IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2023, accepted 28 February 2023, date of publication 3 March 2023, date of current version 10 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252018

RESEARCH ARTICLE

UAV Trajectory Planning for Complex Open Storage Environments Based on an Improved RRT Algorithm

JINGCHENG ZHANG[1,4], YUQIANG AN[2,4], JIANING CAO[3], SHIBO OUYANG[4], AND LEI WANG[4]

[1]Faculty of Science, Kunming University of Science and Technology, Kunming, Yunnan 650500, China
[2]Faculty of Management and Economics, Kunming University of Science and Technology, Kunming, Yunnan 650500, China
[3]Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology, Kunming, Yunnan 650500, China
[4]Logistics Center, Hongyun Honghe Tobacco (Group) Company Ltd., Kunming, Yunnan 650231, China

Corresponding author: Yuqiang An (59672319@qq.com)

This work was supported in part by Hongyun Honghe Tobacco (Group) Company Ltd. Technology Project under Grant HYHH2021XX04 and Grant HYHH2020XX03.

ABSTRACT Multi-rotor UAVs (Unmanned Aerial Vehicles) have been increasingly used for hazardous inspection tasks in complex open-air warehouse storage environments due to their high maneuverability and aerial perspective. To facilitate rapid response to patrol missions and improve the efficiency of UAV trajectory planning. This paper established a rotary-wing UAV trajectory plan model considering UAV patrol efficiency, trajectory cost, and power consumption cost. Secondly, an improved SSA (salp swarm algorithm) is incorporated for the shortcomings of low algorithmic search efficiency and unsmooth paths when planning paths in the traditional RRT (Rapidly-exploring Random Trees). The predation mechanism of the salps group is incorporated into the random sampling of the RRT algorithm, which reduces the invalid sampling of random points and introduces the adaptive leader structure, and reverses the search strategy to improve the algorithm's global search for superiority at the later stage of the search. Finally, the designed LASSA-RRT algorithm is subjected to simulation experiments and compared with RRT, RRT∗, IRRT, and PF-RRT∗ in a cross-sectional manner. The results show that the LASSA-RRT algorithm has an average reduction of 55.83% in sampling times, 51.91% in run time, 13.17% in track length, and 0.1491% in flight cost. In summary, this paper's UAV trajectory planning method can be effectively applied to complex open storage environments. It can provide a helpful reference direction for UAV trajectory planning.

INDEX TERMS RRT, salp swarm algorithm, path planning, UAV, adaptive, reverse search, warehouse inspection.

I. INTRODUCTION

With the continuous development of the Industry 4.0 process, intelligent technologies have been widely used in industry [1]. Warehousing, as an important part of the industrial generation, is also gradually undergoing digital transformation, and drones are commonly used in the warehousing industry due to their high mobility as well as their aerial perspective [2], [3]. Open-air storage is a special type of storage widely used in the storage of tobacco, wood, etc. The fire risk of open storage has always been a key concern for decision-makers in the industry. Frequent manual safety inspections have caused increased costs, while the development of drone technology, sensor technology, and artificial intelligence can provide intelligent solutions for open storage inspections.

Unlike the adjustable environment of indoor storage, the traditional safety inspection of open-air storage requires staff to hold a thermometer to measure the temperature of each stack and has measured the risk of fire. Thanks to the rapid development of 5G technology in recent years, the collection, transmission and processing of data in IoT networks has become increasingly fast and efficient, supporting the use

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash.

of drones as real-time monitoring platforms [4], [5]. The communication transmission efficiency required by UAVs for track planning in different environments has reached conditions [6], [7]. In recent years an increasing number of unmanned maneuvering devices have been used in inspection missions [8].

The excellent performance of UAVs in applications such as grid inspection [9] and industrial pipeline inspection [10] has proven their significant advantages in the automation industry. The key to industrial inspection applications based on UAVs lies in the rationality of trajectory optimisation. Tran et al. designed a UAV trajectory model with total energy consumption minimisation as the optimisation objective based on satisfying the requested timeout (RT) requirement and energy budget [11]. Wu et al., on the other hand, considered visual occlusion and environmental constraints and constructed an optimized model for UAV trajectories in urban environments to provide solutions for UAV applications in complex environments [12]. As for warehouse inspection trajectory optimisation, current research on UAVs in warehousing is mainly based on indoor warehouse inspections, e.g. Liu et al. constructed a UAV warehouse inventory trajectory planning model based on a UAV dynamics model and with energy efficiency and time efficiency minimisation as the optimisation objectives, providing good trajectory planning results for task-driven UAV warehouse inspections [13]. In addition, there is little research on open storage and the results of this paper could fill the gap for the industry.

With the continuous development of AI, more and more intelligent algorithms have been used for the solution of UAV patrol trajectories, and both heuristic algorithms and meta-heuristic algorithms have achieved certain applications. Among them, the multiverse algorithm [14], RRT [15] and A* algorithm [16] have good planning results. Among them, the RRT algorithm has received much attention due to its excellent search performance in cluttered environments. However, due to its large randomness of sampling, the search has more path redundancy points and a slow search speed. To compensate for the shortcomings of the RRT algorithm, Karaman et al. proposed the RRT* algorithm, which adds a process of reselecting a new parent node and wiring it after finding a new node. However, finding a suitable path in a particular environment is challenging [17]. In order to improve the quality of the random tree nodes, the researchers introduced various improvement strategies for optimization in the classical algorithm. Among them, Li et al. introduced the ant colony algorithm and particle swarm algorithm based on the RRT algorithm. They pruned the UAV flight path for optimization, which significantly reduced the redundant turns in the UAV flight and effectively reduced the trajectory length [18]. However, it still generated the path with the RRT algorithm, and the algorithm generated the initial path slowly. Yin et al. proposed an improved RRT algorithm (IRRT), which makes the planned trajectory better by improving the selection strategy of the root node nearest to the random sampling point and introducing the trajectory distance constraint [19]. However, there are problems of high path cost and slow convergence speed. To solve the problems of RRT* algorithm in special environment such as more memory occupation and low planning efficiency of path planning, Fan et al. introduced the goal biasing strategy in the sampling of RRT* algorithm and used dichotomy to create a new parent node, proposed PF-RRT* algorithm to make the sampling more goal-oriented and accelerate the search speed [20]. Huang et al. proposed a Bi-directional-Rapidly Exploring Random Tree (Bi-RRT*) algorithm, which avoids invalid expansion, saves storage space, and improves the convergence speed of the algorithm [21], but is vulnerable in extremely complex environments. In summary, the RRT algorithm has been widely used as a classical path planning algorithm, but it still needs to be adjusted and optimized for the differences and complexity of the scene environment to improve its algorithmic efficiency and effectiveness.

In terms of algorithmic search efficiency, Swarm Intelligent (SI) as a population-based metaheuristic algorithm has received a lot of attention from researchers due to its good generality, excellent search efficiency and optimization accuracy [22]. Among them, Pan et al. introduced two improvement strategies, personal example learning and mirror reflection learning, based on the Golden Eagle optimization algorithm GEO, and verified the superiority of the designed algorithm by comparison of arithmetic cases [23]. Chen et al. then designed an adaptive learning mechanism and four operational operators for improvement based on the standard coyote algorithm, and verified its strong global search capability by offline trajectory planning simulation [24]. As an emerging swarm intelligence algorithm, the salps swarm algorithm has also been effectively applied in UAV trajectory planning [25]. The use of this for UAV trajectory planning is therefore an effective experiment.

In summary, this paper investigates the problem of optimising the UAV inspection trajectory in a complex open storage environment and designs the LASSA-RRT algorithm to solve the problem. To summarise the main contributions of this paper are as follows.

1) An inspection energy and time ratio optimization model of UAV trajectory for complex storage environment was established to enable the UAV to fulfill the requirements of inspection tasks.
2) The adaptive structure is proposed on the basis of the standard bottle sheath swarm algorithm in terms of the number of followers, balancing the exploration and exploitation capabilities of the algorithm; the introduction of a reverse search strategy enhances the algorithm's late-stage optimization-seeking capability and prevents the algorithm from falling into a local optimum. The RRT algorithm is also combined with the solution of UAV trajectory optimization in order to explore a new 3D path planning scheme.
3) The results show that the LASSA-RRT algorithm has good performance. The algorithm successfully solves the path planning problem of open-air warehouse

inspection by quadrotor UAV, with the advantages of a smooth path, fewer sampling points, and low flight power consumption, which verifies the effectiveness of the algorithm.

The remainder of the paper is organized as follows: Section II introduces the investigated UAV inspection problem in a complex open-air storage environment and describes the optimization model developed; in Section III, the designed LASSA-RRT algorithm is presented; in Section IV, the designed algorithm is verified numerically and the designed cruise trajectory method is simulated and compared with cutting-edge algorithms in a cross-sectional manner; finally, the research is summarized and an outlook is presented.

II. PROBLEM DESCRIPTION AND MODELING

The UAV dangerous situation inspection task in the complex open storage environment is carried out based on the rotary-wing UAV. By pre-planning the inspection points, the UAV is equipped with sensors (visible light sensor, infrared sensor, etc.) to realize the inspection of dangerous situations. First, a reference point O is selected in the area with inspection, and a local coordinate system and a 3D model are established to carry out inspection task planning. Among them, $I_i = (x_i, y_i, z_i)$ represents the coordinates of any inspection point and $\forall i \in \{1, 2, \cdots, n\}$. $U = (x, y, z)$ denotes the coordinates of the UAV, then the inspection process of the UAV can be represented by a series of consecutive points, and the trajectory planning of the UAV is the sequential planning of the inspection points.

Model assumptions:

Due to its high mobility, the rotary-wing UAV for inspection can have better targeting during inspection operations, and its complex dynamics will greatly increase the difficulty of the inspection model as well as the computational volume. In order to improve the response speed of the UAV in inspection tasks, the following reasonable assumptions are made.

1) treats the flight of the UAV as uniform, specifying that its uniform flight speed is 80% of its maximum flight speed;
2) negligible energy consumed by the UAV to adjust its attitude [26];
3) Wind speed magnitudes within the workable range of the UAV;

A. MODELING

1) INSPECTION TIME

The trajectory of the UAV between any nodes can be represented as a vector and can be projected to the X-axis, Y-axis and Z-axis directions. The flight distance of a drone between any nodes can be characterized by the Euclidean parametrization of the corresponding vector of that track segment as follows:

$$\overrightarrow{I_{ij}} = \overrightarrow{I_{ij}^{X}} + \overrightarrow{I_{ij}^{Y}} + \overrightarrow{I_{ij}^{Z}}, \tag{1}$$

$$\left\| \overrightarrow{I_{ij}} \right\|_2 = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}, \tag{2}$$

$$\begin{bmatrix} d_{ij}^{x} \\ d_{ij}^{y} \\ d_{ij}^{z} \end{bmatrix} = \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} \left\| \overrightarrow{I_{ij}} \right\|_2, \tag{3}$$

$$\begin{bmatrix} v_{ij}^{H} \\ v_{ij}^{V} \end{bmatrix} = \begin{bmatrix} \sin\gamma \\ \cos\gamma \end{bmatrix} \left\| \overrightarrow{V_{ij}} \right\|_2, \tag{4}$$

where $\overrightarrow{I_{ij}}$ in Eq. (1) is the vector of trajectories between any nodes. $\overrightarrow{I_{ij}^{X}}$, $\overrightarrow{I_{ij}^{Y}}$ and $\overrightarrow{I_{ij}^{Z}}$ are the projection vectors of the track vector track $\overrightarrow{J_{ij}}$ on the X-axis, Y-axis and Z-axis, respectively. Equation (2) indicates the length of the trajectory between any nodes. $\|\cdot\|_2$ denotes the 2-parametric number of vectors i.e. Euclidean parametrization. Equation (3) represents the flight distance of the UAV projected on each axis. α, β and γ denote the angle of the trajectory $\overrightarrow{I_{ij}}$ with the X-axis, Y-axis and Z-axis, respectively. Similarly, the velocity magnitude of the UAV in the horizontal and vertical directions can be obtained, as shown in Equation (4).

The total time spent by the UAV during the inspection process includes both the UAV flight time and the operational time for inspection using sensors at the inspection point, as indicated below.

$$X_{ij} = \begin{cases} 1, & \text{UAV from point i to j} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$Y_i = \begin{cases} 1, & \text{UAV detection at i-point} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$IT = \sum_{i,j} \left| \frac{d_{ij}^{x}}{v_{ij}^{H}} + \frac{d_{ij}^{y}}{v_{ij}^{H}} + \frac{d_{ij}^{z}}{v_{ij}^{V}} \right| \times X_{ij} + \sum_{i} \left(T_i^{\mathrm{A}} + T_i^{W} + T_\delta \right) \times Y_i \tag{7}$$

Equation (5) is a 0-1 variable when $X_{ij} = 1$ means that the UAV travels from point i to point j, and conversely, when $X_{ij} = 0$ means that the UAV does not travel from point i to point j. Equation (6) is a 0-1 variable, when $Y_i = 1$ means that the UAV is performing detection operations at detection point i, and vice versa when $Y_i = 0$ means that the UAV is not performing detection operations at detection point i. Equation (7) indicates the total time spent by the UAV for inspection operations, where T_i^{A} indicates the position adjustment time of the UAV at the inspection point and T_i^{W} indicates the fixed operation time of the UAV for inspection operations using sensors. T_δ denotes a normal random variable with mean 0 and standard deviation σ. It is used to correct for the effect of multiple errors in the actual operation on the estimated value of the detection time.

2) ENERGY CONSUMPTION FOR INSPECTION

Based on the study of Fan et al. [20], the approximate relationship function between the lift force F and power P_{UF} of the UAV is obtained as shown in Equation (8), Combined with the force analysis of the UAV in each flight state, the power

consumed by the UAV is obtained as shown in Equation (15).

$$F = K \times \sqrt{P} - C \tag{8}$$

$$F_{ij}^{H} - G - F_{f}^{H} = 0 \tag{9}$$

$$F_{ij}^{N} \times \cos\gamma - G - F_{f}^{N} = 0 \tag{10}$$

$$f_{ij}^{H} = \frac{1}{2} C\rho S(v_A)^2 \tag{11}$$

$$f_{ij}^{V} = \frac{1}{2} C\rho S(v_{ij}^{V} + v_A)^2 \tag{12}$$

$$\begin{bmatrix} F_{ij}^{H} \\ F_{ij}^{N} \end{bmatrix} = \begin{bmatrix} G + \frac{1}{2} C\rho S(v_A)^2 \\ \frac{G}{\cos\gamma} + \frac{1}{2} C\rho S(v_{ij} + v_A)^2 \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} P_{ij}^{H} \\ P_{ij}^{N} \end{bmatrix} = \frac{1}{K^2} \begin{bmatrix} \left[G + \frac{1}{2} C\rho S(v_A)^2 + C\right]^2 \\ \left[\frac{G}{\cos\gamma} + \frac{1}{2} C\rho S(v_{ij}^{V} + v_A)^2 + C\right]^2 \end{bmatrix} \tag{14}$$

$$IE = \sum_{i,j} \left[P_{ij}^{N} \times \left(\frac{d_{ij}^{x}}{v_{ij}^{H}} + \frac{d_{ij}^{y}}{v_{ij}^{H}} + \frac{d_{ij}^{z}}{v_{ij}^{V}} \right) \times X_{ij} \right] + \sum_{i} \left\{ \left[P_{ij}^{H} \times T_i^{\mathrm{A}} + P_S \times \left(T_i^{W} + T_\delta \right) \right] \times Y_i \right\} \tag{15}$$

Equation (8) is the approximate energy consumption equation of the UAV, where K is the power conversion factor and C is a constant. Equation (9) is the equilibrium equation in the hovering state, Equation (10) is the balance equation in general flight state, where F_{ij}^{H} and F_{ij}^{N} are the lift forces of the UAV in the hovering and general flight, G is the self-weight of the UAV, and F_f^H and F_f^N are the air resistance of the UAV in hovering and general flight, respectively. Equations (11) and (12) are the formulae for the calculation of F_f^H and F_f^N. v_A is the relative wind speed, C is the drag coefficient, ρ is the air density, and S is the force area. Equation (13) is the lift force generated by the UAV. Equation (14) is the working power of the UAV. Equation (15) is the total energy consumed by the UAV in the inspection process, P_S for the sensor-integrated work energy consumption.

3) OBJECTIVE FUNCTION

The objective function is shown in Equation (16), which is determined by the ratio of the energy consumed during the UAV inspection and the time used for the inspection. The larger the objective function is, i.e., the larger the ratio of energy and time, the higher the inspection efficiency of the UAV is indicated.

$$O = \max\left(\frac{IE}{IT}\right) \tag{16}$$

4) CONSTRAINTS

Combining the requirements of the UAV's hazardous inspection task in the open storage environment with the physical constraints of the UAV itself and the working constraints of the sensors it carries, the following constraints were obtained:

$$\sum_{i=1}^{n} X_{0i} = 1 \tag{17}$$

$$\sum_{i=1}^{n} X_{i0} = 1 \tag{18}$$

$$\sum_{i=1} X_{ij} = 1, \forall j \in \{1, 2, \cdots, n\} \tag{19}$$

$$\sum_{j=1} X_{ij} = 1, \forall i \in \{1, 2, \cdots, n\} \tag{20}$$

$$-5^\circ \le \gamma \le 5^\circ \tag{21}$$

$$IE_{\max} - IE \ge P_{e0}^{N} \times \left(\frac{d_{e0}^{x}}{v_{e0}^{H}} + \frac{d_{e0}^{y}}{v_{e0}^{H}} + \frac{d_{e0}^{z}}{v_{e0}^{V}} \right) \tag{22}$$

$$\sum Y_i = n, \forall i \in \{1, 2, \cdots, n\} \tag{23}$$

$$H_{\min}^{u} \le H_i^{u} \le H_{\max}^{u}, \forall i \in \{1, 2, \cdots, n\} \tag{24}$$

$$\pi \left(\left\| \overrightarrow{UI_i} \right\|_2 \times \tan\delta \right)^2 \ge S_i, \forall i \in \{1, 2, \cdots, n\} \tag{25}$$

$$\delta_{\min} \le \delta \le \delta_{\max} \tag{26}$$

Equations (19) and (20) ensure that each point to be inspected is inspected by a UAV. Equation (21) is the pitch angle range for the UAV to maintain stability during flight [21]. Equation (22) is the UAV return energy constraint, Where $IE_{\max}$ is the total energy of the UAV, IE is the energy required for the UAV to complete the inspection task, e is the last inspection point of the UAV, and the right side of the equation indicates the energy required for the UAV to return to the inspection center from the last inspection point. Equation (23) ensures that all devices are inspected. Equation (24) is the safe flight altitude requirement for the UAV. Equation (25) for the detection of UAV operating range requirements, $\left\| \overrightarrow{UI_i} \right\|_2$ is the vertical distance from the UAV to the detection point, δ is the rotation angle of the sensor gimbal carried by the UAV, and S_i is the range to be detected at detection point i. Equation (26) is the gimbal rotation range of the UAV.

III. DESIGN OF RRT ALGORITHM WITH HYBRID SSA ALGORITHM

A. RRT ALGORITHM

1) BASIC RRT ALGORITHM

The fast extended random tree algorithm was first proposed by S.M. Lavall in 1998, and the algorithm has achieved wide application in query planning of paths. Figure 1 shows a schematic of the expansion of the RRT algorithm in a given space, starting from the starting point *Q_init* and finding a random point *q_rand* in the space to find the nearest point *q_near* in the tree, generating a new point according to the step size, and perform collision detection. If *q_rand* and *q_near* are within a certain range, then *q_rand* is taken as the new node *q_near*. When the random tree contains the target point, the exploration stops, starting with the closest leaf node and searching the parent node in reverse to get a path from the root node to the target point.

Since the RRT algorithm traverses the entire space by random sampling, the randomness of the nodes algorithm to find the UAV path is not optimal, and the existence of

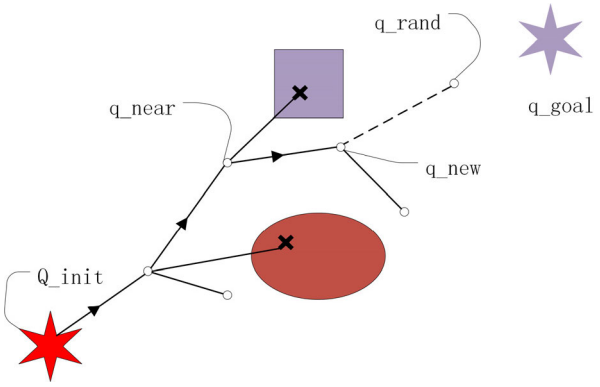


FIGURE 1. Schematic of RRT.

redundant nodes will lead to too tortuous paths, which affects the flight performance of the UAV.

2) IMPROVED RRT ALGORITHM

In this paper, we add a target point-oriented probability function to the RRT algorithm to accept the generated random points with a certain probability to speed up the convergence of the algorithm. The pruning and smoothing algorithm of the flight path is also introduced to the RRT algorithm to build and smooth the path after completing the obstacle avoidance path planning.

Specifically adding a target-guided probability r at each iteration to generate a random point q_rand, When the probability $rand < r$, $q\_rand = (x\_rand, y\_rand, \theta\_rand)$; When $rand < r$, $q\_rand = q\_goal$, where $rand \in [0, 1]$, Based on the RRT algorithm, the planning time is shortened by adding probability settings with guiding effect to make the exploration direction move quickly toward the target point.

3) IMPROVED RRT ALGORITHM

The RRT algorithm uses a random search method when searching for paths, which leads to the disadvantage of

having generated paths with poor smoothness and many redundant points. When in a complex obstacle environment, the UAV will produce too much unnecessary steering and the flight path will be too tortuous, which will reduce the flight performance of the UAV, aggravate the mechanical loss and power consumption of the UAV parts, and reduce the service life of the UAV, so the pruning strategy is used to prune the UAV flight path after the RRT algorithm completes the obstacle avoidance planning.

Take the RRT algorithm to generate any three consecutive points in the path, As shown in Figure 2, X_1, X_2, X_3 are three consecutive points in the path. Remove the redundant points and determine the angle of the line between X_1, X_2 and X_3 points in the following way:

1) $120^0 < \theta \leq 180^0$:The path at node X_2 is straighter and does not need to be processed for node X_2.

2) $0^0 \leq \theta \leq 120^0$:The path curves at node X_2. At this point, the position of the point is trimmed.

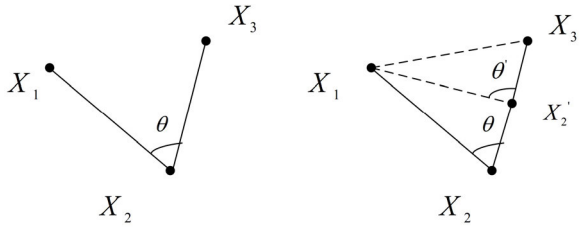


FIGURE 2. Pruning operation diagram.

First, determine whether the line between node X_1 and node X_3 passes through the obstacle; if not, then directly discard node X_2, the line $\overline{X_1X_3}$ as the new path; If the line passes through an obstacle, take the midpoint X_2' of $\overline{X_2X_3}$, and determine whether the path $\overline{X_1X_2'}$ of the line between X_1 and X_2' holds, and if it does not pass through the obstacle, then ac is the new path; If it passes through an obstacle, the original node X_2 is retained.

$$B_{t,N}(t) = C_N^i t^k (1-t)^{N-i}. \tag{27}$$

For a path with N+1 control points, a polynomial curve with N times can be determined. The parametric equation for the individual points of the curve can be obtained as

$$P(t) = \sum_{i=1}^{N} P_i B_{i,N}(t), \tag{28}$$

where $B_{i,N}$ is the Nth B spline basis function, also called the summation function. P_i is the coordinate of the control node. Taking the cubic curve as an example, the parametric equation of the cubic Bezier curve can be obtained as

$$P(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3. \tag{29}$$

The B spline curve is obtained by joining each Bezier curve.

In summary, the basic working steps of the algorithm are as follows:

Step 1: Algorithm initialization.

Step 2: determine whether the target position is reached; if it is reached, then the random tree construction is completed to step 6; if the target point is not reached to step 3.

Step3: randomly sample the search space with a certain probability of getting the point q_rand, and find the nearest point $q\_nearest$ in the tree

Step 4: Generate new points according to the step length q_new

Step5: Determine if the line between $q\_nearest$ and q_new will collide with the obstacle; if it does, then discard the point; if it is feasible, then add it to the tree.

Step6: Find the path from the start point Q_init to the end point q_goal in the constructed tree.

Step7: Prune the planned path.

B. SALP SWARM ALGORITHM

The salp swarm algorithm is an intelligent algorithm proposed by MirjaLiLi et al. based on the aggregation foraging

behavior of the bottlenose sheath. The salps are a group of small pelagic glial chordates that feed on mayfly plants in the water and move through the water by inhaling the exhaled seawater.

1) POPULATION INITIALIZATION

The search space for bottlenose sea squirt individuals is defined as a $D \times N$-dimensional space, where N is the population size and D represents the spatial dimension, The location of each salp in the population is defined as $X_n = [X_n^1, X_n^2, \ldots, X_n^i], n = 1, 2, \ldots, N, i = 1, 2, \ldots, D$.Equation (30) indicates that the position vector X in the salps algorithm consists of N salps with dimension d:

$$\text{X=rand(N,D)} \times (\text{ub}_n^i\text{-lb}_n^i)+\text{lb}_n^i, n \in [1, N], i \in [1, D] \quad (30)$$

where ub_n^i is the upper bound of the search for the nth population in the i-th dimension. lb_n^i is the search lower bound of the nth population in the i-th dimension. In the population, the leader is defined in each dimensional value as X_n^1, and the follower is defined in each dimensional value as $X_n^i, i = 2, 3, \ldots, N$.

2) LEADER POSITION UPDATE

In the SSA algorithm, the position of the leader needs to be determined first, and the position update of the leader needs to determine the food position, that is, the target position F of the salps, and the position of the leader is updated according to the following rules.

$$X_n^1 = \begin{cases} F_i + c_1((ub - lb)c_2 + lb), & c_3 \geq 0.5 \\ F_i - c_1((ub - lb)c_2 + lb), & c_3 < 0.5, \end{cases} \quad (31)$$

where F_i is the value of the target position in the i-th dimension; c_1, c_2, c_3 is the position control parameter, c_2 and c_3 are random numbers between [0,1] that determine the direction and length of the leader's movement to the next position in the j-th dimension. The leader's location update is mainly influenced by the location of the food source, The parameter c_1 is called the convergence factor, which is used to balance the convergence speed of the algorithm in the iterative process, its gradual increase with the number of samples strengthens the algorithm's ability to find the best in complex environments and reduces the number of samples the algorithm has to take. where k is the current number of samples, $k_{\max}$ is the maximum number of samples

$$c_1 = 1/2e^{-(4\times k/k_{\max})^2} \quad (32)$$

For the trajectory planning problem, the location of food source F cannot be determined. In this paper, the leader is updated by adding SSA random perturbation to the RRT algorithm global random point taking and introducing a target biasing strategy. The leader's position $T_{1,i}^l$ in the direction of i in the lth generation is updated by the formula

$$p_i = 0.8(T_{1,i}^l - lb_n^i)/(ub_n^i - lb_n^i)$$

$$T_{1,i}^{l+1} = \begin{cases} T_{1,i}^l - c_1c_2(T_{1,i}^l - lb) & r \leq p_i \\ T_{1,i}^l + c_1c_2(ub - T_{1,i}^l) & p_i \leq r < 0.8 \\ q\_goal & 0.8 \leq r \end{cases} \quad (33)$$

Among them, r = rand(0, 1), $p_i = 0.8(T_{1,i}^l - lb_n^i)/(ub_n^i - lb_n^i)$, denote the probability bounds for the upward and downward searches in three directions.

3) FOLLOWER'S LOCATION UPDATE

In the SSA algorithm, the position of the follower is only related to the position of a salps in front of it. According to Newton's law of motion, the simplified formula is

$$X_i^n = \frac{1}{2}(X_i^{n-1} + X_i^n), \quad (34)$$

where: X_i^n is the position of the n-th($n \geq 2$) follower on the i-th dimension.

Due to the complexity of the position update in the track planning problem, the position update of the follower cannot be derived from a simple kinematic relationship.:

$$T_{1,i}^{l+1,n} = \begin{cases} T_{1,i}^{l,n-1} & r \leq p_i \\ T_{1,i}^{l,n} & p_i \leq r < 0.8 \\ q\_goal & 0.8 \leq r. \end{cases} \quad (35)$$

C. IMPROVED SALP SWARM ALGORITHM

1) ADAPTIVE LEADER STRATEGY

In the basic SSA algorithm, the leader is the first in the population, and this mechanism is prone to fall into local optimality when solving solutions with a large number of local optima. In this regard [22] proposed an adaptive leader structure of the bottle sea-sheath algorithm to enhance the development of the algorithm at a later stage. The adaptive leader strategy allows the algorithm to adaptively increase the number of leaders from 1 to N/2 in an iteration. In the pre-iterative stage, more followers allow the algorithm to explore the current optimal solution, making the algorithm more exploitable. As the iteration progresses, the increase in the number of leaders can generate larger disturbances in the solution space, which enhances the global optimization ability of the algorithm. Denote the number of leaders and followers by N_1 and N_2, respectively

$$N_1 = \lceil (t/T) \times (N/2) \rceil, \quad (36)$$

$$N_2 = N - N_1, \quad (37)$$

where $\lceil\rceil$ denotes rounding up.

2) REVERSE LEARNING

In this paper, we combine the law of refraction of light to find the optimal solution based on reverse learning. The refractive backward learning mechanism is a better improvement mechanism to optimize the initial assignment and optimize the candidate solutions. It is able to better optimize the location area and direction when the followers of the bottle sea squirt

population are updated in their location. The specific equation is as follows.

$$T_{1,i}^{l+1,n^*} = \begin{cases} \frac{ub_n^i + lb_n^i}{2} + \frac{ub_n^i + lb_n^i}{2k} - \frac{T_{1,i}^{l,n-1}}{k} & r \le p_i \\ \frac{ub_n^i + lb_n^i}{2} + \frac{ub_n^i + lb_n^i}{2k} + \frac{T_{1,i}^{l,n-1}}{k} & p_i \le r < 1, \end{cases} \tag{38}$$

where $T_{1,i}^{l+1,n}$ denotes the position of the n-th follower in the current population in the i-th dimension. $T_{1,i}^{l+1,n^*}$ is the refractive inverse solution of $T_{1,i}^{l,n-1}$, ub_n^i and lb_n^i are the minimum and maximum values in the i-th dimension in the search space.

In summary, the pseudo-code of the improved LASSA-RRT algorithm is shown below:

Algorithm: LASSA-RRT
G: maximum number of iteration
pop: Number of populations Output: Optimal flight path

1: population initialization
2: **while(i<G)**
3: random churning in space based on the bottleneck swam algorithm leader sampling
4: finding the nearest location in the tree
5: random sampling based on the location of followers
6: generate new nodes based on step length
7: **if** path is valid *then*
8: add the new point to the tree
9: **else**
10: round off
11: **end if**
12: find the path from the tree,and prune the paths
13: result←calculate the cost of the trajectory
14: i←i+1
15: **end while**
16: perform reverse learning operations
16: Return Tree,result

IV. SIMULATION AND ANALYSIS

A. UAV PLATFORM

This article uses the DJI Royal 2 Industry Advance as the drone platform. The UAV is equipped with a 640 × 512 resolution thermal imaging camera, a 48-megapixel visible light camera, and supports 32x digital zoom to meet the image requirements of storage inspection and detection. In addition, the UAV is equipped with an RTK-GPS module, which can achieve accurate positioning and guarantee the realization of autonomous flight. Table 1 shows the relevant parameters of the UAV.

TABLE 1. UAV-related parameters.

Parameters	Values
UAV quality	1100g
Maximum ascent speed	4 m/s
Maximum descent speed	4 m/s
Maximum horizontal flight speed	14 m/s
Maximum flight altitude	6000 m
Battery Capacity	3850 mAh
Total Energy	59.29 Wh
Rotatable range of the head	tilt and turn：-90°-+30
Angle jitter amount	±0.005°
Gravitational acceleration g	9.8
Power conversion factor K	2.061
Constants C	8.532

B. IRRT ALGORITHM AND PRUNING OPERATION VALIDATION

Firstly, in order to verify the effectiveness of the improved RRT algorithm in path planning, MATLAB2017a is used as the simulation platform in this paper. Simulation experiments are conducted in a 50km×50km environment, exploring a step size of 3 and reducing the obstacle threat to a circle of radius 4. The results of the RRT, LASSA-RRT and pruned LASSA-RRT algorithms are compared by analyzing the search time, path length, and path curvature under the same conditions. Figure 3 shows the operation results of the RRT algorithm, Figure 4 shows the operation results of the LASSA-RRT algorithm, and Figure 5 shows the results after pruning.

TABLE 2. Comparison of simulation data of two methods.

Algorithm / Indicator	RRT	LASSA-RRT	Trimmed and smoothed LASSA-RRT
Average planning time	3.037s s	0.334s	N/A
Average distance	109.67km	82.29km	73.23km
Average number of nodes	35	30	5

The blue paths in the figure are randomly sampled points in the environment, and the red paths are the finalized planning paths. As shown in Table 2, the average planning time of the LASSA-RRT algorithm improved 89% over the RRT algorithm, and the average travel cost of IRRT was 75% of that of the RRT algorithm. It can be seen that in each iteration, the RRT algorithm tends to explore randomly toward the unknown region. The improved RRT algorithm strengthens the orientation of exploration and is able to explore in the direction of *q_goal* in the planning space, reducing the number of redundant points and searches in the wrong direction, decreasing the search time, and enhancing the stability of the

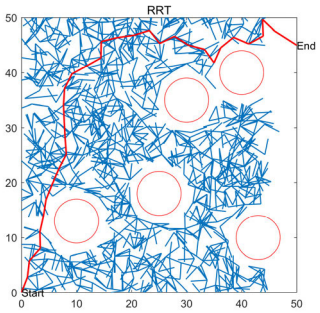


FIGURE 3. The result of RRT algorithm.

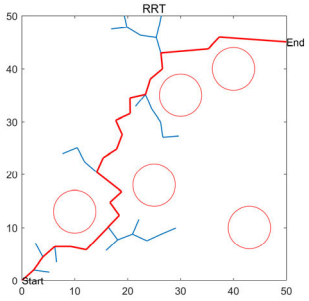


FIGURE 4. The result of LASSA-RRT algorithm.

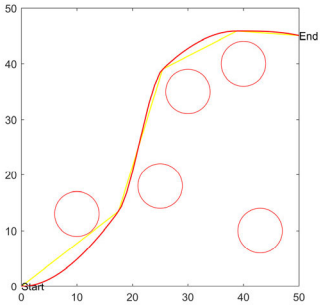


FIGURE 5. The result of Trimmed and smoothed LASSA-RRT algorithm.

path. The above figure shows the route after pruning optimization. After pruning optimization, there are only five path points, and the path length is 73.23km, which is 83% fewer path points and 33% shorter path length. It is proved that the pruning operation can remove a large number of unnecessary redundant nodes, reduce some unnecessary direction changes and shorten the path length. Then after the Bessel curve smoothing, the connection between nodes is smoother and can adapt to the actual flight demand of the UAV.

C. LASSA ALGORITHM VALIDATION

To verify the performance of LASSA algorithm, this paper compares the performance of this algorithm with other heuristic algorithms such as CSA (Crow Search Algorithm), PSO (Particle Swarm Optimization), BES (Bald Eagle Search), and SSA (Sparrow Search Algorithm). Six standard test functions were selected as the fitness functions, and the selected test functions contained functions with different characteristics such as single-peak, multi-peak and fixed-dimensional multi-front functions for testing, and for reliable experimental results. The population sizes were all set to 30, and each standard test function was run 50 times independently for comparison tests, and the objective functions for conducting the tests are shown in Equations (39)-(44).

$$f_1 = \sum_{i=1}^{30} x_i^2 \tag{39}$$

$$f_2 = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j\right)^2 \tag{40}$$

$$f_3 = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right) + 20 + c \tag{41}$$

$$f_4 = \frac{\pi}{30}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{29}(y_i - 1)^2 \times [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4) \tag{42}$$

$$f_5 = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1} \tag{43}$$

$$f_6 = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10 \tag{44}$$

TABLE 3. Test function basic parameters.

Test function	Dimension	Features	Definition Domain	Theoretical optimum
f_1	10	UN	[−100,100]	0
f_2	10	UN	[−100,100]	0
f_3	60	MN	[−32,32]	0
f_4	200	MN	[−50,50]	0
f_5	2	FDM	[−65.53,65.53]	1
f_6	2	FDM	$x_1 \in$[-5,10] $x_2 \in$[0,15]	0.398

The average convergence curves of the six benchmark functions are given in Figures 6 to 11. As can be seen from the figures, LASSA has the fastest convergence speed among the single-peak test functions and has high accuracy in finding the best. LASSA has a flat part in the process of finding the optimum, but it quickly jumps out of the local optimum because the adaptive leader strategy makes LASSA converge quickly from the beginning, and it has a stronger exploration

ability in the early stage while it can jump out of the local optimum in the later stage.

The optimal values in Table 4, the average values reflect the convergence speed and the ability to find the optimal value of the algorithm, and the bold letters indicate the optimal value found when compared with other algorithms. For the single-peak function, the LASSA solution performs the best, and the results all reach the theoretical result values. Due to the complex characteristics of Penalized function, it is more difficult for all algorithms to find the optimal value of the function. As the dimensionality of the solution increases, the solution accuracy of all five algorithms decreases, and when the dimensionality increases to 200 dimensions, the gap between the five algorithms and the ideal value is too large. LASSA reached the theoretical value in five of the tested functions, and the standard deviation was significantly better than the other four algorithms, which verified the effectiveness and better robustness of LASSA.

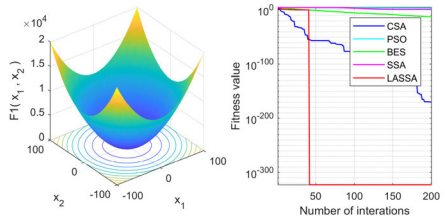


FIGURE 6. f_1: Sphere Function.

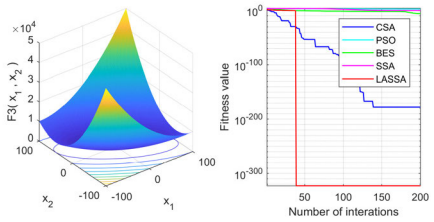


FIGURE 7. f_2: Schwefel's Problem 1.2.

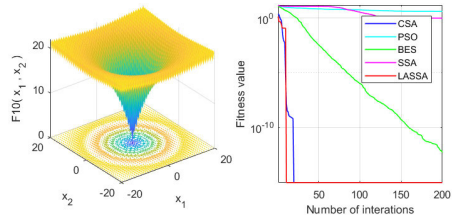


FIGURE 8. f_3:Ackley's Function.

D. LASSA-RRT ALGORITHM VALIDATION

To verify the practicality and effectiveness of the proposed LASSA-RRT algorithm for UAVs to traverse terrain obstacles and complete inspection tasks in open-air warehouses.

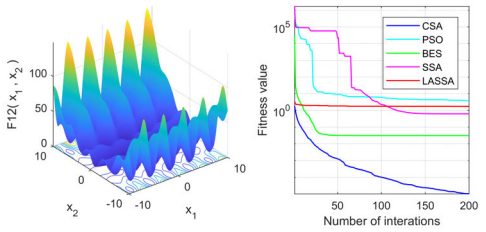


FIGURE 9. f_4:Penalized Function.

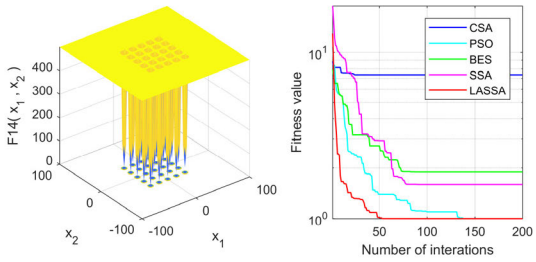


FIGURE 10. f_5:Shekel's Foxhole Function.

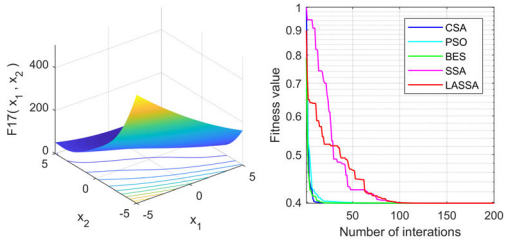


FIGURE 11. f_6:Branin Function.

This section introduces three scenarios with different levels (Shen et al. proposed to measure the complexity of the environment by calculating the information entropy of the environment map by computing the aggregated features of the gray distribution of the images [31], the higher the level, the higher the entropy, the more complex the environment) and designs comparative experiments for comparative analysis. Complete the simulation experiment in MATLAB2017a. As shown in Figure 13 the size of the planning space is set to 1000 × 1000 x 400, the unit length is 1m. Starting from the initial position and returning to the starting point after completing the inspection task at all task points, where blank areas represent safe areas, rectangles, cylinders and spheres indicate obstacles, and circles indicate the task points to be reached. Table 6 shows the parameters in the scenario. We selected five algorithms, standard RRT, RRT* [17], IRRT [19], PF-RRT* [20] and the LASSA-RRT proposed in this paper, for simulation experiments and Table 5 shows the values of relevant parameters taken in the model, where it is assumed that the parameters in the flight of the UAV are consistent with the theoretical values.

We selected the standard RRT, RRT* [17], IRRT [19], PF-RRT* [20] and the LASSA-RRT algorithm proposed in

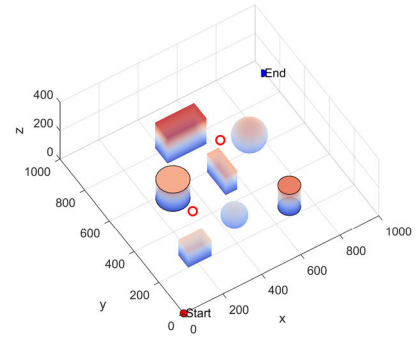
TABLE 4. Test function verification results.

Algorithm	Function	Optimal	Average	Standard deviation
LASSA		**0.00e+00**	**0.00e+00**	**0.00e+00**
SSA		2.6413	0.4952	1.5932
BES	f_1	7.20e-19	1.28e-14	2.39e-14
CSA		**0.00e+00**	8.94e-246	**0.00e+00**
PSO		385.990	844.111	217.220
LASSA		**0.00e+00**	**0.00e+00**	**0.00e+00**
SSA		0.0023	0.1571	0.2272
BES	f_2	8.49e-17	6.30e-08	1.98e-07
CSA		**0.00e+00**	2.36e-196	**0.00e+00**
PSO		72.706	141.245	95.946
LASSA		**8.88e-16**	**8.88e-16**	**0.00e+00**
SSA		9.60e-06	0.5673	0.9189
BES	f_3	4.44e-15	9.99e-13	2.06e-12
CSA		7.83e-15	4.37e-12	1.35e-10
PSO		2.2103	3.4954	0.9447
LASSA		0.68722	1.4687	0.66106
SSA		0.0058997	1.6627	1.3996
BES	f_4	**2.39e-31**	0.0311	0.0983
CSA		1.84e-06	**1.21e-05**	**9.06e-06**
PSO		0.6960	2.9395	1.9872
LASSA		**0.998**	**0.998**	3.76e-08
SSA		**0.998**	1.7916	1.0233
BES	f_5	**0.998**	3.1657	2.7638
CSA		**0.998**	6.5569	5.4585
PSO		**0.998**	**0.998**	**1.36e-08**
LASSA		**0.39789**	**0.39789**	**1.00e-13**
SSA		**0.39789**	**0.39789**	1.24e-13
BES	f_6	**0.39789**	**0.39789**	2.33e-06
CSA		**0.39789**	**0.39789**	3.32e-07
PSO		**0.39789**	**0.39789**	8.49e-06

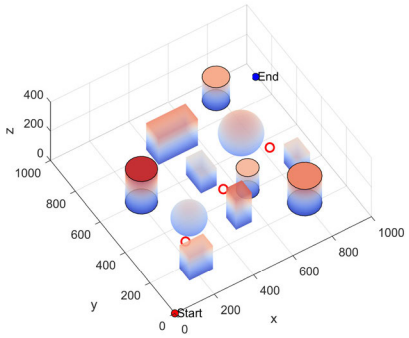
TABLE 5. Algorithm-related parameters.

Parameters	Values
UAV quality	1100g
Maximum ascent speed	4 m/s
Maximum descent speed	4 m/s
Maximum horizontal flight speed	14 m/s
Gravitational acceleration g	9.8
Power conversion factor k	2.061
Constants C	8.532
Air resistance force area s	0.1m^2
Air Density ρ	1.29kg/m^3
Resistance factor c	0.4
Relative wind speed v_a	10m/s
Population size N	50
Maximum number of iterations l_{max}	200
Maximum number of samples k_{max}	2000

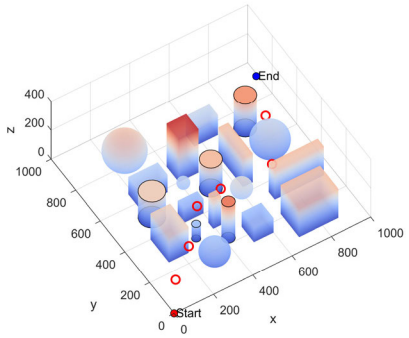
this paper, a total of five algorithms for simulation experiments, and the pseudo-codes of several algorithms are as follows.



(a) Scenario A and a schematic diagram of the inspection task



(b) Scenario B and a schematic diagram of the inspection task

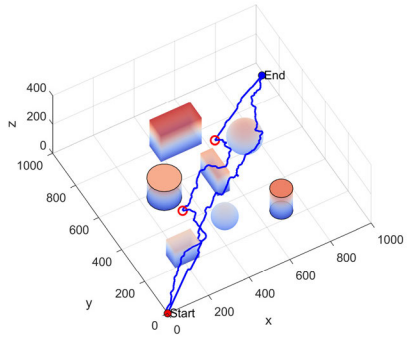


(c) Scenario C and a schematic diagram of the inspection task

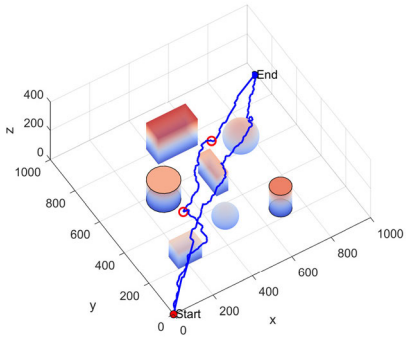
FIGURE 12. Schematic diagram of three scenarios.

Fig. 12 shows a schematic diagram of the three scenarios of different complexity. RRT, RRT*, IRRT, PF-RRT*, and LASSA-RRT were run 50 times in each of the three scenarios. Table 7 shows the average cost, average run time, average path length, and the average number of samples for each algorithm in the three scenarios. The results of the five algorithms in scenario A are shown in Fig.13; The results of the five algorithms in scenario B are shown in Fig.14; The results of the five algorithms in scenario C are shown in Fig.15. Where the blue line is the UAV trajectory.

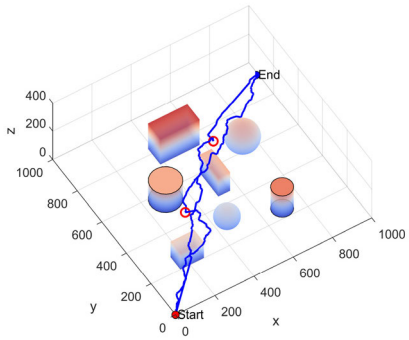
Since the sampling strategy of IRRT is still based on RRT, it has a long search time and many path redundancy points. RRT* and PF-RRT* add the process of reselecting the parent



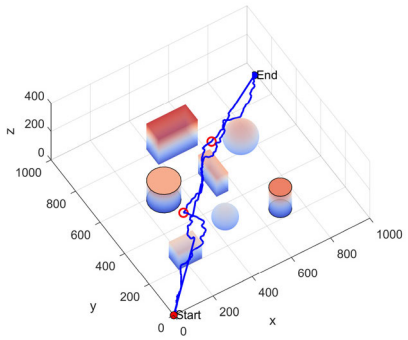
(a) The result of RRT



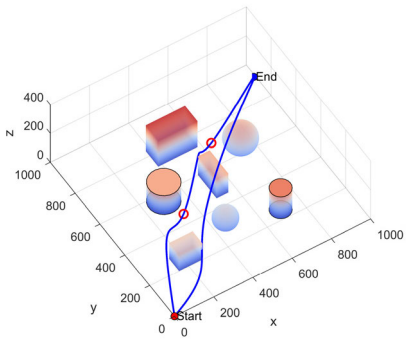
(b) The result of RRT*



(c) The result of IRRT



(d) The result of PF-RRT*



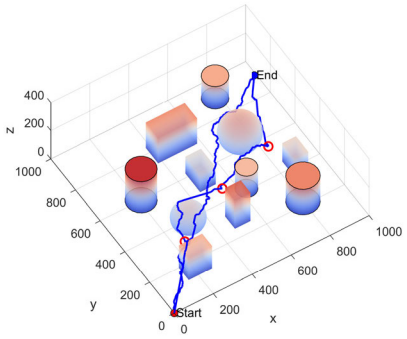
(e) The result of LASSA-RRT

FIGURE 13. Running results of each algorithm under scenario A.

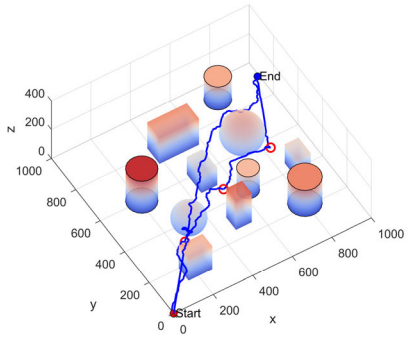
node in the sampling process, the route is straighter, and the search efficiency of the algorithm is improved because PF-RRT* and LASSA-RRT introduce target-based probabilistic sampling. The improved bottle swarm search strategy added by LASSA-RRT enhances the planning capability of the algorithm in the later stages and reduces the number of samples, enabling the algorithm to plan trajectories more effectively in complex scenarios. Pruning and smoothing operations make the planned paths smoother and conform to UAV dynamics constraints.

As the entropy of the scenario increases, it is difficult for all the above algorithms to find good paths in a scenario full of obstacles. As can be seen from Table 7, LASSA-RRT is able to find the fastest solutions in all three scenarios. The average running time of 50 experiments in Scenario 3 is 1.6849s, which is 72% (RRT), 42% (RRT*), 59% (IRRT), and 27% (PF-RRT*) better than the other four algorithms. The total cost, average path length and the average number of samples are reduced by 120,000, 164.3m and 412 times respectively compared to the PF-RRT* algorithm. These data demonstrate the significant superiority of LASSA-RRT for planning tasks in complex scenarios.

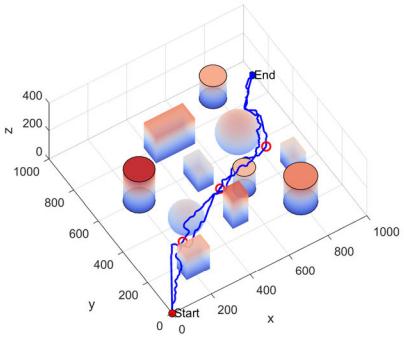
According to the trajectory plots of several algorithms in 3 scenarios, it can be seen that algorithms such as RRT tend to



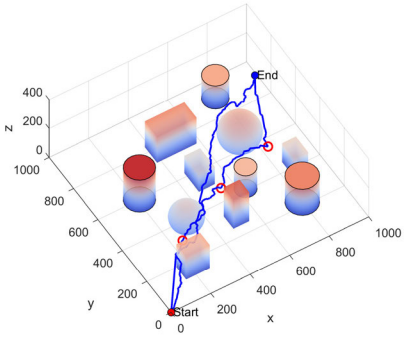
(a) The result of RRT



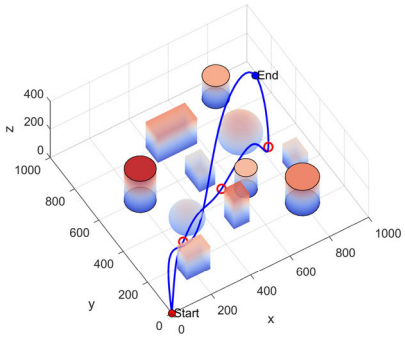
(b) The result of RRT*



(c) The result of IRRT



(d) The result of PF-RRT*



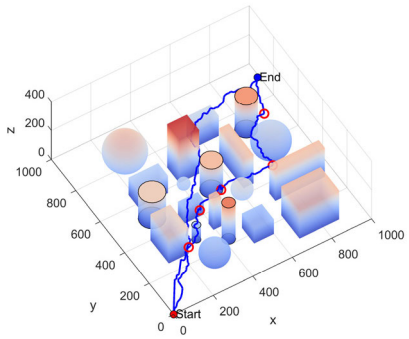
(e) The result of LASSA-RRT

FIGURE 14. Running results of each algorithm under scenario B.

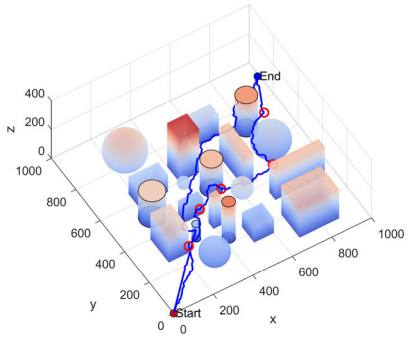
fall into a local minimum in more complex scenarios, resulting in low search efficiency. The trajectory of the proposed algorithm is smoother and more efficient. Compared with other algorithms, the algorithm in this paper has great advantages, reducing the path redundancy points, less running time and less sampling. The results show that LASSA-RRT has the best performance in all scenarios and can well accomplish the open-air warehouse inspection task proposed in this paper.

As shown in Fig.16 and Table 7, the designed LASSA-RRT algorithm reduces 0.18%, 0.14%, 0.17% and 0.06%, respectively, in terms of total cost compared to the RRT, RRT*, IRRT and PF-RRT* algorithms in Scenario A, with an average reduction of 0.1375%. There were 86%, 73%, 56%, and 20% reductions in run time, with an average reduction of 58.75%; 16%, 15%, 10%, and 9% reductions in path length, with an average reduction of 12.5%; and 96%, 22%, 79%, and 14% reductions in the number of samples, with an average reduction of 52.75%.

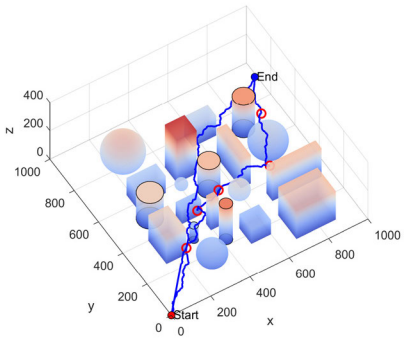
In Scenario B, the designed LASSA-RRT algorithm reduces the total cost by 0.17%, 0.16%, 0.14% and 0.11%, respectively, compared to the RRT, RRT*, IRRT and PF-RRT* algorithms, with an average reduction of 0.145%.



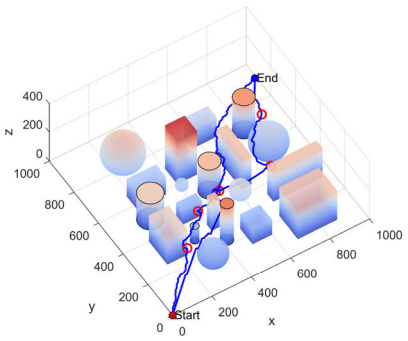
(a) The result of RRT



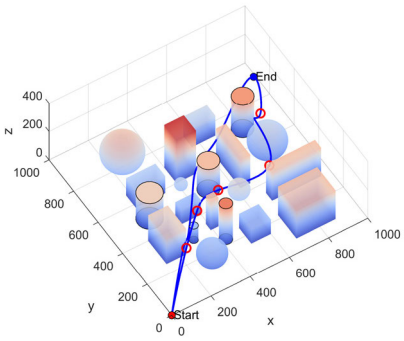
(b) The result of RRT*



(c) The result of IRRT



(d) The result of PF-RRT*



(e) The result of LASSA-RRT

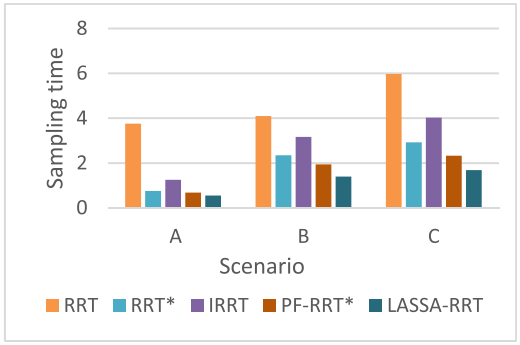
FIGURE 15. Running results of each algorithm under scenario C.

There were reductions of 65%, 40%, 56%, and 28% in run time, with an average reduction of 47.25%; 21%, 17%, 13%, and 6% in path length, with an average reduction of 14.25%; and 94%, 34%, 78%, and 22% in the number of samples, with an average reduction of 57%.

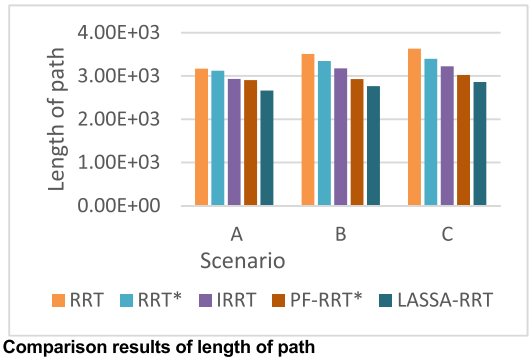
In Scenario C, the designed LASSA-RRT algorithm reduces the total cost by 0.21%, 0.18%, 0.15% and 0.12%, respectively, compared to the RRT, RRT*, IRRT and PF-RRT* algorithms, with an average reduction of 0.165%. The reductions in run time were 72%, 42%, 58% and 27%, with an average reduction of 49.75%; in path length were 21%, 16%, 11% and 5%, with an average reduction of 13.25%; and in sampling frequency were 94%, 30%, 82% and 25%, with an average reduction of 57.75%.

The feasible paths planned based on the RRT* algorithm (RRT*, PF-RRT*) outperform the paths generated based on the RRT algorithm (RRT, IRRT), and LASSA-RRT outperforms other algorithms based on the total cost of flight, running time, path length, and number of samples in all cases.

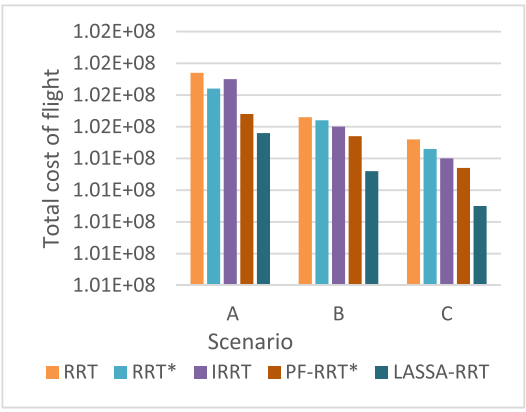
In the three environments with different complexity, LASSA-RRT has an average reduction of 55.83% in sampling times, 51.91% in run time, 13.17% in track length, and 0.1491% in flight cost compared to the other four algorithms. This shows that the design of this paper introduces



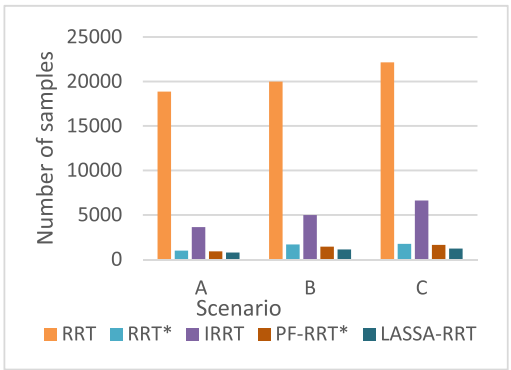
(a) Comparison results of sampling time



(b) Comparison results of length of path



(c) Comparison results of total cost of flight



(d) Comparison results of number of samples

FIGURE 16. Comparison results of five algorithms.

an improved search strategy for the bottle sea squirt group that can effectively improve the quality of the path nodes. The

TABLE 6. Parameters in the scenario.

Scenario / Parameter	A	B	C
Start Point	(0,0,0)	(0,0,0)	(0,0,0)
End Point	(900,850,100)	(900,850,100)	(900,850,100)
Mission Points	(300,400,80) (600,650,100)	(200,230,120) (500,400,100) (800,500,80)	(700,100,80) (200,200,120) (355,380,100) (500,420,80) (730,370,150) (860,630,120)
Information entropy	0.8100	1.1143	1.4256

Algorithm: RRT∗

1: population initialization
2: **while(i<G)**
3: *q_rand*←random churning in space
4: *q_near*←finding the nearest location in the tree
5: *q_new*←generate new nodes
6: **if** path is valid **then**
7: add the new point to the tree
8: **else**
9: round off
10: **end if**
11: *q_parent* ←find a new parent node
12: rewrite the path
13: result←calculate the cost of the trajectory
14: i←i+1
15: **end while**
16: Return Tree,result

TABLE 7. Simulation results.

Scenario	Algorithm	Total Cost	Running time	Path length	Sampling Number
A	RRT	1.0167e+08	3.7543	3.1696e+03	18852
	RRT*	1.0162e+08	0.7528	3.1219e+03	992
	IRRT	1.0165e+08	1.2486	2.9317e+03	3625
	PF-RRT*	1.0154e+08	0.6828	2.9046e+03	903
	LASSA-RRT	1.0148e+08	0.5497	2.6615e+03	778
B	RRT	1.0153e+08	4.0942	3.5082e+03	19982
	RRT*	1.0152e+08	2.3418	3.3438e+03	1683
	IRRT	1.0150e+08	3.1651	3.1753e+03	4984
	PF-RRT*	1.0147e+08	1.9362	2.9251e+03	1429
	LASSA-RRT	1.0136e+08	1.3943	2.7630e+03	1113
C	RRT	1.0146e+08	5.9674	3.6285e+03	22135
	RRT*	1.0143e+08	2.9216	3.3940e+03	1754
	IRRT	1.0140e+08	4.0231	3.2234e+03	6619
	PF-RRT*	1.0137e+08	2.3219	3.0216e+03	1633
	LASSA-RRT	1.0125e+08	1.6849	2.8573e+03	1221

pruning and smoothing strategies can make the paths conform more to the dynamical constraints and optimize the trajectory results effectively.

In summary, this paper presents the inspection planning method of the LASSA-RRT algorithm based on adaptive leader strategy and reverse search strategy under a complex open-air warehouse. Based on the random sampling of RRT, the exploration strategy of the population in the bottle sheath swarm algorithm is introduced, and the corresponding

Algorithm: IRRT

1: population initialization
2: **while(i<G)**
3: *q_rand*←random churning in space
4: *q_near*←finding the nearest location in the tree
5: *q_new*←generate new nodes
6: Selection of the root node according to the improved strategy
7: **if** path is valid **then**
8: add the new point to the tree
9: **else**
10: round off
11: **end if**
12: result←calculate the cost of the trajectory
13: i←i+1
14: **end while**
15: Return Tree,result

Algorithm: PF-RRT∗

1: population initialization
2: **while(i<G)**
3: *q_rand*←random sampling based on probability
4: *q_near*←finding the nearest location in the tree
5: *q_new*←Generation of new nodes based on artificial potential fields
6: **if** path is valid **then**
7: add the new point to the tree
8: **else**
9: round off
10: **end if**
11: *q_parent* ←find a new parent node based on dichotomy
12: Rewrite the path
13: result←calculate the cost of the trajectory
14: i←i+1
15: **end while**
16: Return Tree,result

sampling method is selected according to the complexity of the environment, so that LASSA-RRT can plan a lower-cost inspection path in a shorter time. The adaptive leader and reverse search strategies improve the problem of easily falling into local minima in complex environments and increase the efficiency of tree search. Finally pruning and smoothing operations reduce the redundant points in the random tree and make the flight path smoother to better meet the flight requirements of UAVs.

V. CONCLUSION

This study explores the planning method for UAV inspection in open-air warehouses and proposes a LASSA-RRT algorithm based on an improved SSA algorithm. To verify the effectiveness and optimality of the algorithm, LASSA-RRT is compared with four other advanced algorithms under three environments. It is proved that the algorithm proposed in this paper has the following advantages, (I) The LASSA-RRT algorithm optimizes the sampling strategy and reduces the sampling of invalid nodes. The CPU running time is greatly reduced. (II) The number of nodes of the random tree is reduced and the path length is much lower than that of PF-RRT*, etc. (III) The paths are pruned and optimized to meet the UAV dynamics constraints considering the actual needs of UAV flight. The algorithm proposed in this paper can effectively optimize the search path and reduce the power cost and time cost of the UAV in performing complex open-air warehouse inspection tasks. In future research, we will combine the path planning of UAVs with the dynamic environment, and plan the flight path through real-time feedback on the environment to improve the response capability of UAVs to emergencies and provide more favorable security and technical support to warehouse managers.

REFERENCES

[1] H.-M. Chung, S. Maharjan, Y. Zhang, F. Eliassen, and K. Strunz, "Placement and routing optimization for automated inspection with unmanned aerial vehicles: A study in offshore wind farm," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3032–3043, May 2021, doi: 10.1109/TII.2020.3004816.

[2] K. Kuru, D. Ansell, W. Khan, and H. Yetgin, "Analysis and optimization of unmanned aerial vehicle swarms in logistics: An intelligent delivery platform," *IEEE Access*, vol. 7, pp. 15804–15831, 2019, doi: 10.1109/ACCESS.2019.2892716.

[3] D. Zhai, C. Wang, H. Cao, S. Garg, M. M. Hassan, and S. A. AlQahtani, "Deep neural network based UAV deployment and dynamic power control for 6G-envisioned intelligent warehouse logistics system," *Future Gener. Comput. Syst.*, vol. 137, pp. 164–172, Dec. 2022.

[4] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1441–1456, May 2022, doi: 10.1109/JSAC.2022.3143252.

[5] M. Sun, X. Xu, X. Qin, and P. Zhang, "AoI-energy-aware UAV-assisted data collection for IoT networks: A deep reinforcement learning method," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17275–17289, Dec. 2021, doi: 10.1109/JIOT.2021.3078701.

[6] C. Yan, L. Fu, J. Zhang, and J. Wang, "A comprehensive survey on UAV communication channel modeling," *IEEE Access*, vol. 7, pp. 107769–107792, 2019, doi: 10.1109/ACCESS.2019.2933173.

[7] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020, doi: 10.1109/TWC.2019.2940447.

[8] W. Wei, J. Wang, Z. Fang, J. Chen, Y. Ren, and Y. Dong, "3U: Joint design of UAV-USV-UUV networks for cooperative target hunting," *IEEE Trans. Veh. Technol.*, early access, Nov. 89, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9944188, doi: 10.1109/TVT.2022.3220856.

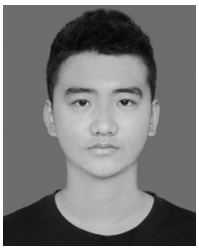
[9] H. Liu, Y. Sun, J. Cao, S. Chen, N. Pan, Y. Dai, and D. Pan, "Study on UAV parallel planning system for transmission line project acceptance under the background of Industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5537–5546, Aug. 2022, doi: 10.1109/TII.2022.3142723.

[10] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. M.-D. Dios, F. Pierri, J. Cortes, A. Santamaria-Navarro, M. A. T. Soto, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez, "The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance," *IEEE Robot. Autom. Mag.*, vol. 25, no. 4, pp. 12–23, Dec. 2018, doi: 10.1109/MRA.2018.2852789.

[11] D.-H. Tran, T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Coarse trajectory design for energy minimization in UAV-enabled," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9483–9496, Sep. 2020, doi: 10.1109/TVT.2020.3001403.

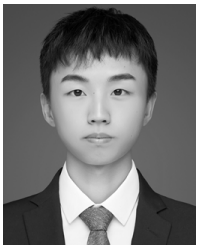
[12] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, and Y. Yu, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm," *Aerosp. Sci. Technol.*, vol. 70, pp. 497–510, Nov. 2017.

[13] H. Liu, Q. Chen, N. Pan, Y. Sun, Y. An, and D. Pan, "UAV stocktaking task-planning for industrial warehouses based on the improved hybrid differential evolution algorithm," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 582–591, Jan. 2022, doi: 10.1109/TII.2021.3054172.
[14] P. Kumar, S. Garg, A. Singh, S. Batra, N. Kumar, and I. You, "MVO-based 2-D path planning scheme for providing quality of service in UAV environment," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1698–1707, Jun. 2018, doi: 10.1109/JIOT.2018.2796243.
[15] N. Wen, L. Zhao, X. Su, and P. Ma, "UAV online path planning algorithm in a low altitude dangerous environment," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 2, pp. 173–185, Apr. 2015, doi: 10.1109/JAS.2015.7081657.
[16] Z. Wang, X. Xiang, J. Yang, and S. Yang, "Composite Astar and B-spline algorithm for path planning of autonomous underwater vehicle," in *Proc. IEEE 7th Int. Conf. Underwater Syst. Technol., Theory Appl. (USYS)*, Dec. 2017, pp. 1–6, doi: 10.1109/USYS.2017.8309463.
[17] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011, doi: 10.1177/0278364911406761.
[18] M. Li, "Research on UAV mission planning methods based on intelligent optimization and RRT algorithm," Ph.D. dissertation, Nanjing Univ. Aeronaut. Astronaut., Nanjing, China, 2012.
[19] G. Y. Yin, S. L. Zhou, and Q. P. Wu, "An improver RRT algorithm for UAV path planning," *Atca Electronica Sinica*, vol. 45, no. 7, pp. 1764–1769, 2017.
[20] J. Fan, X. Chen, Y. Wang, and X. Chen, "UAV trajectory planning in cluttered environments based on PF-RRT algorithm with goal-biased strategy," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105182.
[21] J. Huang and W. Sun, "A method of feasible trajectory planning for UAV formation based on bi-directional fast search tree," *Optik*, vol. 221, Nov. 2020, Art. no. 165213.
[22] H. Tong, T. Andi, Z. Huan, X. Dengwu, and X. Lei, "Multiple UAV cooperative path planning based on LASSA method," *Syst. Eng. Electron.*, vol. 44, no. 1, pp. 233–241, 2022.
[23] J.-S. Pan, J.-X. Lv, L.-J. Yan, S.-W. Weng, S.-C. Chu, and J.-K. Xue, "Golden eagle optimizer with double learning strategies for 3D path planning of UAV in power inspection," *Math. Comput. Simul.*, vol. 193, pp. 509–532, Mar. 2022.
[24] D. Chen and X. Y. Meng, "UAV offline path planning based on self-adaptive coyote optimization algorithm," *Syst. Eng. Electron.*, vol. 44, no. 2, pp. 603–611, 2022.
[25] C. J. Zhang and X. J. Meng, "A path planning method based on hybrid salp swaem-differenctial evolution algorithm," *Flight Dyn.*, vol. 38, no. 6, pp. 49–55, 2020, doi: 10.13645/j.cnki.f.d.20200602.009.
[26] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial internet of UAVs for power line inspection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2705–2714, Jun. 2018.
[27] F. A. N. Yeman, S. H. E. N. Kaicheng, W. A. N. G. Dong, Z. H. A. I. Changyuan, and Z. H. A. N. Haihui, "Optimal energy consumption path planning of UAV on mountain region based on simulated annealing algorithm," *Trans. Chinese Soc. Agricult. Machinery*, vol. 51, no. 10, pp. 1–8, 2020, doi: 10.6041/j.issn.1000-1298.2020.10.004.
[28] C. X. Xiang and J. Mao, "Modeling and simulation of rotor aerodynamics for four-rotor UAV," *Comput. Simul.*, vol. 38, no. 10, pp. 48–52, 2021, doi: 10.3969/j.issn.1006-9348.2021.10.010.
[29] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Comput. Sci. Dept., Iowa State Univ., Ames, IA, USA, Tech. Rep. 14744621, 1998.
[30] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, "Asynchronous accelerating multi-leader salp chains for feature selection," *Appl. Soft Comput.*, vol. 71, pp. 964–979, Oct. 2018.
[31] B. Shen, "Evaluation models and methods on the intelligence of unmanned swarm systems based on collective OODA," *Acta Aeronautica et Astronautica Sinica*, vol. 40, pp. 1–16, Mar. 2023, doi: 10.7527/S1000-6893.2022.28003.



JINGCHENG ZHANG was born in Qujing, Yunnan, China. He is currently pursuing the bachelor's degree in data science and big data technology with the School of Science, Kunming University of Science and Technology, Kunming, China. His research interests include group intelligence, data mining, and in-depth learning.

YUQIANG AN was born in Guiyang, Guizhou, China, in 1985. He received the master's degree from the Kunming University of Science and Technology, Kunming, China, in July 2011. Since 2011, he has been a Deputy Chief and an Engineer with Hongyun Honghe Tobacco (Group) Company Ltd. His research interests include artificial intelligence and intelligent systems.



JIANING CAO was born in Taiyuan, Shanxi, China. He is currently pursuing the B.E. degree in mechanical engineering with the Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology, Kunming, China. His main research interests include swarm intelligence and deep learning.

SHIBO OUYANG was born in Zhaotong, Yunnan, China, in 1982. He received the master's degree from Yunnan University, in July 2009. He is currently an Engineer with Hongyun Honghe Tobacco (Group) Company Ltd. His main research interest includes logistics and system analysis.

LEI WANG was born in Kaiyuan, Yunnan, in November 1976. He received the bachelor's degree in information and electronic science. He is currently an Engineer. His research interest includes logistics engineering.