assignment algorithm[C]. Decision and Control, IEEE, pp. 2911–2916 (2004)
12. Gu, X., Zhang, Y., Chen, J., et al.: Real-time decentralized cooperative robust trajectory planning for multiple UCAVs air-to-ground target attack[J]. J. Aerosp. Eng. **229**(4), 581–600 (2014)
13. Chen, Y.B., Yu, J.Q., Su, X.L., et al.: Path planning for multi-UAV formation[J]. J. Intell. Robot. Syst. **77**(1), 229–246 (2015)
14. Hu, Z.H., Zhao, M., Yao, M.: Cooperative attack path planning for unmanned air vehicles swarm based on grid model and bi-level programming[J]. J. Inform. Comput. Sci. **8**(4), 671–679 (2011)
15. Ozgur, K.S.: Flyable path planning for a multi-UAV system with genetic algorithm sand beaier curves[C]. In: International Conference on Unmanned Aircraft Systems, pp. 14–48 (2013)
16. Oliveira, T., Encarnacao, P.: Ground target tracking control system for unmanned aerial Vehicles[J]. J. Intell. Robot. Syst. **69**(1-4), 373–387 (2013)
17. Zhang, Y., Yang, X.X., et al.: UAV Flyable trajectory generation and its tracking control[J]. Int. J. Contr. Autom. **8**(4), 383–396 (2015)
18. Moon, S., Oh, E., Shim, D.H.: An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments[J]. J. Intel. Robot. Syst. **70**(1-4), 303–313 (2013)
19. Deng, Q.B., Yu, J.Q., Wang, N.F.: Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes[J]. Chin. J. Aeronaut. **26**(5), 1238–1250 (2013)
20. Huang, C., Fei, J.Y., Liu, Y.: Smooth path planning method based on dynamic feedback A * ant colony algorithm[J]. Transactions of the Chinese Society of Agricultural Machinery **48**(04), 39–45+107 (2017)
21. Beard, R.W., McLain, T.W.: Small unmanned aircraft: Theory and practice[M]. Princeton University Press (2012)
22. He, Z.R., Xu, J.X.: On the trackability of a moving target by a fixed-wing UAV using geometric approach[C]. IEEE International Symposium on Industrial Electronics, pp. 1572–1577 (2014)
23. Shi, F., Wang, H., Yu, J.: 30 Cases of Intelligent Algorithms in MATLAB[M]. Beijing University Press, Beijing (2010)

24. Deb, K.: Multi-Objective Optimization using evolutiary Algorithms[M]. New Jersey America (2001)

Chen Xia graduated from Northeast University with a PhD and a postdoctoral degree, Shenyang, China, in 2006. She is now a professor of Automation School of Shenyang Aerospace University, and a master's tutor of pattern recognition and intelligent system.

Liu Yongtai received the M.S degree from Shenyang Aerospace University, Shenyang, China, in 2019. He focuses on the research of Multi-UAV cooperative task allocation?CMulti-UAV cooperative path planning and path planning and Multi-UAV formation control. He has been engaged in the development of artificial intelligence robots.

Yin Liyuan received his M.S degree from Shenyang Aerospace University, Shenyang, China, in 2019. He is currently studying for a doctorate in Harbin Institute of technology, Harbin, China. His current research interests include fuzzy control.

Qi Lijie received the B.S degree from Shenyang Aerospace University, Shenyang, China, in 2018. She is currently working toward the M.S. degree at the Shenyang Aerospace University. Her current research interests include UAV path planning and formation control.