Journal of Systems Engineering and Electronics
Vol. 34, No. 1, February 2023, pp.197 – 213

UAV penetration mission path planning based on improved holonic particle swarm optimization

LUO Jing [1,2], LIANG Qianchao [1], and LI Hao [2,*]

1. College of Power Engineering, Naval University of Engineering, Wuhan 430033, China;
2. Department of Intelligence, Air Force Early Warning Academy, Wuhan 430019, China

Abstract: To meet the requirements of safety, concealment, and timeliness of trajectory planning during the unmanned aerial vehicle (UAV) penetration process, a three-dimensional path planning algorithm is proposed based on improved holonic particle swarm optimization (IHPSO). Firstly, the requirements of terrain threat, radar detection, and penetration time in the process of UAV penetration are quantified. Regarding radar threats, a radar echo analysis method based on radar cross section (RCS) and the spatial situation is proposed to quantify the concealment of UAV penetration. Then the structure-particle swarm optimization (PSO) algorithm is improved from three aspects. First, the conversion ability of the search strategy is enhanced by using the system clustering method and the information entropy grouping strategy instead of random grouping and constructing the state switching conditions based on the fitness function. Second, the unclear setting of iteration numbers is addressed by using particle spacing to create the termination condition of the algorithm. Finally, the trajectory is optimized to meet the intended requirements by building a predictive control model and using the IHPSO for simulation verification. Numerical examples show the superiority of the proposed method over the existing PSO methods.

Keywords: path planning, network radar, holonic structure, particle swarm algorithm (PSO), predictive control model.

DOI: 10.23919/JSEE.2022.000132

1. Introduction

The modern drone, unmanned aerial vehicles (UAVs), can replace human-piloted vehicles in performing “dull, dirty, or dangerous” missions. Because of their advantages, they have long become the best option for military missions and have been adopted in a wide range of capacities [1−4]. There are numerous civilian, commercial, military, and aerospace applications for UAVs, such as reconnaissance intelligence, situational awareness, combat applications, and assessment of war situations combat applications, and assessment of war situations [5−9]. Notably, trajectory planning is a key point for the UAV that determines its efficiency in task completion [10,11].

A series of achievements have been obtained in the research on drone path planning. The path planning of the UAV encompasses three parts, i.e., environment perception and modeling, optimization of the objective function design, and construction of a solution algorithm. In the part of environment perception, Galceran et al. [12] estimated the obstacle to an ellipse and proposed an elliptic tangent path planning algorithm, and a hardware implementation architecture was built to achieve the flight path planning of the quadrotor drone. Liu et al. [13] systematically summarized coverage path planning (CPP) and its improved algorithms for obstacle modeling and avoidance in the track area, guiding obstacle modeling in this paper. Wu et al. [14] proposed a high-resolution environment modeling method and used the grouped traveling salesman problem for track planning, which improved the sensing capability of the environment. Wai et al. [15] designed an anti-collision method based on Kalman filtering by considering the potential collision of drones with other drones in motion and improving the perception of dynamic targets. In developing the objective function, Li et al. [16] optimized flight energy consumption through improved particle swarm optimization (PSO). Sun et al. [17] proposed an enhanced precise cell decomposition method to optimize the drone’s trajectory and compress its flight time and fuel consumption. Levin et al. [18] proposed a multi-objective differential evolution algorithm to improve the recognition accuracy of UAVs and reduce the distance and potential threats simultaneously. It provides a reference for multi-objective optimization. In terms of solving algorithms, Zhu [19] et al. used rapid-exploring random trees (RRTs) to optimize the track, while Wai et al. [20] adopted the Dubins track planning method. Both methods are mature and stable, but their efficacy in addressing multi-objective and high-dimen-

Manuscript received February 18, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61502522) and Hubei Provincial Natural Science Foundation (2019CFC897).

sional optimization problems is limited. Qie et al. [21] used a shallow network for track planning. However, its optimization ability was defective. Therefore, Yue et al. [22] and Zhang [23] resorted to an improved reinforcement learning method to research track planning under multi-objective conditions, and Duan et al. [24] further enhanced the planning efficiency based on deep reinforcement networks. Additionally, Yang et al. [25] successfully achieved drone track planning on the strength of interactive learning. It is easier for the above-mentioned methods based on reinforcement learning [24,25] to meet the requirements of actual trajectory planning and have good scalability. However, the performance of this type of method is highly dependent on the setting of the reward function. When there is a deviation in the setting of the reward function, the performance of the method decreases significantly. Researchers [26−28] used the improved genetic algorithm, the ant colony algorithm, and PSO algorithm to optimize drone trajectory. Intelligent algorithms are experts in searching for the optimal solution in the high-dimensional space but deficient in their tendency to fall into a local optimum and converge prematurely.

The above methods can be used for reference and promotion for the research on UAV trajectory planning. However, when using drones to penetrate defenses, the drones are required to fly as low as possible to reduce the probability of detection. This requires consideration of factors such as terrain, military threats, and the UAV's motion constraints. Previous researches have covered these elements, but they are not comprehensive and detailed. There are deficiencies in the intelligent optimization algorithm used and there is room for further performance improvement. Especially for the sequential decision-making optimization problem involving a large numbers of parameters, a more efficient optimization algorithm is required. Therefore, based on the existing ideas and results, in this paper we adopt an improved intelligent algorithm to study UAV path planning. In terms of environmental modeling, terrain threat, radar threat, fuel consumption, and time factors of the UAV during the penetration process are quantified. When constructing the multi-objective optimization objective function, the requirements of safety, concealment, and real-time in the penetration process of the UAV are factored in. In terms of constructing optimization algorithms, three shortcomings of the algorithm mentioned in [29] are rectified to improve its efficiency, and the optimization function is solved by incorporating the predictive control model. In the end, the feasibility and superiority of the proposed algorithm are validated through simulation.

2. Problem modeling

In the UAV path planning under penetrating missions, it is necessary to consider its limitations and external factors. One major inherent limitation is the flight constraints of the drone, including speed, altitude, and heading angle, while sudden changes are acceptable. External factors mainly include terrain factors, radar threats, and flight path length. In the process of penetration, we must keep away from mountains and radar radiation. Meanwhile, the time requirement and fuel consumption vary from task to task, which is the total length of the route. Therefore, for the objective function of UAV track planning, all the above factors should be considered.

2.1 UAV self-restraint and flight ability

The drone flies in the mission area and it is regarded as the moving particle in three-dimensional space. Its motion formula [30] is

$$\begin{cases} x(k+1) = x(k) + v(k)\Delta t\cos\varphi(k)\cos\theta(k) \\ y(k+1) = y(k) + v(k)\Delta t\sin\varphi(k)\cos\theta(k) \\ z(k+1) = z_i(k) + v\Delta t\sin\theta(k) \\ v(k+1) = v(k) + \Delta v(k), \quad \Delta v(k) \in [-\Delta v_{\max}, \Delta v_{\max}] \\ \varphi(k+1) = \varphi(k) + \Delta\varphi(k), \quad \Delta\varphi(k) \in [-\Delta\varphi_{\max}, \Delta\varphi_{\max}] \\ \theta(k+1) = \theta(k) + \Delta\theta(k), \quad \Delta\theta(k) \in [-\Delta\theta_{\max}, \Delta\theta_{\max}] \end{cases} \tag{1}$$

where Δt is the time step, $[x(k), y(k), z(k)]$ is the spatial position of the drone at the kth time step; Δv represents the speed variation of the drone; $\varphi(k)$ and $\theta(k)$ represent the heading and pitch angles of the drone, respectively; $\Delta\varphi(k)$ and $\Delta\theta(k)$ are the corresponding angular increments; $v_{\max}$ is the maximum speed of the drone; $\Delta\varphi_{\max}$ and $\Delta\theta_{\max}$ are the maximum angular changes that are bounded by maneuverability.

The hypothetical speed and angle constraints are given in (1). However, in actual situations, the flight speed and altitude of the drone are capped by the quality of the drone, such as thrust and altitude. A method is proposed for determining the theoretical limit and judging whether the track is feasible or not [31]. After the route planning of the UAV, it is essential to determine whether the trajectory is flyable or not.

2.2 Terrain threat

Threat modeling is an unavoidable problem in path planning. Generally, for security concerns, drones are geared to bypassing threatening areas to avoid potential accidents as much as possible. This subsection focuses on the terrain threats to drones during path planning.

When the drone is flying at a low altitude, it is susceptible to collision. The terrain threats are mainly caused by mountain peaks. Generally, the following equation is used to construct the basic threat probability model of the

mountain to the UAV, and then it is superimposed and combined [32] according to the actual terrain:

$$f_i(x(k),y(k)) = \mathrm{e}^{-\lambda[(x(k)-x_{pi})+(y(k)-y_{pi})]} \tag{2}$$

where (x_{pi},y_{pi}) represents the position coordinates of the center point of the ith peak in a two-dimensional plane among I peaks, and λ is a parameter representing the distance of the peak threat. $(x(k),y(k))$ represents the coordinate of the drone at the kth moment.

Assuming that the coordinates of the center point of a mountain peak are (15,15) and λ=1, we illustrate the mountain model of (2) in Fig. 1.

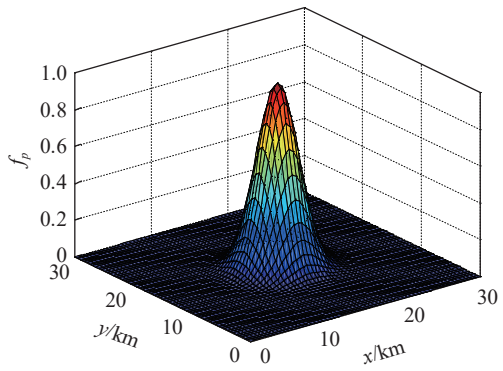


Fig. 1 Three-dimensional mountain peak threat probability model

It can be seen from Fig. 1 that the closer to the center of the mountain peak, the greater the probability of threat.

Different mountain patterns can be formed by adjusting the center position coordinates and parameter λ of the mountain model and combining mountain peaks. In other words, when there are M groups of peaks, $f_1(x(k),y(k))$, $f_2(x(k),y(k)),\cdots,f_m(x(k),y(k))$ are in the space (i,j) and there is an overlap $f_c(x(k),y(k))$, we define the existence of a combined mountain. The values of $f_c(x(i),y(j))$ are calculated via the following formula:

$$f_c(x(i),y(i)) = \max[f_1(x(i),y(i)),f_2(x(i),y(i)),\cdots, f_M(x(i),y(i))]. \tag{3}$$

By (3) and adjusting the peak parameters, the combined peaks can be derived as shown in Fig. 2.

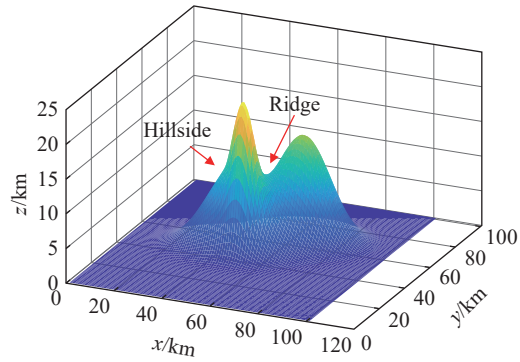


Fig. 2 Schematic diagram of combined mountain

With the direct use of random number generation, the surface generated in this diagram is very different from reality because the random number is independent of one another. This kind of isolation may cause glitches shown in Fig. 2. In fact, the surface can be considered continuous.

Therefore, we propose a method for constructing a surface model of size $n\times n$. First, a random number at (0,0) is generated, which is the height $h_g(0,0)$ of the surface at that point. A zero matrix of $(n+1)\times(n+1)$ order is constructed. Then, it is formed along the diagonal one by one to obtain the surface height. The recursive formula is as follows:

$$h_g(i+1,i+1) = h_g(i,i) + \mathrm{rand}(1),\ \ 1 \leqslant i \leqslant n. \tag{4}$$

Through the above formula, the surface height with a diagonal length of $n+1$ can be shown in Fig. 3.

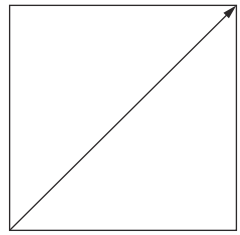


Fig. 3 Schematic diagram of random surface

Subsequently, starting from the points on the diagonals, along the horizontal and vertical directions, the height map of $(n+1)\times(n+1)$ is obtained recursively using (4), as shown in Fig. 4.

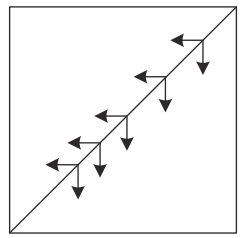


Fig. 4 Global height formation process

Consequently, a height map of $(n+1)\times(n+1)$ is drawn. Finally, perform a smooth height map operation. Starting from the upper left corner, the average value of the four points is the value of the lower left corner, namely

$$h_{gn}(i,j) = \frac{1}{4}h_g(i,j) + h_g(i+1,j) + 1 + h_g(i,j+1) + h_g(i+1,j),\ \ 1 \leqslant i,j \leqslant n. \tag{5}$$

The algorithm repeats this process until it reaches the last node. The process is illustrated in Fig. 5.

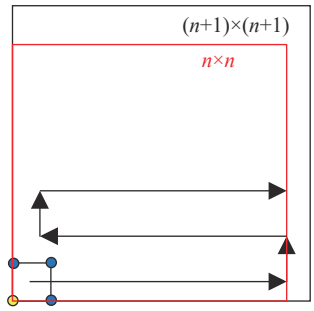


Fig. 5 Height line smoothing process

The four blue points of the small square in the lower left corner are averaged to get the point in the lower left corner of the small square denoted by the yellow dot in Fig. 5. Afterwards, it recurs in order until a smooth height line of $n \times n$ is generated, as shown in Fig. 6. Comparing Fig. 2 with Fig. 6 indicates that the height line aligns with the actual situation to a greater extent.

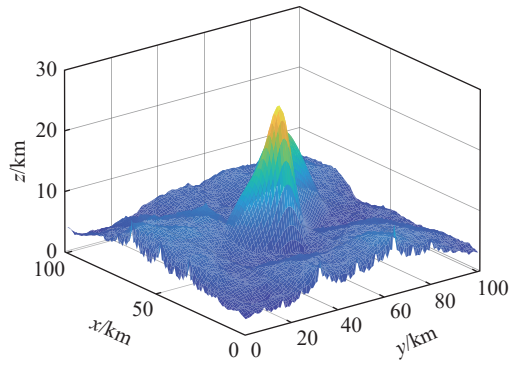


Fig. 6 Surface height map after smoothing

The combined mountain and surface height maps through the above process framework are closer to the actual situation. It can be used to simplify the construction of three-dimensional maps and further research on trajectory planning.

As the distance gets shorter, the time left for the drone to bypass the mountain by changing its trajectory decreases, and thus the difficulty of avoiding potential threats and the probability of collision for the drone increase. Suppose that the drone is threatened at the time k, the terrain threat $\mathrm{Th}_e(k)$ is quantified as follows:

$$\mathrm{Th}_e(k) = \max_{1 \leqslant i \leqslant j} \{f_j(x(k), y(k))\}. \tag{6}$$

The total terrain threat to the UAV can be calculated using the following formula:

$$\mathrm{Th}_e = \sum_{k=1}^{K} \mathrm{Th}_e(k). \tag{7}$$

To sum up, in path planning, the further the distance, the less threatened for the drone by the terrain during the flight.

2.3 Radar threat

2.3.1 Radar equation analysis

Radar is the main equipment for detecting and finding targets in air defense systems and navigating firepower such as air defense missiles to attack targets. In the low-altitude penetration process of the UAV, the major threat comes from the enemy radar network. Unlike a single radar, the radar network is required to cover all the protected areas at the beginning of the deployment.

Therefore, in the course of the penetration, the drone is likely to be within the detection range of the enemy radar and scanned by the radar beam. However, the fact that the drone is within the radar detection range or scanned by the radar beam does not mean that it can be detected by the enemy radar. If the echo of the drone received by the enemy radar receiver is slight or even below its detection threshold, the drone should not be detected.

The classic radar formula is

$$P_r = \frac{P_t G^2 \lambda^2 \sigma F^4}{(4\pi)^3 R^4 C_B L} \tag{8}$$

where P_r represents the radar echo received by the radar, P_t and G are the transmit power and the corresponding antenna gain of the transmitter, λ is the operating wavelength of the radar, σ is the radar cross section (RCS) of the drone, F is the transmitting or receiving pattern propagation factor, R is the distance from the drone to the radar, C_B is the matching coefficient between the filter and the signal waveform, and L is the loss factor of the electromagnetic wave. If P_r is small, it is difficult to effectively detect or identify the state parameters of the drone.

Next, we analyze the factors that affect P_r. In (8), P_t, G, λ, F, and C_B depend on the enemy radar and do not change usually. If the corresponding relationship of these parameters in (8) is denoted as Q, then (8) can be reformulated as

$$P_r = Q \frac{\sigma}{R^4}. \tag{9}$$

That is, the echo P_t is related to σ and R. R can be calculated by the distance equation above, and the RCS of the drone is analyzed in the next subsection.

2.3.2 Analysis of UAV RCS

The situation diagram in the penetration of the UAV is shown in Fig. 7.

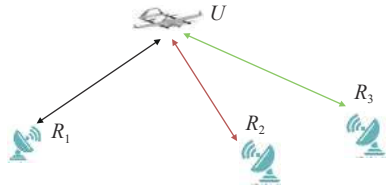


Fig. 7 Space situation map of UAV and networking radar

Because the relative position of the UAV and the networking radar is different, the corresponding RCS areas are also different when different radars irradiate the UAV. Assume that there are a total of M radars for networking, the RCS of the mth radar at time k can be denoted as $\sigma_m(k)$. Since the drone is completely known, the RCS distribution map of the drone under different wavebands and angles can be drew with certain methods.

In order to analyze RCS, the geometric combination model method can be used for simple targets, while the flat triangle triangular element model and parametric surface model are usually for complex geometry. Based on fitting the geometric shape of the UAV, we use computer-aided drafting (CAD) software for segmentation to generate accurate geometric models. First, a collective coordinate system corresponding to the UAV model and definition is built, as displayed in Fig. 8.

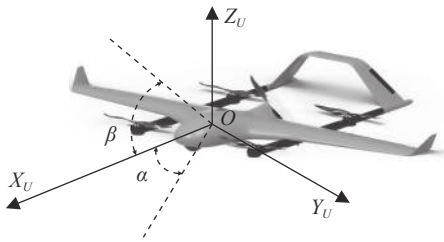
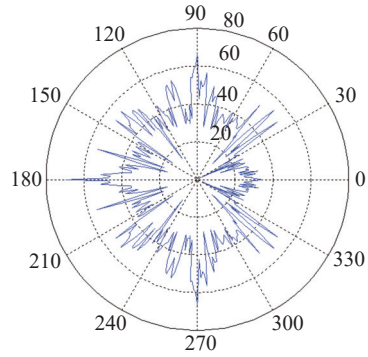


Fig. 8 UAV model and body coordinate definition

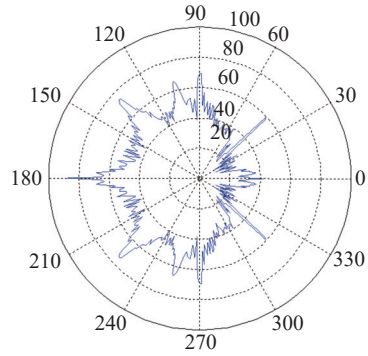
Let the mass center of the drone be the origin O of the coordinate system, the X_U axis be the origin pointing in the direction of the nose, and the Z_U axis is in the vertical plane of the drone and perpendicular to the X_U axis. Then, a collective coordinate system is constructed to decompose the enemy radar detection beam. The radar beam is projected onto the X_UOY_U plane. The angle between the radar beam and the OX_U axis is the azimuth angle α. The projection onto the X_UOZ_U surface is the elevation angle β. This can represent the three-dimensional detection angle of the enemy radar beam.

2.3.3 Radar networking threat metric

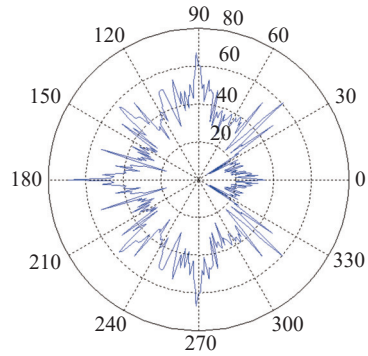
The RCS of UAV can be obtained through hypothetical formula and real exposure to radar. In the former method, modeling and formulation are complicated, so the latter is more widely used for radiation measurement. The RCS varies significantly with radar signals in different frequency bands and in different bandwidths. In case of actual measurement, the radars in all frequency bands need to be tested, which is not effective. To this end, the built-in algorithm of FEKO, a 3D full-wave electromagnetic simulation software, is used to combine the UAV space model constructed in the previous section to simulate the RCS distribution map of the UAV. This method is relatively mature and can be directly achieved through Matlab simulation. The typical simulation results are presented in Fig. 9.



(a) RCS distribution chart when β=−5°



(b) RCS distribution chart when β=0°



(c) RCS distribution chart when β=5°

Fig. 9 RCS distribution of UAVs at different pitch angles

After obtaining the spatial distribution information of enemy radars, the UAV and each radar in the radar network can be used to calculate the mth radar irradiated

unmanned at the kth time in planning the drone's track. The radar's cross-sectional area is $\sigma_m(k)$. Therefore, at the kth moment, for the mth radar in the radar network, (8) can be reformulated as follows:

$$P_r^m(k) = Q_m \frac{\sigma_m(k)}{R_m^4(k)} \tag{10}$$

where Q_m is determined based on the radar's parameters, independent of time. Based on the coordinates of the mth radar and the position of the drone, $\sigma_m(k)$ and $R_m(k)$ can be determined to calculate $P_r^m(k)$. Receiving a radar echo does not mean that a radar signal can be detected. When the energy of the echo is lower than the radar detection threshold S_m, it is considered that no target is detected. $D_m(k)$ is formulated to measure the detection capability of the mth radar as follows:

$$D_m(k) = \begin{cases} \dfrac{P_r^m(k)}{S_m}, & P_r^m(k) \geqslant S_m \\ 0, & P_r^m(k) < S_m \end{cases}. \tag{11}$$

At the same time, considering that low-altitude penetration is the use of the curvature of the Earth to enhance the radar blind zone, the detection limit distance $R_{\lim}^m(k)$ of the mth radar at time k can be formulated based on the empirical formula as follows:

$$R_{\lim}^m(k) = 4.12\left[\sqrt{z_m} + \sqrt{z(k)}\right] \tag{12}$$

where z_m is the height of the radar in the three-dimensional space, and $z_m(k)$ is the height of the drone. Both units are m. The detection limit distance $R_{\lim}^m(k)$ is measured in km. That is, when the height of the radar and the drone is determined, the maximum detection distance of the radar can be derived. When the distance $R_m(k)$ between the drone and the radar is greater than this limit distance, the radar will not detect the drone. Then (13) can be formulated as follows:

$$D_m(k) = \begin{cases} \dfrac{P_r^m(k)}{S_m}, & P_r^m(k) > S_m;\ R_{\lim}^m(k) \geqslant R_m(k) \\ 0, & P_r^m(k) \leqslant S_m;\ R_{\lim}^m(k) < R_m \end{cases}. \tag{13}$$

Because the enemy radar uses a networking mechanism, once the drone is detected by a certain radar, the drone information will be captured by the entire radar network. Therefore, the radar network's detection capability for the UAV depends on the radar with the strongest detection capability among M radars. This statement can be expressed in the following formula:

$$\mathrm{Th}_r(k) = \max_{1 \leqslant m \leqslant M} [D_m(k)]. \tag{14}$$

In the path planning, the total threat of the drone to the enemy radar network is the sum of the threats to all planning points, namely,

$$\mathrm{Th}_r = \sum_{k=1}^{k} \mathrm{Th}_r(k). \tag{15}$$

In the path planning process, the radar threat should be minimized.

2.3.4 Fuel consumption factor

In path planning, the planned path length $s(k)$ at time k can be expressed as follows:

$$s(k) = \sqrt{[x(k+1)-x(k)]^2 + [y(k+1)-y(k)]^2 + [z(k+1)-z(k)]^2}. \tag{16}$$

In (16), it is assumed that the speed of the drone during flight is approximately constant, and that the spacing points are dense during the path planning process. Therefore the flight trajectory of each UAV is approximately a straight line. The corresponding fuel consumption depends on the flight path and status of the drone. There are three UAV motion states, including ascent, level flight and descent. Based on the empirical formula, the fuel consumption can be expressed as follows:

$$\begin{cases} C_a(k) = \eta_a \Delta t p_a \\ C_l(k) = \eta_l \Delta t p_l \\ C_d(k) = \eta_d \Delta t p_d \end{cases} \tag{17}$$

where C represents fuel consumption, a, l, and d represent the three motion states of ascent, level flight, and descent, η represents the thrust to weight ratio, p represents the thrust of the engine in the corresponding state, and $v(k)$ represents the flight speed of the drone at this moment. Thus, the fuel consumption C of the path at this stage can be calculated.

Additionally, the total fuel consumption C_T can be expressed as follows:

$$C_T = \sum_{k=1}^{K-1} C(k) \tag{18}$$

where $C(k)$ represents the fuel consumption in each flight status at time k in (17).

After obtaining the specific and environmental parameters of the drone, the differential equations of the drone flight formulated by the fourth-order Runge-Kuta method can be used to calculate the drone flight fuel consumption more accurately. This paper only proposes a more general and fast method to calculate the fuel consumption of the drone.

For time-constrained tasks or time-sensitive targets, it is required to keep the penetration time of the drone as short as possible. For tasks without tight time limits, the

consideration for time can give way to safety. The longer the total length of the trajectory, the higher the level of corresponding energy consumption. This means that the larger the load of the drone, especially when the total load of the drone is fixed, the more the energy consumption load and the the less the mission load, thus affecting the performance of drones. Similarly, if the drone's load is large enough, consideration for fuel consumption can give way to safety.

2.3.5 Path planning objective function

Subject to the constraints of the UAV kinematics model, the objective function J of the path planning can be expressed as follows:

$$J = \omega_e \mathrm{Th}_e^n + \omega_r \mathrm{Th}_r^n + \omega_C C_T^n \tag{19}$$

where Th_n^e, Th_r^e, and C_T^n are the normalized metric values of the terrain threat, radar threat, and fuel consumption factor, respectively. The normalization algorithm and corresponding value methods are explored later. ω_e, ω_r, and ω_C are the weights of the corresponding terms, and

$$\omega_e + \omega_r + \omega_C = 1. \tag{20}$$

These three weights focus on describing security, concealment, and fuel consumption, respectively. The weight value can be calculated by combining certain weight determination methods based on the actual task requirements. However, the details of this method are not explored in this paper.

After weighing the importance of each factor to determine the corresponding weight value, the corresponding objective function can be defined as follows:

$$J = \min\left(\omega_e \mathrm{Th}_e^n + \omega_r \mathrm{Th}_r^n + \omega_C C_T^n\right). \tag{21}$$

The optimization goal of (21) is to find a set of trajectories, so that the weighted sum of the three factors, namely, threats, concealment, and flight distance, is minimized during the flight of the drone.

3. Improved holonic PSO (IHPSO)

Optimal trajectory planning is a typical NP-hard problem, which can be solved by using intelligent algorithms. To this end, this paper improves the PSO algorithm based on the synthesizing structure to figure out the optimal solution to the path planning problem. This section introduces the architecture and flow of the particle swarm algorithm based on the holonic structure, and probes into the corresponding improvements.

3.1 Holonic structure algorithm flow

The algorithm architecture and main flow of holonic PSO (HPSO) in [29] can be simplified as Fig. 10.

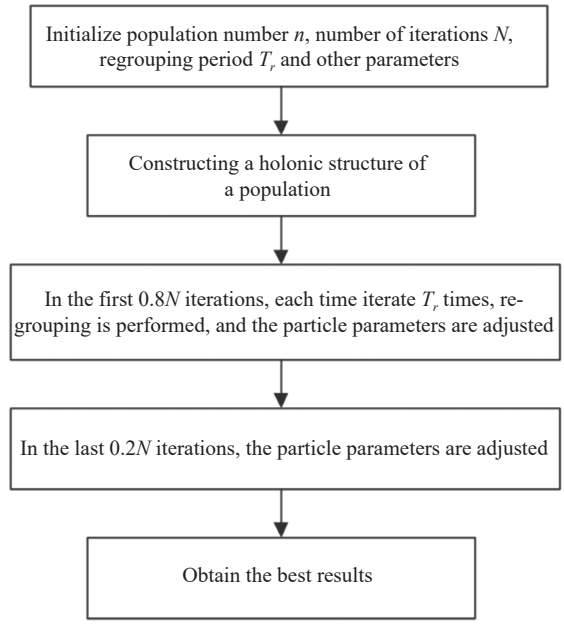


Fig. 10 HPSO algorithm flow

The above process can be summarized as the following steps.

Step 1 In the initialization PSO algorithm, configure the number of population n, the number of iterations N, and the repacking period T_r.

Step 2 Construct the initial synergy structure. Set the number of layers in the structure and the number of groups per layer, and randomly group the particles.

Step 3 The first $0.8N$ iterations are mainly to achieve a wide range of searches and prevent particles from falling into the local best advantage. In [29], through several comparisons and attempts, the author inferred that the ratio of 0.8 is the best ratio for this algorithm. Each time a regrouping period T_r is entered, the particles are regrouped. The rule is that after each grouping, two particles are randomly selected from the group. The two chosen particles can be from different groups.

In the first $0.8N$ iterations, the particle parameters are adjusted using the following equation to obtain the optimal fitness function:

$$\begin{cases} \mathrm{Vel}^{t+1} = w \cdot \mathrm{Vel}^t + \sum_{i=1}^{\text{Level number}} c_i \cdot r_i \cdot \\ \qquad \dfrac{\text{level number} - i + 1}{\text{level number}} \cdot (p\mathrm{Best}_\mathrm{i} - \mathrm{Position}) \\ \mathrm{Position}^{t+1} = \mathrm{Position}^t + \mathrm{Vel}^t \end{cases} \tag{22}$$

where Vel represents the velocity of the particles, t is the number of iterations, ω is the inertia factor, c_i is the acceleration coefficient of the ith layer, r_i is a random number uniformly distributed in [0,1], and level number is the number of layers of the structure, $p\mathrm{Best}_i$ represents the position of the optimal particle in the ith layer, and

Position^t is the position of the particle after t iterations.

Step 4 Generally, after the first $0.2N$ searches, the range in which the optimal solution may appear can be determined. Therefore, the last $0.8N$ searches are mainly to obtain the optimal solution within a small range of accurate searches. Therefore, it is unnecessary to regroup and use the following formula for fine search:

$$\begin{cases} \text{Vel}^{t+1} = w \cdot \text{Vel}^t + c_1 \cdot r_1 \cdot (p\text{Best}_1 - \text{Position}) - \\ \quad \sum_{i=2}^{\text{level number}} c_i \cdot r_i \cdot \dfrac{\text{level number} - i + 1}{\text{level number}} \cdot \\ \quad (p\text{Best}_i - \text{Position}) \\ \text{Position}^{t+1} = \text{Position}^t + \text{Vel}^t \end{cases} \tag{23}$$

where the parameters are the same as those in Step 3.

Step 5 Get the optimization results of the HPSO algorithm.

The above process is the HPSO optimization process. Since [29] only uses the synthesizing structure to improve the PSO algorithm for the first time, there is still the possibility of improvement in some aspects. Therefore, this paper further explores it.

3.2 HPSO algorithm improvements

3.2.1 Grouping strategy

In the HPSO algorithm, Roshanzamie et al. [29] used approximately equal division to construct particle groups. This grouping method may be the most convenient, but not necessarily the best. At the same time, Roshanzamie et al. [29] argued in the conclusion that "it is also possible to create groups by special rules to achieve specified purposes". To this end, this paper proposes two completely opposite grouping strategies based on systematic clustering and information entropy, then analyzes and compares these two grouping strategies.

The systematic clustering method calibrates the similarity of particles according to the distance between each two particles, and the effect is intuitive and obvious. Through the system clustering method, particles with a short distance or a high degree of similarity can be assigned into a group. That is, this classification method can obtain a local area constructed by several particles, and it is easy to find the local optimum of a certain area. In this way, the optimization problem is converted into a dynamic adjustment grouping method, and the local area is optimized, and eventually the overall optimization process is performed. This method is commonly used in statistics. The algorithm flow and software implementation are relatively mature, and there is no need for further discussion here.

Information entropy is a concept in information theory that describes the probability of different states of the system. In this way, the probability that the particles fall into the global optimum is smaller. This grouping method is to determine the number of groups and maximize the information entropy in each group. By dividing the particles that are farthest from each other into one, the total information entropy of the system is the largest.

Under the premise of setting the number of groups, the group with larger information entropy will have more possibilities. Thus the probability that particles falling into a local optimum is lower. Assume that n elements are divided into Q groups, a total of P grouping methods are set, the information entropy in each group of each classification method is $E_{Z1}, E_{Z2}, \cdots, E_{ZQ}$, and the information entropy is the largest

$$\max_{z \in P} \left(\sum_{s=1}^{Q} E_{zs} \right). \tag{24}$$

This classification method has the largest space for exploration, and the probability of falling into a local optimum is significantly reduced.

Apparently, (24) is also an optimization problem. If all possibilities are traversed, the amount of calculation will increase. Therefore, this paper uses an approximate method. The larger the information entropy of a group, the weaker the relationship between every two particles and the greater the distance between them. Just in the opposite of the systematic clustering method, the systematic clustering method groups the particles with a short distance into one class, and the information entropy groups multiple particles with a long-distance together. Therefore, in the system clustering method, the elements in the distance matrix are all inverses, while the other operations are precisely the same, and the particles that are the farthest from each other can be clustered into one class. At the same time, the amount of calculations is significantly reduced.

In Subsection 3.1, (22) and (23) can be taken as two search strategies, i.e., broad search and accurate search, and the balance between the speed and the accuracy of the algorithm.

Because both search strategies involve grouping particles, and combining the two grouping methods proposed in this paper with the two search strategies produces four results, as shown in Table 1.

Table 1 Grouping and search strategy combination

Scheme	First stage extensive search	Second stage accurate search
Scheme 1	System clustering method	System clustering method
Scheme 2	Information entropy method	Information entropy method
Scheme 3	System clustering method	Information entropy method
Scheme 4	Information entropy method	System clustering method

This paper proposes two opposite grouping strategies with the object to improve the performance of the algo-

rithm. If the efficiency of wide-area search or accurate search is higher, the algorithm performance will be enhanced. Therefore, by the fourth strategy, namely, "information entropy in extensive search stage, and systematic clustering method in accurate search stage ", a wider range when searching in a wide area and a more accurate search when performing an accurate search can be made.

3.2.2 State transition condition

Let the trend of the absolute value of the slope of the fitness function (slope described later) be

$$k_i = |J_{\text{fit}}(i) - J_{\text{fit}}(i-1)| \tag{25}$$

where i represents the number of iterations, $J_{\text{fit}}(i)$ represents the fitness function value of the ith iteration, and k_i represents the absolute value of the change rate of the fitness function. $|\cdot|$ means taking the absolute value.

k_i can reflect the search status of particles. At the initial stage, the fitness function value decreases rapidly, and no switching is required. When the particle searches for a local or global optimum, the fitness function and the rate of change as a whole decreases gradually, which can be switched to an accurate search. To this end, conditions are factored into the following expression to judge the switch from broad search to accurate search described as

$$\frac{k_{i-2j} - k_{i-j}}{k_{i-2j}} < \omega \frac{k_{i-j} - k_i}{k_{i-j}} \tag{26}$$

where $(k_{i-j} - k_i)/k_{i-j}$ represents the ratio between the jth iteration fitness function change amount $(k_{i-j} - k_i)$ and the original one k_{i-j}. As the particles approach the optimal solution, this ratio increases progressively, that is, it decreases more and more slowly. ω is the weight coefficient. Generally, the value is within [0.8,0.9], and j may be a smaller positive integer. Because of the judgment conditions of reverse switching, which is discussed later, the values of ω and j can be arbitrary, even if the initial value is not good. It can also be switched back through the reverse switching strategy. The two complement each other and improve the efficiency of optimization, which is another great advantage of this method.

Considering that the local optimization may occur during the search process, which means that the fitness function may be 0, and the denominator of (26) is invalid, thus (27) is used to determine the actual optimization process:

$$k_{i-j}(k_{i-2j} - k_{i-j}) < \omega k_{i-2j}(k_{i-j} - k_i). \tag{27}$$

The right side of (27) is the multiplication form, which can be switched by satisfying (27).

However, there is a situation where (22) is satisfied, that is, the handover condition is satisfied, but in fact, the handover should not be performed at this time. That is, if the $(i-2j)$th and the $(i-j)$th searches fall into the local optimum, the fitness functions of the two times are similar, and the difference is small. In this way, the difference in the parentheses on the left side of the inequality in (22) will be small. It is easy to cause the switching strategy to be adopted regardless of the calculation result on the right. And if the algorithm has already jumped out of the local optimum during the ith search, there is no need to switch at this time.

Therefore, in order to avoid the above situation, a discriminant condition should be added before (22) to determine whether to jump out of the local optimum, namely,

$$k_1 < \alpha k_{i-j} \tag{28}$$

where α can be a positive integer between 2 and 10. If the local optimum is jumped out, the change rate of the fitness function will increase sharply, which is significantly larger than the previous slope. Equation (28) is used as the first judgment condition, and (27) comes to the second. At this time, the system switches to the accurate search mode based on system clustering, otherwise, it continues to perform extensive search.

3.2.3 Adaptive termination condition

The performance of the HPSO algorithm is correlated with the number of iterations. If the number of iterations set is too much, it will inevitably lead to a heavy calculation burden for the algorithm. On the other hand, if the number of iterations is too small, the algorithm will end prematurely and will probably not converge to the better solution. At the same time, for different problems and fitness functions, different calculation accuracy and requirements, the number of iterations of the algorithm may be vastly distinct. It is difficult to estimate the specific problem without any prior knowledge or experimental basis. The number of iterations severely limits the scope of the algorithm.

In addition to the fitness function, the position of the particles is constantly changing. The traditional method is that when the particles all converge to a point, that is, the coordinates of all particles are almost identical, the algorithm ends. In this way, it is difficult to ensure whether it converges to the global optimum. For this reason, this paper creates discriminant conditions based on the dynamic position of the particles to achieve a system state switch.

First of all, a double discrimination condition is constructed. Whether the particle converges to the global or local optimum, its fitness function no longer decreases, and the corresponding slope is 0. The first discrimination condition is as follows:

$$k_i = 0. \tag{29}$$

If the condition of (29) is not satisfied, the system will

continue to perform an accurate search; otherwise, the next determination will be made. When the condition of (29) is satisfied, however, according to the previous classification method of system clustering and information entropy, the particles are classified again based on information entropy, and only the classification is not searched. Define the distance sum in each group as SD_p, that is,

$$\mathrm{SD}_p = \mathrm{sum}\{d_{ij}\}, \quad p \in [1, z]. \tag{30}$$

Suppose that the information entropy grouping $\{\mathrm{SD}_{pe1}, \mathrm{SD}_{pe2}, \cdots, \mathrm{SD}_{pez}\}$, the distance and the smallest is $\mathrm{SD}_{pe\min}$, and the system clustering method is $\{\mathrm{SD}_{ps1}, \mathrm{SD}_{ps2}, \cdots, \mathrm{SD}_{psz}\}$, the distance and the largest are $\mathrm{SD}_{ps\max}$, that is,

$$\mathrm{SD}_{pe\min} = \min\{\mathrm{SD}_{pe1}, \mathrm{SD}_{pe2}, \cdots, \mathrm{SD}_{pez}\}, \tag{31}$$

$$\mathrm{SD}_{ps\max} = \max\{\mathrm{SD}_{ps1}, \mathrm{SD}_{ps2}, \cdots, \mathrm{SD}_{psz}\}. \tag{32}$$

The second-level discrimination condition is

$$\mathrm{SD}_{p\min} \leqslant \mathrm{SD}_{ps\max}. \tag{33}$$

That is, when the distance sum of the closest grouping based on the information entropy method is not greater than the most extensive grouping based on the system clustering method, it can be determined that the particles are converged to the global good value, and the algorithm ends.

The discriminant condition of (29) is to show that all particles are aggregated within a certain area. At this time, the search is only to find the best in this area. The subsequent search strategy is not changed, and the accurate search strategy is adopted until all particle positions are gathered at one point, and this process is called convergence. This method can achieve autonomous convergence and better adaptability.

3.3 IHPSO algorithm flow

This subsection constructs an IHPSO algorithm flow as shown in Fig.11.

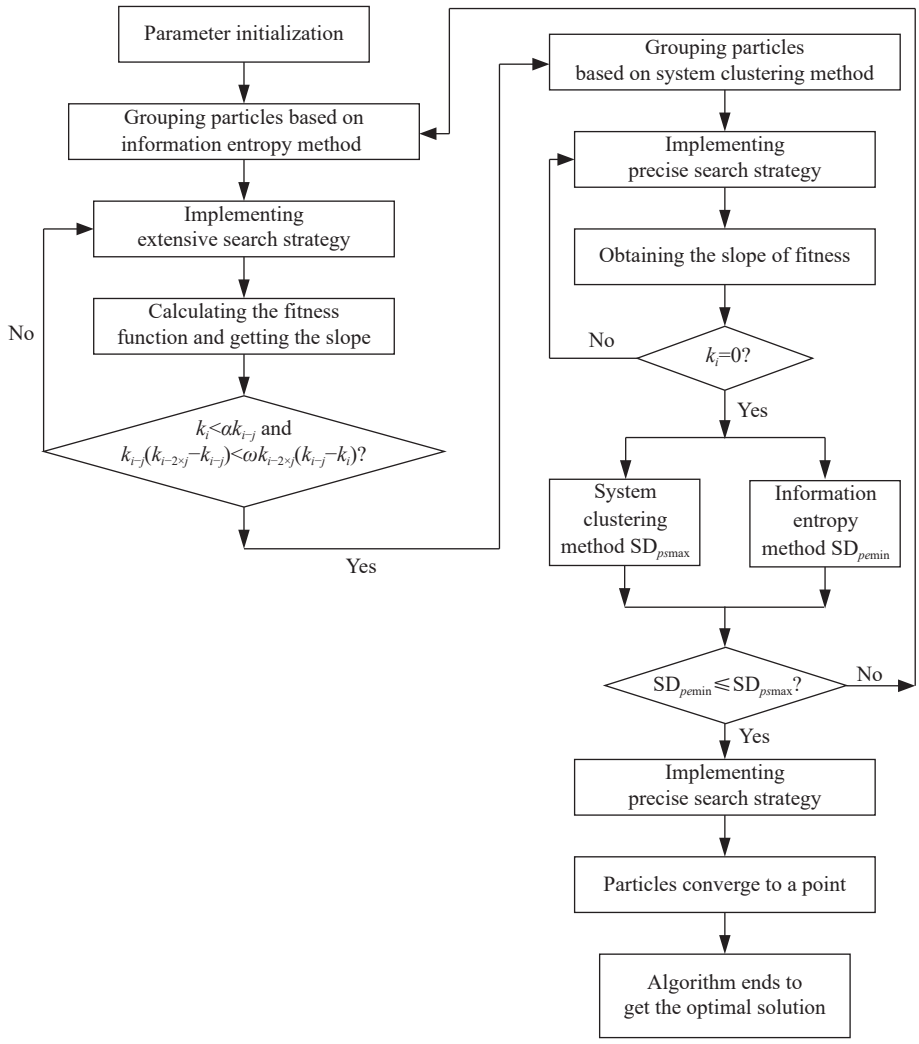


Fig. 11 IHPSO algorithm flow

The above flowchart can be detailed into the following steps.

Step 1 Initialize global terrain information, radar position information, and landing waypoints of the drone.

Step 2 Use information entropy to group the particles, start employing a broad search strategy, and calculate the fitness function and slope.

Step 3 Use (27) and (28) to make a judgment. Whether one of the or both conditions are not satisfied, get back to Step 2. When both conditions are satisfied, execute Step 4.

Step 4 Use systematic clustering to group particles.

Step 5 Execute the exact search strategy and calculate the fitness function and slope.

Step 6 Determine whether the slope of the fitness function is 0. If it is not 0, return to Step 5. Otherwise, execute Step 7.

Step 7 Two methods are used to group the particles which remain motionless. $\mathrm{SD}_{pe\min}$ and $\mathrm{SD}_{ps\max}$ are calculated and their relative sizes are determined. When $\mathrm{SD}_{pe\min}$ is greater than $\mathrm{SD}_{ps\max}$, the algorithm returns to Step 2. Otherwise, the search strategy is no longer switched, and an accurate search is performed until all particles converge to a point.

Step 8 Get the optimal worthy coordinates, which is the optimal solution, and the algorithm ends.

Through Step 3 and Step 7, the switching of the system state is achieved. The system search strategy is adapted to the actual state of the particles. More importantly, the number of iterations is not required to be set in advance.

4. Solving UAV path planning process using IHPSO algorithm

Based on the objective function of the UAV track planning and the IHPSO algorithm constructed earlier, this section explores the process of using the improved algorithm to optimize and solve the optimal track.

4.1 Predictive control model

The drone can be regarded as a control system. According to (1), the state of the drone can be described as $\boldsymbol{S}(k)=[x(k),y(k),z(k),\phi(k),\theta(k)]$, and the formula of state is as follows:

$$\boldsymbol{S}(k+1)=f(\boldsymbol{S}(k),\boldsymbol{u}(k)) \tag{34}$$

where $\boldsymbol{u}(k)=[\Delta v(k),\Delta\phi(k),\Delta\theta(k)]$ is the input of the system and $f(\cdot)$ is the state transition function of the system. Starting from time k, given the H-step control input, the trajectory of the UAV within the H-step can be predicted. By adjusting the input of the system, the drone's trajectory is planned, and the UAV penetration, which takes safety, concealment, and failure into consideration, is achieved. The prediction model of the system is as follows:

$$\begin{aligned}&\boldsymbol{S}(k+j+1\mid k)=f(\boldsymbol{S}(k+j\mid k),\boldsymbol{u}(k+j\mid k)),\\&\qquad j=0,1,\cdots,H-1,\end{aligned} \tag{35}$$

where H is the prediction period, and $\boldsymbol{S}(k+j+1|k)$ is the state of the prediction system at time $k+j+1$. At the same time, this state depends on the system state $\boldsymbol{S}(k+j+1|k)$ and the control input $\boldsymbol{u}(k+j+1|k)$, which is a recursive state.

Assume that the system state at time k is $\boldsymbol{S}(k)$, then the input of H-step predictive control is as follows:

$$\boldsymbol{U}(k)=[\boldsymbol{u}(k),\boldsymbol{u}(k+1),\cdots,\boldsymbol{u}(k+H-1)]. \tag{36}$$

The cost function of the system after H-step prediction is expressed as

$$J^{(H)}(\boldsymbol{S}(k),\boldsymbol{u}(k))=\sum_{j=0}^{H-1}\alpha^{j}J(\boldsymbol{S}(k+j\mid k),\boldsymbol{u}(k+j\mid k)) \tag{37}$$

where α is the discount factor and represents the importance ratio between the current path planning cost and the long-term cost.

Then the optimization model corresponding to the optimal input of the system at time k can be expressed as follows:

$$\begin{aligned}&\boldsymbol{U}^{(H)}(k)=\arg\min J^{(\mathrm{H})}(\boldsymbol{S}(k),\boldsymbol{U}(k))\\&\text{s.t.}\begin{cases}\boldsymbol{S}(k+1+j\mid k)=\\\qquad f(\boldsymbol{S}(k+j\mid k),\boldsymbol{u}(k+j\mid k))\\\boldsymbol{S}(k\mid k)=\boldsymbol{S}(k)\end{cases}\end{aligned} \tag{38}$$

where $\boldsymbol{U}^{(H)}$ is the optimal control input of the system and the optimization function in this paper.

In offline planning, H can be set to a large value, which is much larger than the upper planning time limit K, that is, the entire trajectory of the drone is directly planned. In this way, it is possible to obtain the best trajectory, and the calculation burden is extremely heavy. If the actual computing resources and planning time constraints are excluded, this method is completely feasible. However, in actual operation, this condition is unlikely to be satisfied, even the on-board computer of the drone is used for self-planning. Therefore, a smaller H can be set according to actual conditions, which can be set to 3, 5, 10, and other parameter values, and real-time planning, that is, predicting five steps and executing one step repeatedly until the track is obtained. This method can obtain a track that is

feasible but probably not optimal. The specific settings are implemented according to actual requirements and constraints.

In summary, using the previous IHPSO algorithm, ΔV, $\Delta\phi$, and $\Delta\theta$ are recursively optimized in step H until the track is derived.

4.2 Algorithm flow

Based on the IHPSO algorithm for track planning, the algorithm flow is illustrated in Fig. 12.

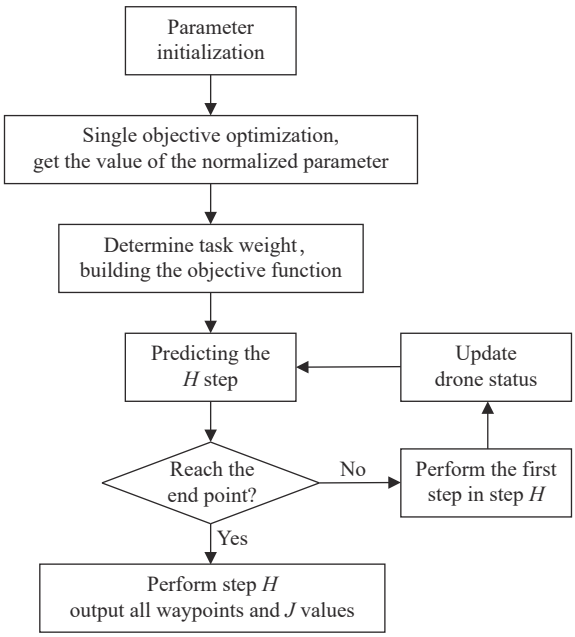


Fig. 12 Track planning process based on IHPSO algorithm

This flowchart can be detailed into the following steps:

Step 1 Initialize the global terrain information, the radar position information, and landing waypoints of the drone.

Step 2 Normalize the parameters. Let ω_e, ω_r, and ω_s be 1,0, and 0 in sequence. When only terrain threats are considered, take H=10 for 20 times of track planning, record these 20 plans, and record the smallest single terrain threat in multiple H- step predictions $\mathrm{Th}_{e\min}$, with maximum radar threat $\mathrm{Th}_{r\max,e}$ and maximum path length $s_{\max}$.

Step 3 Similarly, let ω_r and ω_s be 1, and the other two weights be 0, and plan for 20 times. Finally, the minimum value of the three sets of parameters, ($\mathrm{Th}_{e\min}$, $\mathrm{Th}_{r\min}$, $s_{\min}$), and the maximum values of the three sets of parameters $\{(\mathrm{Th}_{e\max,r},\mathrm{Th}_{e\max,s})$, $(\mathrm{Th}_{r\max,e},\mathrm{Th}_{r\max,s})$, $(s_{\max,e}, s_{\max,r})\}$, whichever is the greater of each group, are used to obtain the maximum value of the three groups of parameters ($\mathrm{Th}_{e\max}$,$\mathrm{Th}_{r\max}$,$s_{\max}$). After getting the maximum value of the parameters, normalize the parameters, that is,

$$x^n = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{39}$$

In (39), x^n is the normalized parameter, and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the above parameters, respectively. In this way, the normalization process of the three variables can be realized, the dimensions of the three variables are flattened, and the target constituted by the three variables is optimized.

Step 4 Determine three weights according to the task requirements, construct the objective function, initialize the time k = 1, and set H.

Step 5 Based on predictive control, use the IHPSO algorithm to optimize and get the optimal result in step H.

Step 6 Determine whether the last step of step H is the end of the track. If so, execute all steps, and output all track points and the optimized objective function J. Otherwise, go to Step 7.

Step 7 Execute the first step in the optimal H-step planning, update the drone status, and return to Step 5.

Step 8 Repeat Step 2−Step 7 for 10 times to get 10 sets of optimized trajectory and corresponding objective function J.

Step 9 Find the track corresponding to the smallest J, which is the planned track.

Two points in the above process need further explanation. The first is to use Step 2 and Step 3 to determine the normalized maximum value of the parameters. A more general method does not involve fixed parameter values, especially for uncontrolled problems as track planning. Due to the threat of natural mountains and rivers, the deployment of enemy radar is not supposed to be adjusted by our path planning, which means it is impossible to characterize these parameters with universal parameters under various and dynamic conditions of environment and mission background. The quality of parameter normalization seriously affects the final optimization result. Therefore, a method is required to determine the corresponding parameters according to the background factors. Although the method proposed in this paper consumes computing resources, it allows parameter adjustments according to the actual environment. Therefore, this complicated operation is needed.

Another point is the repeated planning of Step 8 and Step 9. The number of repetitive plans set in this paper is not large, and its calculation amount is negligible compared with the workload of making H much larger than K. Because the intelligent algorithm is used for optimization, the algorithm may be precocious or fall into a local optimum. Therefore, based on the ideas of Monte Carlo experiment, by conducting multiple experiments and taking the global good value of multiple experiments as the optimization result, efficiency of track planning can be

improved.

4.3 Algorithm logic block diagram

The logic of trajectory optimization constructed in this paper is shown in Fig. 13.

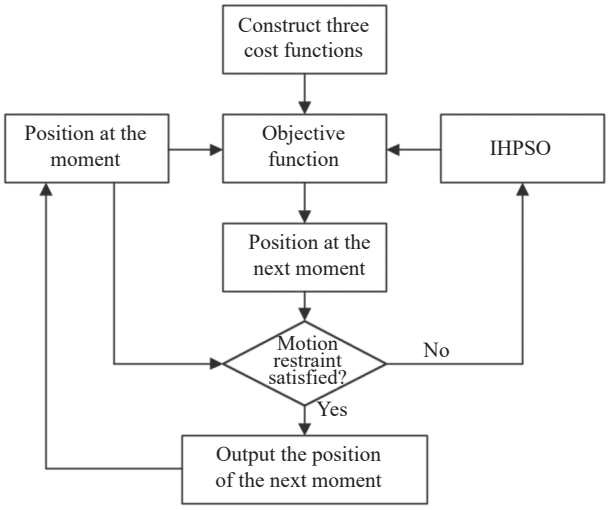


Fig. 13 Algorithm logic diagram

The input of the trajectory planning algorithm constructed in this paper is the space position coordinates and environmental parameters of the UAV at this moment. The output, that is, the decision variable of this algorithm is the coordinate of the UAV at the next moment. In way of a loop iteration, a set of spatial coordinates is obtained by the recursive method, which is the track obtained by optimization. The specific logic can be described as follows:

First, input the space coordinates and environmental parameters of the UAV at this time to construct the three cost functions mentioned above, and use multi-step prediction and weighted summation to construct the total cost function, that is, the objective function to be optimized.

After that, the IHPSO algorithm is used to optimize the space trajectory points at the next moment. The optimized space of the particle swarm is the 3D space of the drone. The initial positions of the particles are randomly distributed around the drone coordinates.

After obtaining the optimal next time coordinate, calculate the relative relationship between the coordinate and the current time coordinate to determine whether the flight constraint is satisfied. If it is not satisfied, use the IHPSO algorithm to re-optimize until it is satisfied.

After satisfying the constraints, a new coordinate point is obtained. Recirculate the above process until the drone flies to the target point.

The above is the logical block diagram of this paper. Through the above process, the trajectory planning of the UAV can be realized in an iterative manner.

5. Simulation

To verify the feasibility and advantages of the proposed method, the above algorithm is simulated. First, set the motion constraints of the drone, and let the upper limit of the single change of $\Delta\phi$ and $\Delta\theta$ be 15° and the maximum speed be 10 m/s, plan once every 5 min, and predict five steps each time. The weighting factors ω_e, ω_r, and ω_s are 0.3, 0.4, and 0.3, respectively.

After setting the environmental parameters, let the drone depart from (0,0) km and fly to (100,100) km. The corresponding threats are terrain threats and radar threats.

The terrain threat is mainly comprised of natural terrain and mountain topography. The height of the natural terrain obeys the joint two-dimensional Gaussian distribution. To be true to reality, multiply the coefficient by 200 before joint distribution. For the other part of the mountain threat described earlier, the coordinates are [(10,10,500),(40,25,650),(45,50,750)], the coordinate units of the x and y axes are km, and the coordinate unit of the z axis is m.

The radar threat mainly comes from four radars, which are located at [(11,55,234),(35,45,397),(55,65,468.8),(90,60,450.6)], respectively, where the units are the same as above. To simplify the analysis, the performance parameters of the four radars are set to be the same limit detection performance for a target with an RCS of 0.1 m^2 and a detection distance of 100 km. In this way, although Q is unknown for (4), according to (8), when the position of the drone and radar is known, the radar can be calculated. Obviously, this is closer to the actual combat. Many parameters in Q are difficult to obtain or need to be accurately estimated, whereas, for the enemy radar, specific parameters cannot be obtained. The threat is determined by the ratio of the two.

To validate the performance of the algorithm proposed in this paper, it is compared with the HPSO algorithm. Regarding the algorithm parameters, the population size is set as 100, particles are divided into five groups, and the combined structure is three layers. The algorithm iterates 100 times, ω is randomly selected from the set of [0.95,1], and c=1.366 7. In the improved grouping strategy, let j=1, ω=0.8, and α=3. At the same time, this algorithm is compared with local PSO (LPSO) [33], hierarchical PSO (PS2O) [34], multi population cooperative PSO (MCPSO) [35], and competitive and cooperative PSO with information sharing mechanism (CCPSO-ISM) [36].

The simulation condition is I7-4960, the main frequency is 2.60 GHz with a memory of 16 G, and simula-

tion experiments are performed based on Matlab 2014a. The simulation results are shown in Fig. 14.

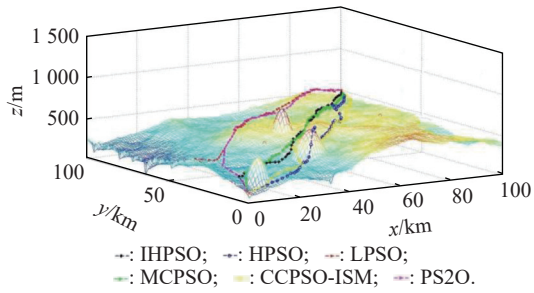


Fig. 14 Comparison of track planning (ω_c=0.3, ω_r=0.4, ω_s=0.3)

The terrain effect described above is shown in Fig. 14. The position of the radar is clearly shown by the red circles in Fig. 14. Considering safety, concealment, and timeliness, the above six methods are used to plan the UAV penetrating trajectory based on the predictive control model. The results are displayed in Fig. 14, in which all the tracks are effectively planned from the takeoff point to the endpoint.

Based on the simulation results, the trajectory can be divided into three categories. The first category is red and purple tracks. Such tracks are less threatened as a whole, but the overall route is longer than the counterparts. During track planning, when the red track is under the threat of radar R_1, a significant inflection point appears near it. This inflection point conforms to (8), that is, the threat of radar is inversely proportional to the fourth power of the distance. When the UAV is at the turning point, if the speed still has a component pointing to R_1, the distance will be further shortened, resulting in a significant increase in the threat of radar to the UAV. Therefore, in this situation, the drone will adjust its motion state, leading to an inflection point. The second category is blue and yellow tracks. Due to the difference in route selection when avoiding mountain peak M_1 at the initial stage, the drone must pass through mountain peak M_2 in order to reach the end as soon as possible. In the two routes, the drone is required to fly over and bypass the mountain peaks, putting it into greater natural threats. The third category is black and green routes, in which the drone basically flies diagonally to the destination, saving a considerable amount of time. Although the drone is threatened by the radars R_2 and R_3 due to the low flight altitude, the radar threat is weak in this case, according to (9). It also reflects the advantages of the predictive control model, which enables the drone to maintain a low altitude flight according to the terrain. It can also be seen from the figure that the black track is slightly stronger than the green track because the distance of the green track is more threatened by the radar R_2 than the black track, namely, the track planned in this paper.

To further quantify the performance differences of the six methods, this paper conducts 20 Monte Carlo experiments on the six methods to calculate the mean value of the objective function J. The results are shown in Fig. 15.

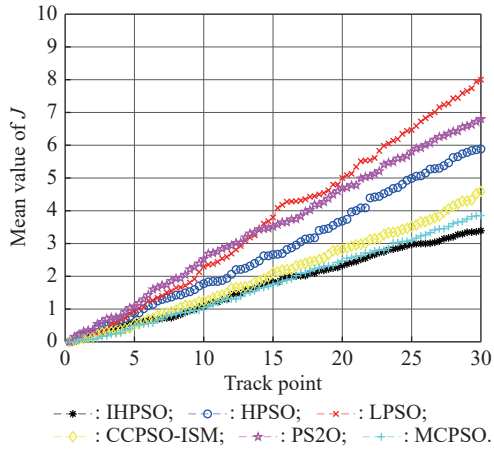


Fig. 15 Relationship between the mean value of the objective function J and the track points (ω_c=0.3, ω_r=0.4, ω_s=0.3)

As shown in Fig. 15, the objective function of this method is smaller than its counterparts. The black curve floats around 65, which is mainly threatened by radar R_3. However, the overall process objective function is better than other methods.

To further compare the performance of the algorithm, the complexity of the environment is increased. The weights of mountain threat, radar threat and fuel consumption are 0.3, 0.5, and 0.2, respectively. The simulation comparison chart and the fitness function curve are drawn as the following figures.

By comparing Fig. 16−Fig. 18, it can be seen that when the weight of the radar threat is increased to 0.5, the trajectory avoids the radar as much as possible, while the impact of fuel consumption decreases. The round-the-clock flight is used for UAVs to avoid mountain peaks.

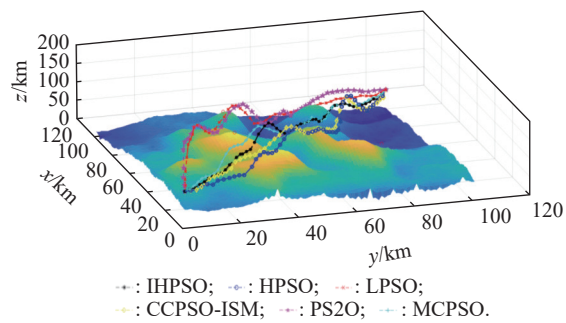


Fig. 16 Comparison of track planning (ω_c=0.3, ω_r=0.5, ω_s=0.2)

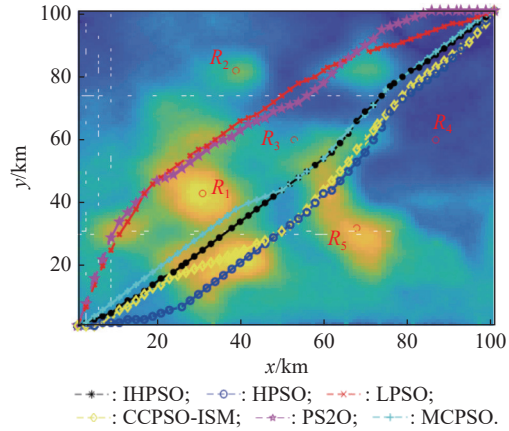


Fig. 17 Top view of track planning

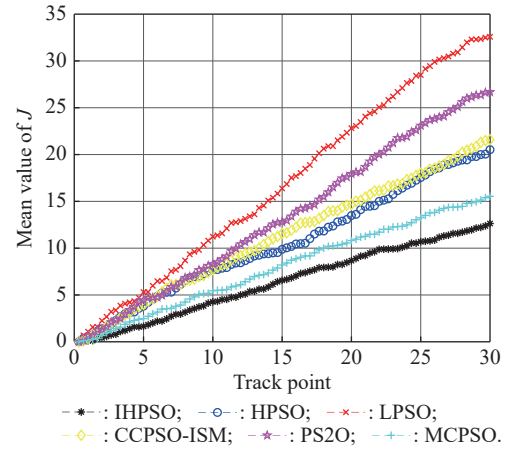


Fig. 18 Relationship between the mean value of the objective function J and the track points (ω_c=0.3, ω_r=0.4, ω_s=0.3)

It can be seen from the above comparison that the advantages of the algorithm in this paper are evident because the PSO algorithm mainly balances the contradiction between wide-area search and local accurate search. The wide-area search is to allow the particles to execute a more comprehensive search and prevent particles from falling into a local optimum. By contrast, the accurate search is to enable the particles to find the optimal value of the function more accurately.

For the HPSO [29] algorithm under different problem backgrounds, 80% wide-area search and 20% accurate search may not necessarily guarantee the optimal solution. It is prone to have excessive wide-area searches, while the local search is inadequate, making it difficult to reach the optimal solution. If the search rate is higher, the total number of searches must be increased, and the corresponding calculation time is significantly increased.

The performance of the LPSO [33] algorithm is related to the local best or optimal value in the neighborhood of the particle, which is equivalent to improving the search coverage of a single particle, However, apparently, there is no global best among the particles, and no particles are formed. The interaction of the algorithm makes the algorithm susceptible to falling into the local optimization.

PS2O [34] achieves information sharing through the symbiosis among particles and the extinction method, abandons the search for some areas where no optimal solution can be found, and improves search efficiency. However, the degree of particle information sharing of the PS2O algorithm is weaker, and thus its performance is more limited compared with the algorithm in this paper.

MCPSO [35] mainly focuses on the prototype of the hierarchical structure between particles. However, the affiliation between the corresponding particles remains unclear, and regrouping is not included. The initial information interaction between particles has become a mutual restraint between subsequent particles, restricting the effectiveness of the algorithm.

CCPSO-ISM [36] records the optimal information in the optimization process by constructing the blackboard method. This results in all particles being affected by the optimal particles, and all particles approach the optimal particles, thereby weakening the ability of the particles to perform the wide-area search.

The difference between the above six algorithms is mainly because the cost function is the result of integrating the influence of each threat. And this kind of synthesis has a variety of composition modes, thus forming different track points. Moreover, this paper uses three-step prediction for planning, and the optimal track points obtained by the three-step prediction are also different. In addition, the track points that can be predicted by trajectory planning are very limited, and it is impossible to adopt a global planning method. As a result, the initial trajectory has a huge impact on the subsequent trajectory points. The difference of the initial track points directly affects the subsequent track points. Therefore, different algorithms in Fig. 18 have different trends.

Therefore, the algorithm in this paper is superior in performance.

Compare the complexity of HPSO and IHPSO and the following results are found. For complex optimization functions, the number of grouping and regrouping of these two algorithms is far less than the number of optimization iterations, and the amount of calculation is negligible. The search strategies are the same and the complexity of particle state update is identical in both algorithms. In the HPSO algorithm, the number of iterations should be set. After each iteration, the maximum number of iterations should be determined, that is, one addition and one judgment should be performed. In the IHPSO algorithm, although it is a double judgment condition, if

the first judgment condition is not satisfied, it will return to the search. Additionally, the calculation is mainly for the first judgment. In the conversion from extensive search to exact search, the first criterion is (12), which requires two subtractions, one multiplication and one judgment. From accurate search to extensive search or algorithm termination, the discriminant is (13), which requires subtraction and judgment. Although the number of discriminating parts of IHPSO is slightly higher than that of HPSO, the amount of the extra calculation has little effect on a single iteration. In solving the above problems, IHPSO involves 78% iterations of the HPSO, saving at least 20% of the calculation time, which is significant for optimizing multiple complex functions.

6. Conclusions

This paper proposes a three-dimensional path planning algorithm based on IHPSO. When designing the objective function, terrain threat, radar threat and timeliness requirements of the UAV during the penetration process are comprehensively considered and quantified. At the same time, the HPSO algorithm is improved, and the objective function is solved based on the predictive control model to achieve three-dimensional path planning.

In researching radar threats, by taking the network radar as the countermeasure and starting from the perspective of the echo strength received by the network radar, the analysis method of the RCS is constructed to quantify the threat degree of the network radar, making it more accurate to measure radar threats.

The three shortcomings of the strong randomness of the grouping strategy, the weak generalization ability of the transition conditions, and the unclear setting of the number of iterations in the HPSO algorithm are well addressed in this paper. Grouping strategies based on systematic clustering and information entropy are proposed correspondingly. State transition conditions are constructed based on the fitness function of particles. Algorithm termination conditions are constructed using the distances between particles to improve the performance of the algorithm.

UAVs are planned with predictive control models and IHPSO algorithms and compared with mainstream algorithms. It can be seen from the simulation results that the model in this paper can meet the requirements of security, concealment, and real-time performance of UAV penetration. Overall, the algorithm in this paper is better than other versions of the PSO in the optimization of the objective function.

References

[1] WANG L, LIU Z, CHEN C L P, et al. A UKF-based predictable SVR learning controller for biped walking. IEEE Trans. on Systems, Man, and Cybernetics: Systems, 2013, 43(6): 1440–1450.

[2] WANG H C, WANG J L, DING G R, et al. Completion time minimization with path planning for fixed-wing UAV communications. IEEE Trans. on Wireless Communications, 2019, 18(7): 3485–3499.

[3] ZHANG H D, HE Y Q, LI D C, et al. Marine UAV-USV marsupial platform: system and recovery technic verification. Applied Sciences, 2020, 10(5): 1583.

[4] MARDANI A, CHIABERGE M, GIACCONE P. Communication-aware UAV path planning. IEEE Access, 2019, 7: 52609–52621.

[5] CHUANG H M, HE D, NAMIKI A. Autonomous target tracking of UAV using high-speed visual feedback. Applied Sciences, 2019, 9(21): 4552.

[6] GIUSTI A, GUZZI J, CIRESAN D C, et al. A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robotics and Automation Letters, 2015, 1(2): 661–667.

[7] WANG J G, WANG Y F, ZHANG J M, et al. Resolution calculation and analysis in bistatic SAR with geostationary illuminator. IEEE Geoscience and Remote Sensing Letters, 2012, 10(1): 194–198.

[8] YANG P, TANG K, LOZANO J A, et al. Path planning for single unmanned aerial vehicle by separately evolving waypoints. IEEE Trans. on Robotics, 2015, 31(5): 1130–1146.

[9] VAN NGUYEN H, REZATOFIGHI H, VO B N, et al. Online UAV path planning for joint detection and tracking of multiple radio-tagged objects. IEEE Trans. on Signal Processing, 2019, 67(20): 5365–5379.

[10] KUMAR P, GARG S, SINGH A, et al. MVO-based 2-D path planning scheme for providing quality of service in UAV environment. IEEE Internet of Things Journal, 2018, 5(3): 1698–1707.

[11] LIU Y S, WANG Q X, HU H S, et al. A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. IEEE Trans. on Systems, Man, and Cybernetics: Systems, 2018, 49(11): 2362–2372.

[12] GALCERAN E, CARRERAS M. A survey on coverage path planning for robotics. Robotics and Autonomous Systems, 2013, 61(12): 1258–1276.

[13] LIU C, ZHANG S H, AKBAR A. Ground feature oriented path planning for unmanned aerial vehicle mapping. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(4): 1175–1187.

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

[15] WAI R J, PRASETIA A S. Adaptive neural network control and optimal path planning of UAV surveillance system with energy consumption prediction. IEEE Access, 2019, 7: 126137–126153.

[16] LI Y, CHEN H, ER M J, et al. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics, 2011, 21(5): 876–885.

[17] SUN Z C, WU J J, YANG J Y, et al. Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution. IEEE Trans. on Geoscience and Remote Sensing, 2016, 54(11): 6444–6457.

[18] LEVIN J M, NAHON M, PARANJAPE A A. Real-time motion planning with a fixed-wing UAV using an agile

maneuver space. Autonomous Robots, 2019, 43: 2111–2130.
[19] ZHU M N, ZHANG X H, LUO H, et al. Optimization dubins path of multiple UAVs for post-earthquake rapid-assessment. Applied Sciences, 2020, 10(4): 1388.
[20] SUN L L, CAO Y H, WU W H, et al. A multi-target tracking algorithm based on Gaussian mixture model. Journal of Systems Engineering and Electronics, 2020, 31(3): 482–487.
[21] QIE H, SHI D X, SHEN T L, et al. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. IEEE Access, 2019, 7: 146264–146272.
[22] YUE W, GUAN X H, WANG L Y. A novel searching method using reinforcement learning scheme for multi-UAVs in unknown environments. Applied Sciences 2019, 9(22): 4964.
[23] ZHANG W. Coarse-to-fine UAV target tracking with deep reinforcement learning. IEEE Trans. on Automation Science and Engineering 2018, 16(4): 1522−1530.
[24] DUAN H B, LI P, SHI Y H, et al. Interactive learning environment for bio-inspired optimization algorithms for UAV path planning. IEEE Trans. on Education, 2015, 58(4): 276–281.
[25] YANG Q, YOO S J. Optimal UAV path planning: sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms. IEEE Access, 2018, 6: 13671–13684.
[26] SHAO Z. Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization. Applied Sciences, 2019, 9(13): 2621.
[27] ROBERGE V, TARBOUCHI M, LABONTE G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. on Industrial Informatics, 2012, 9(1): 132–141.
[28] WANG Y B, BAI P, LIANG X L, et al. Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms. IEEE Access, 2019, 7: 105086–105099.
[29] ROSHANZAMIE M, BALAFAR M A, RAZAVI S N. Empowering particle swarm optimization algorithm using multi agents' capability: a holonic approach. Knowledge-Based Systems, 2017, 136(11): 58–74.
[30] ZENGIN U, DOGAN A. Real-time target tracking for autonomous UAVs in adversarial environments: a gradient search algorithm. IEEE Trans. on Robotics, 2007, 23(2): 294–307.
[31] GILLES L. On determining the flyability of airplane rectilinear trajectories at constant velocity. Advances in Aircraft and Spacecraft Science, 2018, 5(5): 551–579.
[32] LIU L, QU G Q, KONG W. UAV 3D trajectory planning by using dynamic programming and potential theory. Computer Engineering and Applications, 2013, 49(20): 235–239.
[33] SUGANTHAN P N. Particle swarm optimiser with neighbourhood operator. Proc. of the Congress on Evolutionary Computation, 1999, 3: 1958–1962.
[34] CHEN H N, ZHU Y L. Optimization based on symbiotic multi-species coevolution. Applied Mathematics and Computation, 2008, 205(1): 47–60.
[35] NIU B, ZHU Y L, HE X X. Multi-population cooperative particle swarm optimization. Proc. of the European Conference on Artificial Life, 2005: 874–883.
[36] LI Y, ZHAN Z H, LIN S, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. Information Sciences, 2015, 293(2): 370–382.

Biographies

LUO Jing was born in 1984. She received her B.E. degree of electronic information, and M.E. degree of information and communication engineering from Wuhan University of Technology, Wuhan, China. She has been pursuing her Ph.D. degree at Naval Engineering University since 2017. Her current research interests are swarm intelligence, intelligence system and signal processing.
E-mail: 94523685@qq.com

LIANG Qianchao was born in 1961. He received his M.E. degree of power engineering from Naval Engineering University, Wuhan, China. And completed his Ph.D. degree in power engineering and engineering thermal physics from Huazhong University of Science and Technology in 2004, Wuhan, China. His current research interests are power engineering and simulation and optimization of power mechanical system.
E-mail: lqc163cc@163.com

LI Hao was born in 1981. He received his Ph.D. degree in electronic science and technology from Air Force Engineering University in 2017, Xi'an, China. His current research interests are swarm intelligence, UAV swarm, intelligence systems and signal processing.
E-mail: afeu_li@163.com