Key Technology in Multi-UAV Conflict Detection and Resolution Strategy

*TANG Xinmin**, *JI Xiaoqi*, *LI Teng*

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R.China

(Received 14 March 2020; revised 14 April 2020; accepted 18 April 2020)

Abstract: Based on ADS-B surveillance data, this paper proposes a multi-unmanned aerial vehicle (multi-UAV) collision detection method based on linear extrapolation for ground-based UAV collision detection and resolution, thus to provide early warning of possible conflicts. To address the problem of multi-UAV conflict, the basic ant colony algorithm is introduced. The conflict simplification model of the traditional basic ant colony algorithm is optimized by adding a speed regulation strategy. A multi-UAV conflict resolution scheme is presented based on speed regulation and heading strategies. The ant colony algorithm is improved by adding angle information and a queuing system. The results show that the improved ant colony algorithm can provide multi-UAV joint escape routes for a multi-UAV conflict situation in airspace. Unlike the traditional ant colony algorithm, our approach converges to the optimization target. The time required for the calculation is reduced by 43.9%, and the total delay distance caused by conflict resolution is reduced by 58.4%.

Key words: unmanned aerial vehicle (UAV); ground station; ant colony algorithm; conflict resolution

CLC number: U8 **Document code:** A **Article ID:** 1005-1120(2020)02-0175-12

0 Introduction

Thanks to the progressing drone technologies targeting lower cost, simpler operation, and stronger maneuverability, drones have been widely used in many fields such as agricultural plant protection, public security policing, and forest fire protection[1]. And research in techniques that use multiple unmanned aerial vehicles (UAVs) has experienced a significant increase in recent decades[2]. The expansion of the drone market has created severe challenges for airspace management[3]. Up to now, the prevalent operation and control strategy for UAVs has been mainly using fixed airspace regulation methods to reduce security risks by avoiding or strictly controlling the entry of other types of aircraft into the designated airspace. However, because airspace resources are limited and the number of drones is steadily increasing, it is necessary to study ground station-based drone collision-detection technology and to develop a multi-UAV conflict resolution strategy to ensure safe operation of drones.

Researchers all around the world have divided the multi-UAV conflict path planning problem into two parts: Conflict detection and escape route planning. In terms of conflict detection, the major problem of conflict detection and resolution lies in the accurate collision prediction and efficient adjustment of the flight status among conflicting UAVs[4-5]. Conflict detection and resolution (CDR) algorithms for UAV swarms often merge with formation control algorithms. Control algorithms with CDR can be sorted into two general types: Rule-based approaches and optimization-based approaches[6]. CDR methods that can solve possible collisions between UAVs based on the CDR method. Ho et al. introduced an adaptation of ORCA, called adapted ORCA, which can minimize the ORCA generated deviations from the nominal flight path and a value for separation distance between UAVs that use the

*Corresponding author, E-mail address: tangxinmin@nuaa.edu.cn.

How to cite this article: TANG Xinmin, JI Xiaoqi, LI Teng. Key technology in multi-UAV conflict detection and resolution strategy[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(2): 175-186.

http://dx.doi.org/10.16356/j.1005-1120.2020.02.001

airspace efficiently can be determined by simulating realistic UAV traffic for delivery[7]. Farlik et al. analyzed the possibility of detecting flying aircraft based on calculations and simulations, and experiments in laboratories and measurements of the exterior were subsequently performed[8]. Cheng et al. proposed a learning based framework where the conflict and avoidance problem can be formulated as a Markov decision process and the maneuver policies for the UAV were learned[9]. Navarro et al. developed a sense and avoid strategy based on a deep learning approach[10]. Hu et al. proposed distributed velocity-aware algorithm to serve motion planning of multiple UAVs[11]. During motion, a decentralized consensus algorithm was adopted for agents to converge to their positions for the desired formation and to maintain a stable geometric configuration. A temporal and spatially integrated conflict-detection model was improved, especially for UAV swarms. A token-allocation strategy was especially improved for distributed coordination to resolve the partial knowledge of the airspace condition. Damping of the coordination was proposed to address data dropouts and transmission delays[12]. Reich et al. carried out research on establishing flight safety intervals and quantifying conflict probabilities[13-15]. To address the inconsistency problem caused by the state information of multi-UAV communication under a dynamic environment, a state estimation method was proposed based on the Kalman algorithm[16]. A real-time conflict resolution algorithm based on model predictive control (MPC) was introduced to address the flight conflict resolution problem in multi-UAV scenarios by Chen et al. And the controller structure of the distributed nonlinear model predictive control (DNMPC) was designed to predict potential conflicts and calculate control variables for each UAV. Numerical simulations of multi-UAV coordination were performed to verify the performance of the proposed algorithm[17]. An advanced flocking strategy was developed to address the autonomous multi-UAVs' cooperative flight[18]. In Ref.[19], tracking-learning-detection (TLD) was used for target tracking from a UAV and achieved a good result in a short term. Fu et al. introduced a novel robust visual tracking algorithm for UAVs in the midair to track an arbitrary aircraft at real-time frame rates, together with a unique evaluation system[20]. Tang et al. and Xu et al. considered drone performance constraints, established an airborne conflict detection model, and classified conflict alert levels. According to these, a corresponding escape strategy was developed[21-22]. These studies have outlined multi-UAV collision detection approaches.

As for multi-UAV escape route planning, current mature and popular algorithms include the A* algorithm, the artificial potential field method, genetic algorithm[23] and the ant colony algorithm. The A* algorithm is applied to traverse a graph to plan conflict-free flight routes[24]. Granger et al. improved the A* algorithm to generate a coordinated and decentralized embarked conflict solver[25]. Tang et al. assumed the destination of a known drone, introduced the A* algorithm to improve the open table and to reduce computational complexity, and considered the performance of the drone. They finally proposed a plan for determining an escape route[26]. Researchers introduced artificial potentials[27] for obstacle avoidance as well as leader-follower schemes for goal searching[28]. Zhu et al.[29] studied how to use the artificial potential field method to plan a conflict resolution path for UAVs in a dynamic environment. Wu et al. proposed an artificial potential field-guided evolutionary algorithm to track single targets[30]. The advantage of the artificial potential field method is that it can plan a smooth route, but easily fall into local optima simultaneously. In 1991, Dorigo first proposed the ant colony algorithm[31]. By releasing different quantities of pheromones for different feasible paths, the pheromone concentration of different paths was controlled, and the concentration was increased in the optimal line to guide a gradual search for the optimal path. This algorithm has expertise at solving function optimization and path planning problems. Sara et al. studied an optimized ant colony algorithm based on the probability and space attributes of the target, which effectively reduced the time required to reach the search target[32]. Li et al. combined the UAV performance index with the ant colony algorithm and used the artificial potential field method to obtain the prior value, which shortened the flight distance of the multi-UAV mission[33]. Zhuang et al. improved the ant colony algorithm based on angle information and an elite strategy and obtained better conflict resolution strategies for multiple aircraft[34]. These studies laid the foundation for research on ground-based unmanned aircraft conflict resolution.

In summary, at present, for multi-UAV path planning problems, the ant colony algorithm can provide a path planning solution. However, most current studies consider a single condition, usually adopting only a steering strategy, and fail to take advantage of the drone's flexibility. In addition, due to the large amount of searching in the initial stage of the algorithm, it takes a long time for the optimal path to converge to the result. Therefore, in this study, based on UAV performance, a UAV flight protection zone is established, and the ground station is used to detect and predict flight conflicts in the airspace. Further, based on the basic ant colony algorithm, an improved ant colony algorithm with angle information and a sorting system is proposed. The multi-UAV algorithm is introduced to the conflicting multi-UAV airspace and a coordinated speed regulation and direction adjustment strategy is developed for collision-free paths with the objective of minimizing total delay distance. The function provides the optimally linked escape strategy under multi-UAV conflict scenarios.

1 Collision Detection for UAVs

Ground station-based drone collision detection requires three steps.

Step 1 Set up a flight protection zone for the drones and convert the acquired surveillance data into spatial coordinates with the ground station as the origin.

Step 2 Stratify the drones in the airspace according to their respect heights, analyze the target trajectories of the drones at the same level, and initially screen target pairs for potential conflicts.

Step 3 Carry out specific analysis using a linear extrapolation method and calculate the UAVs that have been screened to determine the conflict situation more precisely.

1.1 Safety interval of UAVs

A safety interval for UAVs involves establishing a corresponding flight protection zone for each drone to ensure that the drone has reasonable space to perform its mission while avoiding the risk of collision. The construction of each UAV's flight protection zone depends on the flight performance of the UAV, including its maximum operating speed and the time it takes to respond to commands. Based on these factors, the flight protection zone is larger than the size of the drone itself. The protected area contains two elements: The vertical safety interval D_{ver} and the horizontal safety interval D_{hor}[17]. Fig.1 shows a model of the protected area centered the drone.

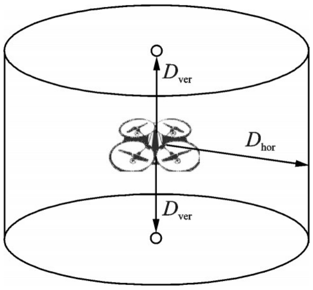

Fig.1 UAV flight protection area

1.2 Conflict pre-judgment algorithm based on linear extrapolation

A local cartesian coordinate system with the ground base station as the origin is established by transforming the WGS84 coordinate system into a local rectangular spatial coordinate system. The UAV's latitude and longitude and its velocity in the airspace are converted into position and velocity vectors relative to the ground station by a coordinate conversion method. UAVs at different heights correspond to different two-dimensional planes.

(1) Conflict screening

For any drone among the N UAVs in a two-dimensional plane, it is necessary to make $N-1$ conflict judgments. For all drones, it is theoretically necessary to make $N(N-1)/2$ judgments. In reality, many targets are far apart and will not generate conflicts any time soon. Therefore, setting a distance threshold can reduce the number of detections and improve collision detection efficiency.

In the two-dimensional plane, set the horizontal distance threshold R_{T_A}. When the horizontal interval is greater than the threshold, the target is filtered. R_{T_A} is set as

$$R_{T_\mathrm{A}} = 2T_\mathrm{A} \cdot v_\mathrm{mh} + D_\mathrm{hor} \tag{1}$$

where T_A is the set warning time, v_mh the maximum horizontal flight speed of the target, and D_hor the horizontal interval of the target protection zone.

Linear extrapolation is applied to objects that move linearly. The algorithm is based on the current position and velocity vectors of the two machines and calculates whether a distance less than the safety level interval is possible and the whether safety height interval between the two objects during the warning time $[0, T_\mathrm{A}]$, is a possibile.

(2) Horizontal warning detection

In the horizontal direction, it is assumed that the two objects to be evaluated are a, b. According to the established local coordinate system, the horizontal coordinate of a is (v_{ax}, v_{ay}), the horizontal velocity of a is (v_{ax}, v_{ay}), the horizontal coordinate of b is (x_b, y_b), and the horizontal velocity of b is (v_{bx}, v_{by}). A few definitions are needed for the relative states of a, b flying in a horizontal plane.

Definition 1 The true distance between a and b is

$$D_{ab} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \tag{2}$$

Definition 2 Speed of intruder b relative to the x-axis of drone a

$$\Delta v_{bx} = v_{bx} - v_{ax} \tag{3}$$

Definition 3 Speed of intruder b relative to the y-axis of drone a

$$\Delta v_{by} = v_{by} - v_{ay} \tag{4}$$

The steps for detecting collisions in the horizontal direction are as follows.

Step 1 According to the drone speed information, find the relative speed Δv. If $|\Delta v| = 0$, go to Step 2; if $|\Delta v| \neq 0$, go to Step 3.

Step 2 Compare the horizontal interval $D_{ab}(0)$ with the safety interval D_hor. When $D_{ab}(0) < D_\mathrm{hor}$, there is a conflict in the horizontal direction, and the conflict period $[T_{h1}, T_{h2}] = [0, T_\mathrm{A}]$; otherwise, $[T_{h1}, T_{h2}] = \varnothing$, and the evaluation ends.

Step 3 Calculate the horizontal interval as

$$D_{ab}(t) = \sqrt{[x_b - x_a + \Delta v_{bx} t]^2 + [y_b - y_a + \Delta v_{by} t]^2} \tag{5}$$

The time range when $D_{ab}(t) < D_\mathrm{hor}$ is determined, that is, in the time range $[T_{h1}, T_{h2}]$ the conflict occurs.

2 Multi-UAV Conflict Resolution Modeling

2.1 Basic ant colony algorithm application modeling

A simplified model of multi-UAV ground station-based conflict resolution planning can be formulated as follows.

(1) The drone has good maneuverability and can even hover in the air. However, because the power consumption during hovering is far greater than the working power consumption, it is assumed that the drone does not use the hovering strategy when a collision is imminent, but one of three speed-regulation strategies has to be chosen: Low, medium, or high speed.

(2) Assume that the drone can use any of three steering maneuvering strategies when implementing an escape strategy: Maintaining the original heading, yawing 30° to the left, or yawing 30° to the right.

(3) Assume that the drones are equipped with ADS-B surveillance equipment, thus the position, speed, and altitude of the drone at each point in time can be obtained, and the target position of the drone is known.

The n UAVs are divided into l steps along the line joining their current positions to their target position. During the forward propulsion process, there are three heading strategies and three speed strategies available to the UAVs. It is assumed that when a drone moves from position A at time $t-1$ to position B at time t, there are nine possibilities for position C that the drone may reach at time $t+1$. As shown in Fig.2, C_1, C_2, and C_3 are positions that represent high-speed, medium-speed, and low-speed left yaw at 30°; C_4, C_5, and C_6 are positions that represent high-, medium-, and low-speed propulsion along the original heading; C_7, C_8, and C_9 are positions that represent high, medium, and low speed with a right yaw at 30°; the nine possible paths are $BC_1, BC_2, \cdots, BC_9$. The current UAV position and velocity vector matrix is represented as $\boldsymbol{A}_k = [x, y, \theta, v]$, and the next UAV matrix can be expressed as

$$\boldsymbol{A}_{k+1} = [x + vt\cos\theta', y + vt\sin\theta', \theta', v'] \tag{6}$$

where θ' can be $\theta + \frac{\pi}{6}$, θ, or $\theta - \frac{\pi}{6}$.

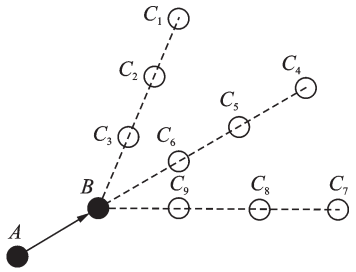

Fig.2 UAV path selection

Based on the above three conditions, collision detection is performed on all the UAVs in the airspace. When the conflict conditions are met, the trajectory can be changed by adjusting the heading and speed of the drone, and the distance of the deviation is guaranteed to be the shortest under the premise of escaping the conflict.

The three-dimensional coordinates of drone i after operating for k steps are denoted as $\boldsymbol{A}_i^k$. The objective function of the algorithm is

$$f = \min \sum_{i=1}^{n} D_i \tag{7}$$

where the objective function f is required to be the minimum value, that is, the shortest distance of the drone from the original target in the conflict resolution situation; D_i represents the delay distance caused by drone i in this heading change. The delay distance generated by the drone can be expressed as the distance between position $\boldsymbol{A}_i^l(x_i, y_i)$ and the destination coordinates $\boldsymbol{B}_i(X_i, Y_i)$ after l steps. The expression is

$$D_i = \left\| \boldsymbol{A}_i^l - \boldsymbol{B}_i \right\| \tag{8}$$

The expression can be expanded as follows

$$D_i = \left\| \boldsymbol{A}_i^l - \boldsymbol{B}_i \right\| = \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \tag{9}$$

To make the conflict escape process reasonable, during maneuvering, a minimum horizontal interval constraint is added. This means that, at each moment, a minimum horizontal interval must be maintained between any two drones in the airspace. Assume that drones $i(x_i, y_i)$ and $j(x_j, y_j)$ must satisfy the following expression

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > \delta \tag{10}$$

where δ indicates the minimum security interval when the drone is operating.

According to the previous assumptions and definitions, the ant colony algorithm is used to calculate the fundamental conflicts for n drones in the airspace.

First, the probability $P_k(t)$ of the drone passing through a certain path k to another spatial position at time t is calculated as

$$P_k(t) = \frac{\tau_k(t)}{\sum_{k=1}^{9} \tau_k(t)} \tag{11}$$

where $\tau_k(t)$ is the pheromone value of path k at time t. The value is updated after each iteration is completed, and the updated expression is

$$\tau_k(t+1) = (1-\rho)\tau_k(t) + \Delta\tau_k(t)$$
$$0 < \rho < 1 \tag{12}$$

where $\tau_k(t+1)$ is the pheromone value on the k-segment trajectory at $t+1$; ρ the attenuation coefficient of the pheromone value; $1-\rho$ the pheromone retention coefficient; and $\Delta\tau_k(t)$ the total amount of pheromone obtained by path k in this iteration.

Regarding the problem of pheromone distribution during execution of the ant colony algorithm, there are usually three models, and their calculation expressions are as follows.

(1) Ant cycle system model

$$\Delta\tau_k(t) = \begin{cases} Q/L_i & \text{The } i\text{-th ant chooses path } k \\ 0 & \text{Others} \end{cases} \tag{13}$$

where Q is a constant that represents the sum of pheromones that an ant can produce in each cycle and L_i the total distance that ant i has traveled.

(2) Ant density system model

$$\Delta\tau_k(t)=\begin{cases}Q & \text{The } i\text{-th ant chooses path } k\\ 0 & \text{Others}\end{cases} \tag{14}$$

(3) Ant quantity system model

$$\Delta\tau_k(t)=\begin{cases}Q/d_i & \text{The } i\text{-th ant chooses path } k\\ 0 & \text{Others}\end{cases} \tag{15}$$

where d_i is the distance that ant i passes through the section.

It can be seen from Eqs.(13)—(15) that each model considers a different range. The ant density system model considers the ants passing through the path, whereas the ant quantity system model and the ant cycle system model separately consider the distance traveled by the ants from local and overall perspectives and then assign the pheromone concentrations. This study uses the path with the least delay from the perspective of global optimization. Therefore, using the ant cycle system pheromone release model, the shorter the path, the greater the pheromone increments that are obtained during each iteration.

According to Eq.(13) for the ant cycle system model, the expression for the pheromone release amount $\Delta\tau_k(t)$ is

$$\Delta\tau_k(t)=\sum_{i=1}^{m}\frac{Q}{f} \tag{16}$$

where Q is the total amount of pheromone left after a drone has gone completely and f the sum of the final delay distances of all drones.

After the objective function, constraints, path selection method, and pheromone release method have been determined, the operation of the ant colony algorithm can be divided into five steps.

Step 1 Assign initial values to the variables and parameters that need to be used during the operation of the ant colony algorithm, including the number of drones $M=N\times m$, the starting pheromone at each node φ, the volatilization factor ρ, the maximum number of loop iterations $C_{\max}$, and the sum of the pheromones released by the drone at each cycle Q, and the starting value of the iterative computation k.

Step 2 Divide the M UAVs into N groups and m batches and place them at N different starting points. For each drone, the probability of passing to the next target is obtained according to Eq.(11), and a random selection is conducted according to probabilities to guide the drone to complete the step node selection.

Step 3 Constrain the selection of each batch of drone nodes according to Eq.(10), which should meet the minimum safety interval of the drone. If the selected path does not satisfy the constraint, the batch of drones is re-selected, and the path combination is no longer selected at the same time to ensure that no drone at any time has conflict with other drones.

Step 4 Calculate the distance between each drone and the target point after completing the path. Calculate the target function value of each batch of drones using Eqs.(7)—(9). The minimum objective function value is obtained by comparing the objective function values in one cycle, and then each node pheromone value is updated according to the pheromone release expression.

Step 5 If the number of cycles is less than the set maximum number of iterations, return to Step 2 and continue the iterations. Otherwise, stop iterating and output the escape path.

Fig.3 shows the operation of the basic ant colony algorithm.

2.2 Improved ant colony algorithm application model

2.2.1 Introduction of angle information

In the selection process of the UAV at time t to the next time target, the basic ant colony algorithm is based solely on the ratio of the pheromone at the next node to the total pheromone at all feasible nodes, and this ratio is proportional to the probability of selecting the node. On this basis, the UAV is constrained to the angle between node $t+1$ and the starting point, which means increasing the probability of selecting a node with a smaller angle between the end point and the starting line, as shown in Fig.4.

In Fig.4, the two points S and D are the starting point and the destination of the drone. The

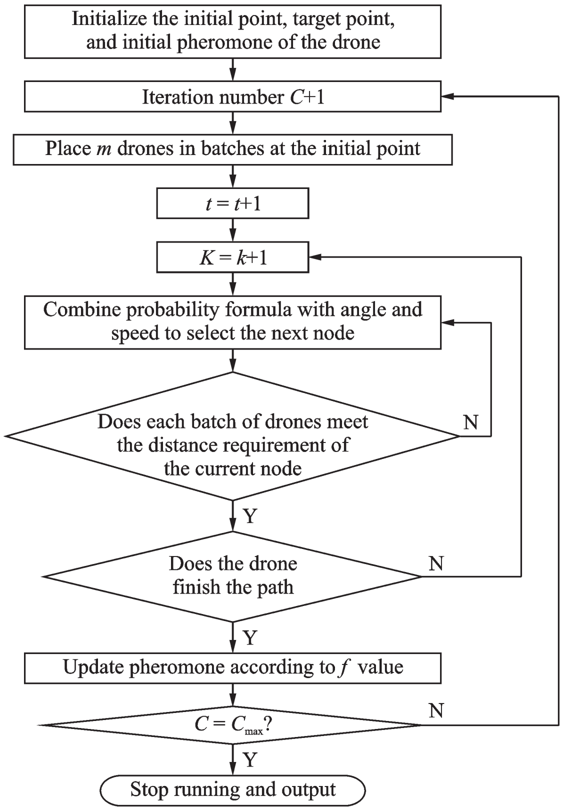
Initialize the initial point, target point, and initial pheromone of the drone
Iteration number C+1
Place m drones in batches at the initial point
$t=t$+1
$K=k$+1
Combine probability formula with angle and speed to select the next node
Does each batch of drones meet the distance requirement of the current node
N
Y
Does the drone finish the path
N
Y
Update pheromone according to f value
$C=C_{\max}$?
N
Y
Stop running and output

Fig.3 Ant colony execution process

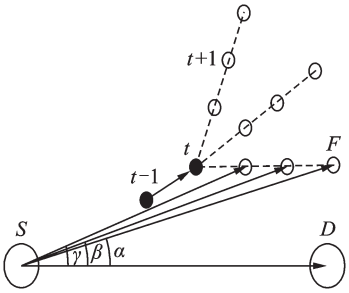

Fig.4 Preferred angle diagram

drone has nine nodes to choose from at time t. The available nodes are connected to the starting point, and then the starting point is connected to the target point. The smaller the angle formed by the two lines, the more likely it is to be selected. In this example, the angle α formed by the selection of node F is the smallest, and therefore the probability that node F will be selected is increased when the next node is selected.

2.2.2 Introduction of ant system based on sorting

A sorting-based ant colony system is introduced in this study. After each iteration, the objective function values f calculated by all batches are sorted. The smaller the value, the higher the ranking. Only the top γ unmanned drones will release pheromone through the path without missing the optimal solution, which ensurs that the best path has more pheromone superposition, speeds up algorithm convergence, and obtains the optimal solution faster.

The pheromone update formula is

$$\tau_k(t+1)=(1-\rho)\tau_k(t)+\Delta\tau_k^*(t)$$
$$0<\rho<1 \tag{17}$$

$$\Delta\tau_k^*(t)=\begin{cases}\sum_{i=1}^{m}\frac{Q}{f} & i\in p^k\\ 0 & \text{Others}\end{cases} \tag{18}$$

where f is the sum of the final delay distances of a group of drones; p^k the set of paths in the top k; and $\Delta\tau_k^*(t)$ the sum of the pheromone released by the drone on path k in one iteration after the sort is introduced.

After introducing the angle information and sorting the ant system, the operation of the ant colony algorithm can be divided into five steps.

Step 1 Assign initial values to the variables and parameters that need to be used during operation of the ant colony algorithm, including the number of drones $M=N\times m$, the starting pheromone concentration at each node φ, the volatilization factor ρ, the maximum number of loop iterations $C_{\max}$, the sum of the pheromones released by the drone at each cycle Q, and the starting value of the iterative computation k.

Step 2 Divide the M UAVs into N groups and m batches and place them at N different starting points. Add the factors of angle selection by weight and combine the pheromone concentrations to find the probabilities of going to the next target. According to the probabilities, a random selection is made to guide the drones to complete the l-step node selection.

Step 3 Constrain the selection of each batch of drone nodes according to Eq.(10), which should meet the minimum safety interval of the drone. If the selected path does not satisfy the constraint, the batch of drones reselects the path and does not select the path combination at the same time, which ensures that there is no conflict between any drone and other drones at any time.

Step 4 Calculate the distance between each drone and the target point after completing the path, and calculate the objective function value f of each batch of drones by Eq.(7). The objective function values in a loop are compared to obtain the minimum objective function value. The minimum objective function values are sorted from small to large, and the first γ values are recorded. At the same time, the pheromone concentration at each node is updated according to the improved pheromone release expression.

Step 5 If the number of cycles is less than the set maximum number of iterations, return to Step 2 and continue the iterations; otherwise, stop iterating and output the escape path.

3 Multi-UAV Conflict Resolution Simulation

3.1 Basic ant colony algorithm simulation

Based on the above theory, a model was built, and a simulation scenario was designed. In the airspace range of 18 km×18 km, there were four drones, U_1, U_2, U_3, and U_4. The initial state of the four drones is shown in Table 1.

Table 1 Drone start state

Drone number	Starting point coordinates/km	End point coordinates/km	Heading/(°)	Speed/(m•s^{-1})
1	[0,9]	[18,9]	0	10
2	[9,18]	[9,0]	−90	10
3	[18,18]	[0,0]	−135	15
4	[18,0]	[0,18]	135	15

The process of getting the drone from the starting point close to the target point was decomposed into 20 steps using a distance value. Each drone could adopt one of three flight speeds: 5 m/s, 10 m/s, or 15 m/s. As shown by the path of the UAVs simulated in Fig.5, the path running in the current state shows that a multi-UAV conflict condition will occur during the flight. Therefore, the basic ant colony algorithm, the improved ant colony algorithm added to the sorting system, and the improved ant colony algorithm with the ordering system and angle information were used to perform the conflict escape simulation. During the simulation, the total number of drones was set to $M=80$ (divided into four groups), $\rho=0.3$, $Q=100$, and the maximum number of iterations $C_{\max}=200$.

For the multi-UAV ground station-based collision resolution problem, the conflict resolution path planned by the basic ant colony algorithm is shown in Fig.6.

According to the escape path obtained by the basic ant colony method, although the conflict situation between the multiple machines can be understood, the UAVs are far away from the target position, and the quality of the escape path plan is not

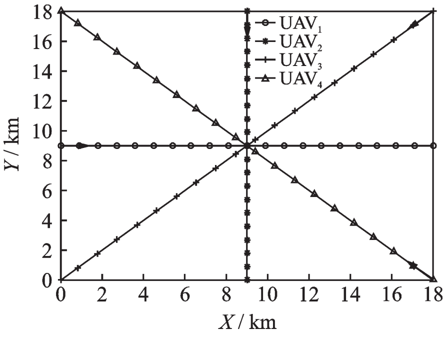

Fig.5 Simulation plan design

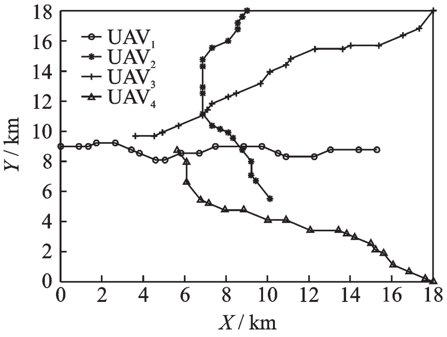

Fig.6 Escape path plan by basic ant colony algorithm

good. The reason is not only that the initial conflict path selection is computationally intensive, but also that the low-quality a priori path causes the basic ant colony algorithm to converge finally to a local optimal solution instead of the global optimal solution. Therefore, the algorithm is improved by adding angle information and a sorting system to reduce the deviation of the azimuth, reduce the amount of calculation, shorten the calculation time, and verify the effect of the improved algorithm.

3.2 Improved ant colony algorithm simulation

The algorithm was revised to ensure that the best path has more pheromone superposition, speeding up algorithm convergence, and faster speed of obtaining the optimal solution.

The results of the escape path plan, shown in Fig.7, show that the improved algorithm objective function value is better than that obtained from the basic ant colony algorithm, and that the delay generated by the improved algorithm by introducing the sorting system and angle information is significantly reduced.

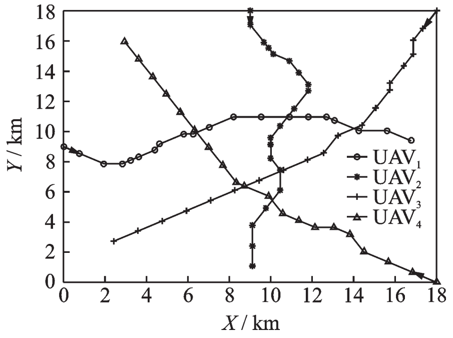

Fig.7 Improved ant colony algorithm conflict resolution results

In the face of multiple unmanned aircraft conflicts, the drone needs to maintain a minimum interval between itself and other drones. Fig.8 shows the real-time distances between the drones during the unmanned aircraft unblocking process. At this time, the distance between UAV$_2$ and UAV$_3$ in the operation to Step 14 is a minimum of 1.58 km, which is greater than the minimum interval and meets the minimum interval requirement in the conflict resolution process.

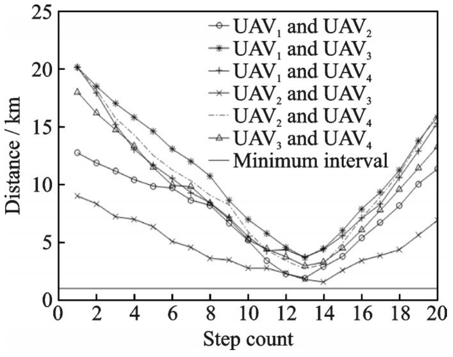

Fig.8 Distances between two drones

The two algorithms were simulated 30 times each, and the f-values obtained by both algorithms were recorded. The length of the minimum delay was calculated every three times, as shown in Fig.9. It can be seen that the improved ant colony algorithm reduces the final total delay distance.

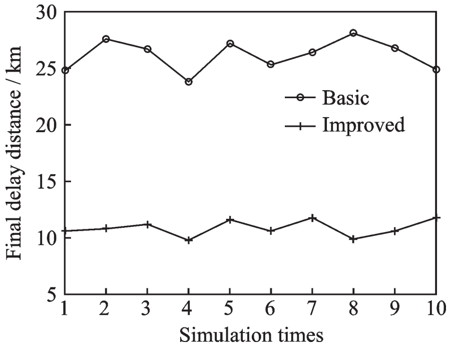

Fig.9 Final delay distance based on the conflict resolution algorithm

In the simulation, each algorithm was simulated 30 times. The time taken by the various algorithms was recorded (Table 2), and the shortest time was selected 10 times. Obviously, the improved ant colony algorithm reduces computation time.

Table 2 Calculation time of the optimization algorithm s

Number of simulation	1	2	3	4	5	6	7	8	9	10	Average
Basic ant colony algorithm	48.3	48.6	49.5	50.2	49.7	48.8	50.3	50.4	50.6	49.5	49.6
Improved ant colony algorithm	28.6	28.5	28.7	27.6	26.8	26.9	27.4	27.4	28.2	28.4	27.8

3.3 UAV priority allocation

Depending on the length of time that each drone is waiting in the air and the priority of each drone's operation, it is determined whether the drone needs to be prioritized. By increasing the proportion of priority drones in the optimal target, the system ensures that UAVs that need to be prioritized consistently obtain a better solution.

During the simulation, the priority levels of UAV_1 and UAV_2 were increased to ensure their priorities in reaching their destination, and the escape routes were simulated. The simulation results (Fig. 10) showed that the UAV conflict can be solved and that the priority-guaranteed UAVs can obtain a better path plan in the simulation process, that is, it will be closer to its target point after completing the operation according to the planned route, whereas the other unmanned machines will sacrifice distances from their target points. After performing multiple simulations, the average calculation time was 28.4 s, which was not much different from the optimization algorithm in this paper. However, average delay distance was 13.3 km, which was 13% longer than the no-priority situation. Therefore, this method can effectively realize priority drone path planning at the expense of a small amount of overall optimization.

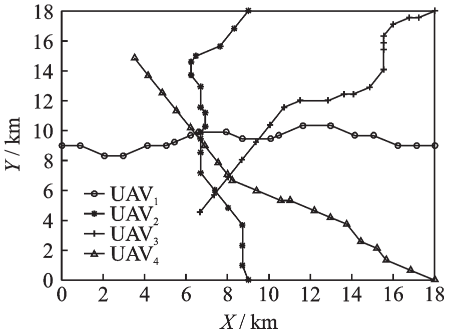

Fig.10 UAV priority allocation release results

4 Conclusions

With the objective of resolving the multi-UAV conflict situation that may occur when operating UAVs in airspace, this paper proposes an inter-plane linear extrapolation method for collision detection to determine possible multi-UAV conflicts. Based on the basic ant colony algorithm, the ordering system and angle information are combined, and speed and heading adjustment strategies for the drones are also considered. The minimum delay distance is the optimization target to plan the path of the drone under imminent collision. Case analysis demons trates the basic ant colony algorithm can provide solutions for multiple unmanned aircraft conflicts, but the total delay distance is large, and the calculation time is long. With the improved ant colony algorithm proposed in this paper, computational efficiency is improved, the time required for the algorithm to converge to the optimal target path is reduced by 43.9%, and the final total delay distance is reduced by 58.4%.

References

[1] SHAKHATREH H, SAWALMEH A, ALFUQAHA A, et al. Unmanned aerial vehicles: A survey on civil applications and key research challenges[EB/OL]. (2018-04-19)[2020-01-15]. https://arxiv.org/abs/1805.00881.

[2] CAI G, WANG B, CHEN B, et al. Design and implementation of a flight control system for an unmanned rotorcraft using RPT control approach[C]//Proceedings of the 30th Chinese Control Conference. Yantai, China: IEEE, 2011: 6492-6497.

[3] DU Siliang, WANG Ce, SUN Hongjia, et al. Numerical analysis of rotor / fuselage aerodynamic interference during tilting quadrotor UAV transition[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2018, 50(2): 179-185. (in Chinese)

[4] CONDE R, ALEJO D, COBANO J A, et al. Conflict detection and resolution method for cooperating unmanned aerial vehicles[J]. Conflict Detection and Resolution Method For Cooperating Unmanned Aerial Vehicles, 2012, 65: 495-505.

[5] PALLOTTINO L, FERON E M, BICCHI A. Conflict resolution problems for air traffic management systems solved with mixed integer programming[J]. IEEE Trans Intell Transp Syst, 2002, 3: 3-11.

[6] KURIKI Y, NAMERIKAWA T. Consensus-based cooperative formation control with collision avoidance for a multi-UAV system[C]//Proceesings of 2014 American Control Conference. Portland, USA: IEEE, 2014: 2077-2082.

[7] HO F, GERALDES R, GONÇALVES A, et al. Improved conflict detection and resolution for service

UAVs in shared airspace[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1231-1242.

[8] FARLIK J, KRATKYORCID M, CASAR J, et al. Multispectral detection of commercial unmanned aerial vehicles[J]. Sensors, 2019, 19(7): 1517.

[9] CHENG Q, WANG X, YANG J, et al. Automated enemy avoidance of unmanned aerial vehicles based on reinforcement learning[J]. Applied Sciences, 2019, 9(4): 1-22.

[10] NAVARRO D V, LEE C, TSOURDOS A. Sense and avoid using hybrid convolutional and recurrent neural networks[J]. IFAC-Papers OnLine, 2019, 52(12): 61-66.

[11] HU Y, YAO Y, REN Q, et al. 3D multi-UAV cooperative velocity-aware motion planning[J]. Future Generation Computer Systems, 2020, 102: 762-774.

[12] WAN Y, TANG J, LAO S. Distributed conflict-detection and resolution algorithm for UAV swarms based on consensus algorithm and strategy coordination[J]. IEEE Access, 2019, 7: 100552-100566.

[13] REICH P G. Analysis of long-range air traffic system: Separation standards I II III[J].Journal of the Institude of Navigation, 1966, 19(2): 169-186.

[14] PRANDINI M, HU J, LYGEROS J, et al. A probabilistic approach to aircraft conflict detection[J]. IEEE Trans Intelligent Transport System, 2000, 1(4): 199-220.

[15] LEE C Y, JAMES K K. Using intent information in probabilistic conflict analysis[C]//Proceedings of AIAA Guidance, Navigation, and Control Conference. Boston, USA: AIAA, 1998: 10-12.

[16] WU Z, LI J, ZUO J, et al. Path planning of UAVs based on collision probability and Kalman filter[J]. IEEE Access, 2018, 6: 34237-34245.

[17] CHEN H, NAN Y, YANG Y. Real-time conflict resolution algorithm for multi-UAV based on model predict control[J]. Algorithms, 2019, 12(2): 47.

[18] TANG Y, HU Y, CUI J, et al. Vision-aided multi-UAV autonomous flocking in GPS-denied environment[J]. IEEE Transactions on Industrial Electronics, 2019, 66(1): 616-626.

[19] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking learning detection[J]. IEEE Trans Pattern Anal Mach Intell, 2010, 34(7): 1409-1422.

[20] FU C, CARRIO A, OLIVARES-MENDEZ M A, et al. Robust real-time vision-based aircraft tracking from unmanned aerial vehicles[C]//Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China: IEEE, 2014: 5441-5446.

[21] TANG Xinmin, CHEN Ping, LI Bo. Receding horizon optimization of en route flight conflict resolution strategy[J]. Journal of Traffic and Transportation Engineering, 2016, 16(5): 74-82. (in Chinese)

[22] XU Yunhong, ZHOU Rui, XIA Jie, et al. Conflict detection and optimal control for UAVs in automatic avoiding of dynamic obstacles[J]. Electro-Optical & Control, 2014, 21(1): 1-6. (in Chinese)

[23] XU Zhijiang, ZHAO Wanzhong, WANG Chunyan, et al. Local path planning and tracking control of vehicle collision avoidance system[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2018, 35(4): 729-738.

[24] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Trans Syst Sci Cybern, 1968, SSC-4(2): 100-107.

[25] ALLIOT J M, DURAND N, GRANGER G. FACES: A free flight autonomous and coordinated embarked solver[J]. Air Traffic Control Quart, 2000, 8(2): 109-130.

[26] TANG Xiaodong, WU Jing. Research on three-dimensional route planning based on improved A* algorithm[J]. Journal of Electronic Technology, 2015, 41(5): 163-166. (in Chinese)

[27] TANNER H G. Flocking with obstacle avoidance in switching networks of interconnected vehicles[C]//Proceedings of IEEE International Conference on Robotics and Automation. New Orleans, USA: IEEE, 2004: 3006-3011.

[28] SHI H, WANG L, CHU T. Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions[J]. Physica D, 2006, 231(1): 51-65.

[29] ZHU L, CHENG X, YUAN F G. A 3D collision avoidance strategy for UAV with physical constraints[J]. Measurement, 2016, 77: 40-49.

[30] WU Xueli, HUO Jianan, ZHANG Jianhua. Longitudinal minimum interval of aircrafts based on ADS-B monitoring technique[J]. Journal of Hebei University of Science and Technology, 2017, 38(1): 52-58. (in Chinese)

[31] DORIGO M. Message-based bucket brigade: An algorithm for the apportionment of credit problem[J]. European Working Session on Learning on Machine, 1991, 482: 235-244.

[32] SARA P C, EVA B P. Ant colony optimization for multi-UAV minimum time search in uncertain do-

mains[J]. Applied Soft Computing Journal, 2018, 62 (1): 789-806.

[33] LI T, JIANG J, ZHEN Z, et al. Mission planning for multiple UAVs based on ant colony optimization and improved Dubins path[C]//Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC). Nanjing, China: IEEE, 2016: 954-959.

[34] NI Zhuang, XIAO Gang, JING Zhongliang, et al. Path planning method for aircrafts conflict resolution based on improved ant colony algorithm[J]. Journal of Sensors and Microsystems, 2016, 35(4): 130-133. (in Chinese)

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 61773202), the National Key Laboratory of Air Traffic Control (No. SKLATM201706), and the Sichuan Science and Technology Plan Project (No. 2018JZ0030).

Author Prof. **TANG Xinmin** is currently a professor of College of Civil Aviation, Nanjing University of Aeronautics and Astronautics. His research interests include Petri Net and discrete event dynamic system theory, hybrid system theory, and intelligent air traffic system.

Author contributions Prof. **TANG Xinmin** designed the study, complied the models and wrote the manuscript. Ms. **JI Xiaoqi** conducted the analysis. Mr. **LI Teng** contributed to the discussion and background of the study. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

基于改进蚁群算法的无人机冲突解脱技术

汤新民，季小淇，李　腾

（南京航空航天大学民航学院，南京 211106，中国）

摘要：基于ADS-B监视数据，提出了一种基于线性外推法的机载无人机冲突探测与解脱方法，为可能发生的冲突提供预警。为解决多无人机冲突问题，引入基本蚁群算法，并通过加入速度调整策略来优化传统的基本蚁群算法的冲突简化模型。提出了一种基于速度调整策略和航向调整策略的多无人机冲突解脱方案。通过加入排序系统和角度信息，改进了蚁群算法。结果表明，改进的蚁群算法可为空域内多无人机规划无人机冲突解脱路径。改进后的蚁群算法，提高了计算效率，收敛出最优化目标所需时间减少了43.9%，且最终总延误距离减少了58.4%。

关键词：无人机；地面站；蚁群算法；冲突解脱