An Efficient UAV Coverage Path Planning Method for 3-D Structures

Jiaxin Du, Baoqi Huang, *Senior Member, IEEE*, and Bing Jia, *Member, IEEE*

***Abstract*—The use of uncrewed aerial vehicles (UAVs) to perceive surface images and other information of complex 3-D structures is a key component in infrastructure inspection tasks, such as for wind turbines and bridges. As a result, UAV path planning for 3-D structure coverage has attracted significant attention from researchers. However, existing studies primarily focus on minimizing the viewpoint connection path length under the full coverage condition, neglecting UAV flight constraints, which leads to a significant discrepancy between the planned path and the actual UAV flight trajectory. This article presents an efficient UAV path-planning method for 3-D structure coverage using an improved ant colony optimization (ACO) algorithm. A multiconstraint UAV flight-trajectory-length minimization problem is formulated to guarantee full coverage of 3-D surfaces under kinematic limits, turning-angle and attitude-rotation constraints, thereby enhancing path consistency and smoothness. A cost function based on Minimum Snap trajectory planning replaces simple waypoint-connection length to more accurately reflect actual flight effort. A beta distribution-driven ant state transition rule and an adaptive pheromone evaporation strategy mitigate the tendency of traditional ACO to stall in local optima, balancing exploration and exploitation for rapid and reliable convergence. Simulation results demonstrate that the method proposed in this article outperforms existing typical 3-D structure coverage path-planning methods, validating its feasibility and effectiveness.**

***Index Terms*—Ant colony optimization-optimized parameter distribution (ACO-OPD), coverage path planning (CPP), uncrewed aerial vehicle (UAV).**

I. Introduction

In RECENT years, with the rapid development of the uncrewed aerial vehicle (UAV) industry, they have been widely used as inspection tools in various scenarios, such as bridges [1], [2], buildings [3], [4], [5], power transmission lines [6], and wind turbines [7]. Consequently, methods for UAV coverage path planning (CPP) have become mainstream. Initial CPP missions focused on targeted 2-D spatial frameworks, such as those used by painting robots and lawnmowers, with the objective of covering planar surfaces. However, with the increasing demand for coverage mission across various industries, 2-D CPP has evolved into more 3-D spatial scenarios. 3-D CPP tasks can be divided into two categories: 1) offline and 2) online. Offline tasks require prior knowledge of the 3-D structure to ensure complete coverage, whereas online tasks, typically employed in unknown environments, are used for applications like 3-D reconstruction. Considering that offline methods typically yield superior path performance, we have chosen the offline approach for 3-D CPP.

Building on the offline framework, we define a scenario in which a single UAV achieves full coverage of a 3-D structure represented by a mesh model. This is accomplished by following a path generated from a subset of viewpoints selected from a candidate set derived from the mesh. Given that the selection and sequencing of these viewpoints directly affect path efficiency, various swarm intelligence optimization algorithms, including particle swarm optimization (PSO), genetic algorithm (GA), and ant colony optimization (ACO), have been widely used in recent years to optimize trajectory planning. However, related studies [8], [9], [10] usually ignored the constraint of the UAV kinematic, e.g., reasonable turning angles and attitude rotations of the UAV during the planning process, leading to poor applicability for actual flight.

To address this issue, this study proposes a constrained optimization problem, aiming to minimize the path length under coverage constraint and UAV kinematic constraints. First, to explore multiple candidate solutions for the viewpoint sequence, the next viewpoint is gradually selected from a uniformly distributed candidate viewpoint set based on the coverage of the already selected viewpoints. Second, to bridge the gap between theoretical optimization and the actual flight path length, we employ a minimum snap trajectory generation method with temporal allocation. Given the NP-hard nature of the problem, heuristic algorithms are typically employed for its resolution. Inspired by [11], which demonstrated that the ACO algorithm can effectively handle dynamically changing candidate solution sets, but did not consider the constraints imposed by fixed heuristic parameters on search capability, as well as the limitations of a fixed evaporation rate on dynamic adaptability resulting in being prone to local optima. We improve the ACO algorithm by optimizing the distribution of heuristic parameters and the pheromone update rules, thereby avoiding local optima caused by fixed parameter settings. Simulation experiments confirm that the proposed method

Received 2 April 2025; revised 11 May 2025; accepted 24 May 2025. Date of publication 2 June 2025; date of current version 25 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62262046 and Grant 42161070; in part by the Science and Technology Plan Project of Inner Mongolia A. R. of China under Grant 2024KJHZ0003, Grant 2023YFSW0017, and Grant 2023KJHZ0016; in part by the Ordos Science and Technology Plan under Grant YF20240029; in part by the Fund of supporting the Reform and Development of Local Universities (Disciplinary construction); in part by the Natural Science Foundation of Inner Mongolia A.R. of China under Grant 2023MS06004; and in part by the fund of First-class Discipline Special Research Project of Inner Mongolia A. R. of China under Grant YLXKZX ND-036. *(Corresponding author: Baoqi Huang.)*

The authors are with the School of Computer Science, Inner Mongolia University, Hohhot 010021, China (e-mail: 32209036@mail.imu.edu.cn; cshbq@imu.edu.cn; jiabing@imu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2025.3575374

offers significant advantages over existing approaches in terms of reducing path length.

In summary, the main contributions of this article are as follows.

1) This study introduces a 3-D structure CPP algorithm that utilizes an offset-based method to generate a set of candidate viewpoints and dynamically selects viewpoints during the path planning process. Under coverage constraints, it addresses the optimization of the number of viewpoints from a global perspective.
2) Based on a newly designed objective function that minimizes path length under the coverage constraint and the UAV kinematic constraints, including turning angle and attitude rotation, the ACO with optimized parameter distribution (ACO-OPD) algorithm effectively enhances the kinematic feasibility of coverage paths for 3-D structures.
3) Using the minimum snap method, trajectory optimization improved the quality of the planned path, with feasibility validated through simulations on three 3-D structures: a) Wind Turbine; b) Big Ben; and c) Christ.

II. Related Work

Generally, the primary means of addressing path planning problems is to establish a mathematical model and solve it. Based on the performance constraints of the UAV and the geographic environment of the workspace, a model is constructed and then an optimal path is derived using the corresponding algorithms.

Traditional layered coverage methods for 3-D structures, such as wind turbines, have made strides in addressing CPP. However, they often face difficulties with complex-shaped models. To address these limitations, recent advances have seen a shift toward grid-based approaches, which involve sampling viewpoints from triangular mesh representations. This technique optimizes their locations to meet sensor constraints while minimizing overall path costs, thus improving the efficiency of the coverage mission. In this context, various algorithms have been employed to tackle the traveling salesman problem (TSP), which is essential for efficient path planning. Similarly, Helsgaun [12] first proposed the Lin–Kernighan heuristic (LKH) algorithm for the TSP, greatly enhancing the efficiency of finding optimal or near-optimal solutions, especially for large instances. However, it can only solve problems with a fixed set of path points. Cao et al. [13] introduced a multiresolution hierarchical framework, effectively using the LKH algorithm across varying resolutions of subspaces. However, this approach required additional viewpoints to ensure comprehensive coverage, highlighting the ongoing challenge of balancing coverage and efficiency. Furthermore, Bircher et al. [14] utilized a triangular mesh representation along with a two-step optimization process. Their method carefully selects viewpoints that ensure full coverage while minimizing connection costs. However, due to the lack of optimization for the number of viewpoints, redundant viewpoints have a significant impact on the path length. Feng et al. [15] employed a skeleton-based space decomposition (SSD) technique, breaking the scene into simpler subspaces. This innovative approach facilitates efficient identification of free space and enables the generation of a minimal set of specialized viewpoints, ensuring comprehensive coverage with optimal resource utilization.

In addition, the application of swarm intelligence has been explored in the domains of path planning, further demonstrating the flexibility and potential of these techniques. For example, Bui et al. [9] proposed a swarm intelligence-based UAV path planning approach using 3-D models and ACO, Li et al. [8] proposed an improved ACO with variable pheromone (ACO-VP) algorithm and a greedy allocation strategy for efficient UAV path planning in dynamic environments, demonstrating enhanced performance in target coverage tasks compared to other algorithms. Yan et al. [16] proposed a chaotic-polarized-simulated scheme in the ACO (CPS-ACO) algorithm, which combines 3-D terrain modeling and dynamic obstacle avoidance strategies for trajectory planning in complex maritime environments. The method significantly enhances flight efficiency and path kinematic feasibility of UAVs in edge computing networks, while ensuring obstacle avoidance safety, through chaos-based initialization, elite path selection, and geometric detouring strategies. Hashim and Hussien [17] proposed a swarm intelligence-based metaheuristic, the snake optimization (SO) algorithm, which simulates snake mating competition in dynamic environments. It uses adaptive parameters, such as temperature decay and food thresholds, to balance global search and local exploitation. And Yao et al. [18] proposed the enhance SO (ESO) algorithm that improves performance by incorporating an opposition-based learning strategy and dynamic update mechanisms, including parameter adjustment, sine-cosine perturbation, Tent-chaos and Cauchy mutation. Li et al. [19] introduced the heuristic smoothing ACO with differential information (HSDACO), which improves traditional ACO algorithm by incorporating heterogeneous populations, smoothing techniques, differential information updates and evolutionary state adjustments, achieving enhanced solution accuracy and faster convergence for small and mid-scale TSP instances, Jiao et al. [20] enhanced intelligent wheelchair navigation by employing ACO algorithm with adaptive state transitions and dynamic information updates. This resulted in improved accuracy and stability in route planning, illustrating how ACO algorithm can be tailored to meet the specific requirements of different mobility challenges. Chen et al. [21], [22] proposed two approaches. The first approach uses spatiotemporal clustering combined with a GA and static load balancing for cluster center allocation. Generates initial paths using a greedy strategy, optimized through an evolutionary algorithm, with a focus on real-time task processing in 2-D static scenarios, although it is limited by local optimization. The second approach integrates adaptive clustering with a symbiotic biological search algorithm, dynamically selecting cluster centers, and assigning tasks via a parameter-free optimization algorithm. This method supports 3-D environment expansion, dynamic threat avoidance, and high-precision requirements for complex scenarios. The first approach excels

in static scenario efficiency and real-time performance, while the second shines in dynamic adaptability and complex constraint handling. Gong et al. [23] proposed a generalized approach for optimizing energy consumption and UAV count in 3-D scenarios, solved using successive convex approximation (SCA) algorithm and modified PSO (MPSO) algorithms. Considering energy consumption is highly important for path planning. Collectively, these studies illustrate the evolving landscape of CPP methodologies, emphasizing the potential of integrating various optimization techniques to address the unique challenges posed by 3-D structures and improve the effectiveness of UAV operations.

To overcome the limitations of previous work, this study integrates viewpoint selection (VS), path planning, and trajectory optimization to generate paths that have kinematic constraints. In addition, selecting the required set of viewpoints is a challenging task that demands a stronger exploration capability from the algorithm. Optimizing the distribution of pheromone heuristics and cost heuristic parameters enhances the exploration range, while the improved pheromone update mechanism helps the algorithm avoid local optima and converge more quickly to the optimal path.

TABLE I
MAIN NOTATIONS OF THE MODEL

Notation	Remarks
N	Number of triangular meshes
S	The set of triangular meshes
V	The set of selected viewpoints
v	The flight velocity of UAV
q	The attitude of UAV
θ_h	The horizontal angle
θ_v	The vertical angle
$C(v_i)$	The set of triangular meshes covered by the i−th viewpoint
v_i	The position of the i−th viewpoint
P	Path length consumption
θ_i	The turning angle of the i−-th turn
ψ_i	The angle of rotation of the attitude of the i−-th rotation

Fig. 1. Comparison before and after geometry-aware model simplification, the left is the original high-resolution model and the right is the resulting lightweight model.

III. SYSTEM MODEL AND PROBLEM STATEMENT

This section first introduces the system model of a single UAV in the 3-D structure coverage scenario, and then presents an optimization problem aimed at minimizing the path length while minimizing the number of viewpoints. The notations used are described in Table I.

A. System Model

In this study, we assume that a UAV equipped with a fixed camera is used to perform a coverage inspection of the surface of a 3-D structure. To achieve a precise quantitative analysis of surface coverage, the 3-D surface is represented as a high-resolution mesh model, which is further simplified to retain key structural features, as illustrated in Fig. 1. The resulting lightweight model preserves structural fidelity through N triangular mesh elements, where each mesh element represents a potential coverage target in the set $S = \{s_1, s_2, \ldots, s_N\}$. The CPP task involves selecting a sufficient set of viewpoints $V \subseteq \{v_1, v_2, \ldots, v_N\}$ to ensure complete coverage of all mesh elements in S. To balance computational feasibility and physical constraints, each UAV is characterized by the set of parameters $U = \{v, q, \theta_h, \theta_v, d_{\max}\}$, where v and q represent the flight velocity and attitude of UAV U, expressed using quaternions. The parameters θ_h and θ_v denote the horizontal and vertical Field-of-View (FoV) angles of the camera, respectively, while $d_{\max}$ represents the maximum detection range, and the coordinates of the four corners of the far end of the quadrilateral pyramid can be expressed as

$$P_i = \begin{bmatrix} d_{\max} \\ s_y \cdot d_{\max} \cdot \tan\left(\frac{\theta_h}{2}\right) \\ s_z \cdot d_{\max} \cdot \tan\left(\frac{\theta_v}{2}\right) \end{bmatrix} \tag{1}$$

where $s_y, s_z \in \{-1, 1\}$ define the sign combinations for the four corners, allowing systematic representation of all edges of the quadrilateral pyramid. Since the UAV operates in a global 3-D space, the camera's FoV must be transformed from the local coordinate system to the world coordinate system based on the viewpoint v_i and orientation R. The orientation of the UAV is represented by its yaw, roll, and pitch, forming a combined rotation matrix

$$R = R_x(\text{roll}) R_y(\text{pitch}) R_z(\text{yaw}) \tag{2}$$

where R_x(roll), R_y(pitch), and R_z(yaw) are the standard rotation matrices around the respective axes. Applying this transformation, the four far-end points in the world coordinate system are given by

$$P_i' = R \cdot P_i + v_i. \tag{3}$$

B. Problem Statement

This section presents an optimization problem for minimizing the path length under coverage constraints, turning angle constraints, and attitude rotation constraints. To effectively solve this problem, it is decomposed into a set covering problem (SCP) and an asymmetric TSP (ATSP). The SCP focuses primarily on ensuring that the selected viewpoints can cover all mesh elements, while the ATSP optimizes the sequence of viewpoints to minimize the total flight path length, ensuring that all 3-D model surface meshes are covered while optimizing the UAV flight trajectory. For each viewpoint v_i, the set of mesh elements it can cover is denoted as $C(v_i) \subseteq S$. Consequently, the coverage constraint can be expressed as

$$\bigcup_{v_i \in V} C(v_i) \supseteq S \tag{4}$$

where $\bigcup_{v_i \in V} C(v_i)$ represents the total mesh elements covered by the selected viewpoints, S denotes the set of all mesh elements on the model surface, $\supseteq$ indicates that the coverage

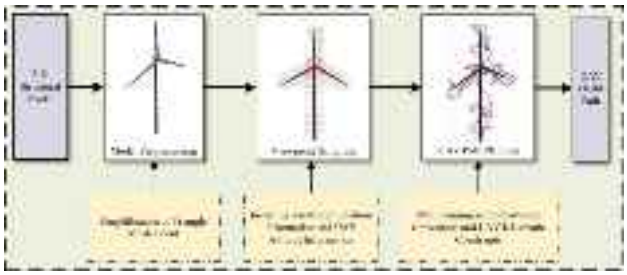

Fig. 2. Overall research scheme framework.

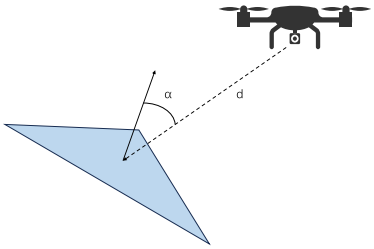

Fig. 3. Illustration of the range d and incidence angle α of a triangle facet with respect to the camera.

set must include or be equal to the entire set of mesh elements S.

The turning angle is incorporated as a constraint during the path search process, effectively limiting excessive sharp turns to ensure a kinematically feasible flight path [24]. The turning angle θ_i is calculated based on the vector angle between two consecutive path segments [25], [26], as shown in

$$\theta_i = \arccos\left(\frac{\overrightarrow{v_{i-1}v_i}\cdot\overrightarrow{v_iv_{i+1}}}{\left|\overrightarrow{v_{i-1}v_i}\right|\cdot\left|\overrightarrow{v_iv_{i+1}}\right|}\right) \tag{5}$$

where the vector $\overrightarrow{v_iv_{i+1}}$ represents the vector from the coordinates of the ith viewpoint to the coordinates of the $(i+1)$th viewpoint.

The angle of rotation of the attitude is determined by the orientation of the viewpoints, serving as a constraint that influences local path decisions and aims to optimize the local camera attitude transitions [27]. Rotation of attitude can be addressed by calculating the change in quaternion between two consecutive flight directions [28], [29], as shown in

$$(x, y, z, \omega) = \left(\upsilon_x \sin\frac{\alpha}{2}, \upsilon_y \sin\frac{\alpha}{2}, \upsilon_z \sin\frac{\alpha}{2}, \cos\frac{\alpha}{2}\right) \tag{6}$$

$$q_i = (x_i, y_i, z_i, \omega_i) = \left(\upsilon_i \sin\frac{\alpha_i}{2}, \cos\frac{\alpha_i}{2}\right) \tag{7}$$

$$\begin{aligned} q_{i+1} &= (x_{i+1}, y_{i+1}, z_{i+1}, \omega_{i+1}) \\ &= \left(\upsilon_{i+1} \sin\frac{\alpha_{i+1}}{2}, \cos\frac{\alpha_{i+1}}{2}\right) \end{aligned} \tag{8}$$

$$\begin{aligned} Q_i &= q_{i+1} \otimes q_i^{-1} = \left(\upsilon_i' \sin\frac{\psi_i}{2}, \cos\frac{\psi_i}{2}\right) \\ &= (u_i, \omega_i) \end{aligned} \tag{9}$$

$$\psi_i = 2\arccos(\omega_i) \tag{10}$$

where q_i represents ith the attitude of viewpoint along the path, x, y, and z denote spatial coordinates and ω represents the cosine of the half rotation angle, and $\upsilon = (\upsilon_x, \upsilon_y, \upsilon_z)$. (6)–(8) respectively define the specific attitudes of these two viewpoints. Equation (9) further defines the amount of attitude rotation represented by quaternion, and (10) calculates the rotation angle.

The path length is obtained by accumulating the Euclidean distances between adjacent interpolation points. According to the minimum snap optimization trajectory described in Section IV-B, the total path length P is calculated as shown in

$$P = \sum_{i=1}^{N-1}\sum_{j=1}^{M-1} d\left(\mathbf{p}_j, \mathbf{p}_{j+1}\right) \tag{11}$$

where i represents the ith path segment, and $\mathbf{p}_j$ represents the jth interpolated point in the path segment.

The objective function and constraints of the coverage mission can be formulated as

$$\begin{aligned} \min_{\mathbf{p}_1,\ldots,\mathbf{p}_j} \quad & P \\ \text{s.t.} \quad & 0 \le \theta_i \le \max \quad \{\min \quad \{\theta_{i1}, \ldots, \theta_{in}\}, \theta_{\max}\} \\ & i = 1, \ldots, N-2 \\ & 0 \le \psi_i \le \max \quad \{\min \quad \{\psi_{i1}, \ldots, \psi_{in}\}\psi_{\max}\} \\ & i = 1, \ldots, N-1 \\ & \bigcup_{v_i \in V} C(v_i) \supseteq S \end{aligned} \tag{12}$$

where $\{\theta_{i1}, \ldots, \theta_{in}\}$ and $\{\psi_{i1}, \ldots, \psi_{in}\}$, respectively, represent the set of rotational angles of the remaining candidate viewpoints and the set of attitude rotation angles and n represents the number of candidate viewpoints. $\theta_{\max}$ and $\psi_{\max}$ represent the desired maximum angle. If all candidate viewpoints exceed this range, the viewpoint closest to meeting this condition will be selected.

IV. Method

In order to better solve the problems raised in the above content, this article proposed the ACO-OPD, which designs the optimal CPP scheme for the UAV to ensure that the total path length is minimized, as shown in Fig. 2.

A. Viewpoint Sampling

We assume that the UAV is equipped with a camera fixed relative to the platform's orientation and possessing a limited field of view. For a triangular mesh on a 3-D surface to be considered visible by the camera, several conditions must be satisfied. First, all three vertices of the triangular mesh must lie within the camera's field of view. Second, the distance from the center of the triangular mesh to the camera must be within the effective range, indicated by $d \in [d_{\min}, d_{\max}]$. Third, the incident angle between the triangular mesh and the camera's Line of Sight (LoS) must not exceed the maximum effective angle, defined as $\alpha \le \alpha_{\max}$. Finally, the triangular mesh must not be occluded by other parts of the surface. Fig. 3 illustrates an example of a triangular mesh that lies within the camera's field of view. The viewpoint sampling under this constraint involves two steps: 1) generation and 2) selection. A meshed object has N surface elements and N viewpoints, where each viewpoint is defined by the position of the camera c_j and the

direction of the gaze g_j. The visibility matrix $V \in \mathbb{R}^{N\times N}$ shows which surfaces are visible from each viewpoint, as shown in (13). The aim is to ensure complete coverage of the target by generating viewpoints and lines of sight for high-precision detection. This process involves the generation of candidate viewpoints using the normal offset method [30], which creates viewpoints by applying an offset based on the surface gradient of a mesh. Specifically, the viewing direction is defined by the normal vector in the center of the mesh. The advantage of this normal viewing angle is that it is nearly perpendicular to the surface plane, thereby minimizing image distortion. And determining which surfaces are covered, leading to the creation of a visualization matrix [31]

$$V[i,j] = \begin{cases} 0, & \text{if element } s_j \text{ is not visible from } v_i \\ 1, & \text{if element } s_j \text{ is visible from } v_i \end{cases} \tag{13}$$

where the row $V[i,s]$ corresponds to viewpoint v_i, and similarly, the column $V[v,j]$ corresponds to the mesh's surface element s_j. Based on the visualization matrix, this study proposes a VS method. After a viewpoint is selected, the surfaces it covers are marked as no longer visible. In addition, the $V[i,j]$ of viewpoints within a radius equal to the average distance between viewpoints is optimized to guide subsequent VS. Once the entire row of a viewpoint in the $V[i,j]$ becomes zero, it is removed from the candidate set. This method effectively reduces the number of viewpoints dynamically throughout the planning process.

After completing the viewpoint sampling, obstacle avoidance is necessary during path planning. To achieve this, we first construct a 3-D Euclidean signed distance field (ESDF) based on the given surface mesh, which provides the distance from any viewpoint in space to the nearest obstacle, obtained through (14). During the viewpoint generation phase, we impose the constraint that the next viewpoint must maintain a LoS [25] connection with the current viewpoint. Collision detection is carried out by sampling positions along the line segment connecting the two viewpoints and checking their ESDF values for collisions with the mesh

$$g(p) = \begin{cases} 0, & \text{if ESDF}(p) > r \\ -1, & \text{otherwise} \end{cases} \tag{14}$$

where p represents the positions sampled from the polynomial trajectory, r is the radius of the UAV. Based on collision constraints, from the generated set of candidate viewpoints, we can determine which viewpoints can interconnect, forming a disconnected graph. After obtaining the viewpoint sequence, local optimization of the path is performed using the 2-optimal algorithm [32], and the rapidly-exploring random tree star (RRT*) algorithm [33] is used to find the shortest path connecting viewpoints that cannot be directly linked. Then, the path is optimized by minimizing snap to obtain an energy-efficient trajectory.

B. Trajectory Planning Based on Minimizing Snap

Inspired by [34], we observe that the lower velocity of the UAV limit significantly impacts energy consumption during forward flight, vertical ascent, and vertical descent. To optimize energy consumption, we control the lower limit of the average velocity. This article generates trajectories suitable for UAV motion using continuous multidegree differentiable polynomial-time functions under velocity constraints. When the trajectory function is differentiated with respect to time, the first derivative represents the instantaneous velocity of the UAV at that moment, while the second derivative represents the acceleration, specifically a polynomial of degree n

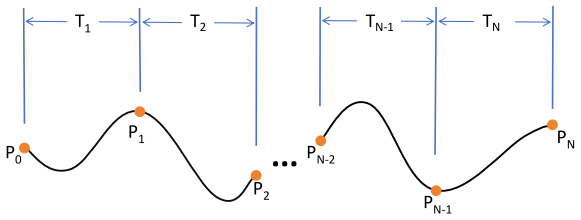

Fig. 4. Multisegment trajectory planning.

$$p(t) = p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n = \sum_{i=0}^{n} p_i t^i \tag{15}$$

where $p_0, p_1, \ldots, p_n$ are the trajectory parameters, for a total of $n+1$. Letting the parameter vector be $p = [p_0, p_1, \ldots, p_n]^T$, the trajectory can be rewritten in vector form as

$$p(t) = \left[1, t, t^2, \ldots, t^n\right] \cdot p. \tag{16}$$

The advantage of this representation is that it allows us to compute the position, velocity, acceleration, jerk, and snap of the trajectory at any given time, as shown in

$$\begin{aligned} \text{snap}(t) &= p^{(4)}(t) \\ &= \left[0, 0, 0, 0, 24, \ldots, \frac{n!}{(n-4)!} t^{n-4}\right] \cdot p. \end{aligned} \tag{17}$$

When there are many path points, a single polynomial may be too simple, and the complexity of the trajectory may make it difficult to represent with a single polynomial. Therefore, it is necessary to process it in segments, representing each segment of the trajectory as a polynomial curve. At the same time, certain constraints must be met, such as passing through specific intermediate points (position constraints) and continuity constraints at these points (which may include position, velocity, and acceleration). See Fig. 4 for an example.

The multisegment trajectory is represented as follows:

$$p(t) = \begin{cases} \left[1, t, t^2, \ldots, t^n\right] p_1 & t_0 \le t < t_1 \\ \left[1, t, t^2, \ldots, t^n\right] p_2 & t_1 \le t < t_2 \\ \vdots & \\ \left[1, t, t^2, \ldots, t^n\right] p_k & t_{k-1} \le t < t_k \end{cases} \tag{18}$$

where k denotes the segment index, and $p_i = [p_{i0}, \ldots, p_{in}]^T$ represents the parameter vector for each segment.

Before trajectory planning, the time allocation for each segment must also be determined, as time allocation is a key issue in trajectory planning that affects the quality of the planned trajectory. By allocating time for each segment, the velocity and acceleration of the UAV can be kept within a certain range, reducing the probability of energy loss due to excessive velocity.

In our approach, the time allocation is performed by first averaging the velocity across segments. Given the total planned path length, we assume a constant velocity within each segment, which allows for a proportional time allocation. Furthermore, each segment's time interval is normalized to [0, 1] by scaling it with the total time, ensuring consistency and computational efficiency in the trajectory planning process.

In trajectory planning, common optimization objectives include minimizing acceleration, snap, and jerk. Minimizing snap is more energy efficient, as snap, in physical terms, corresponds to the derivative of thrust. By minimizing the derivative of the thrust, a smoother thrust profile is achieved, which reduces mechanical resistance and thus conserves energy. Therefore, this article minimizes snap for trajectory optimization, further reducing energy consumption while maintaining the lower limit of the average velocity. The objective function to minimize when solving the trajectory is $f(p)$. The objective function, which incorporates (19), can be formulated as

$$\begin{aligned}\min_{p} \quad f(p) &= \min_{p} \quad \sum_{i=0}^{k-1} \int_{t_i}^{t_{i+1}} \left[p_i^{(4)}(t)\right]^2 dt \\ &= \min_{p} \quad \sum_{i=0}^{k-1} \int_{t_i}^{t_{i+1}} \left(\left[0, \ldots, \frac{n!}{(n-4)!} t^{n-4}\right] \cdot p_i\right)^T \\ &\quad \left[0, \ldots, \frac{n!}{(n-4)!} t^{n-4}\right] \cdot p_i \, dt \\ &= \min_{p} \quad \sum_{i=0}^{k-1} p_i^T Q_i p_i. \end{aligned} \tag{19}$$

The above expression illustrates the constraints at the trajectory's starting point; the constraints at the end point are similar. In addition, continuity constraints (in terms of position, velocity, and acceleration) are required

$$\left[\underbrace{0, \ldots, 0}_{(i-1)(n+1)}, 1, t_i, \ldots, t_i^n, -1, -t_i, \ldots, -t_i^n, \underbrace{0, \ldots, 0}_{(k-i-1)(n+1)}\right] p = 0. \tag{20}$$

If there are k segments on the path, there are a total of $4k + 2$ equality constraint conditions. These include the 6 constraints for position, velocity, and acceleration at the start and end points, $k - 1$ position constraints at intermediate points, and $3(k - 1)$ continuity constraints at intermediate points. Although this study chooses to minimize the snap for trajectory planning, the continuity of jerk is not required; this is determined by the optimality criterion.

It is evident that this optimization problem is a quadratic programming problem which can be solved directly using a quadratic programming solver to obtain p. Once p is determined, the trajectory polynomial can be derived. This article adopts the reference method [34] to quickly obtain the trajectory polynomial.

C. Improved Ant Colony Optimization Algorithm With Dynamic Parameter

Building on viewpoint sampling, we address the path planning problem by leveraging an improved ACO. First, we provide an overview of the basic ACO framework, followed by a detailed explanation of the proposed improvements. In ACO, ants exhibit foraging behavior through a positive feedback mechanism. Each ant deposits pheromones along its path, leading others. Transition probabilities are based on pheromone concentration and heuristic functions, allowing ants to generate a movement list and select movements according to state transition rules. The core of the algorithm focuses on these transition probabilities and pheromone updates [35]. Each ant starts at an initial viewpoint, uses roulette wheel selection to choose destination viewpoints, and constructs its path, eventually returning to the starting point. This process mimics natural ant behavior, where ants follow pheromone trails to find food sources. The transition probability $p_{ij}^k(t)$ for the k-th ant moving from viewpoint i to viewpoint j during the n-th iteration can be calculated as

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in \Omega_i^k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & \text{if } j \in \Omega_i^k \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

In the t-th iteration of the algorithm, each path from viewpoint i to viewpoint j is associated with a pheromone concentration $\tau_{ij}(t)$ and a value of the heuristic function $\eta_{ij}(t)$. Here, α represents the influence factor of the pheromones, while β is the influence factor of the heuristic function. The kth ant, when choosing the next viewpoint, makes a selection from its set of feasible viewpoints Ω_i^k. To ensure the shortest path is found, the heuristic information function $\eta_{ij}(t)$ is defined as the reciprocal of the Euclidean distance between viewpoint i and viewpoint j [36], as follows:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \tag{22}$$

where d_{ij} represents the straight-line distance between viewpoint i and viewpoint j and t represents the tth iteration.

The transition probabilities are calculated on the basis of the pheromone concentrations and heuristic functions. The total number of ants is set to n, and as they search for paths, they deposit pheromones on the paths. Over time, these pheromones evaporate, reflecting the natural fading of pheromone trails. After each iteration, the pheromone concentration on each path is updated according to the following formulas (23) and (24), reinforcing successful paths with higher pheromone levels to guide future ants while gradually neglecting less favorable routes

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \Delta \tau_{ij}(t) \tag{23}$$

$$\Delta \tau_{ij}(t) = \sum_{k=1}^{n} \Delta \tau_{ij}^k(t) \tag{24}$$

where ρ represents the pheromone evaporation coefficient, which ranges between 0 and 1. $\Delta \tau_{ij}(t)$ represents the total pheromone increment from viewpoint i to viewpoint j in the t

TABLE II
MAIN PARAMETERS IN ACO-OPD

Parameters	Remarks
N_{ant}	Population size
N_{iter}	Maximum number of iterations
α	Pheromone factor
β	Heuristic function factor
ρ	Pheromone evaporation coefficient
ν	Pheromone enhancement factor
τ_{ij}	Pheromone concentration on the path from i to j
η_{ij}	Heuristic function on the path from i to j
σ_{ij}	Increment of the coverage area on the path from i to j
ϕ_{ij}	The angle of the turn on the path from i to j
d_{ij}	The distance on the path from i to j

th iteration. Additionally, $\Delta\tau_i(t)$ is the amount of pheromone left on the path from viewpoint i to viewpoint j by the kth ant during the tth iteration, calculated as follows:

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{Q}{L_k}, & \text{if any } k \text{ goes from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

where Q is a constant representing the coefficient of pheromone increment, while L_k is the path length traveled by the kth ant.

To address the problem of traditional ACO algorithm becoming trapped in local optima, we propose a dynamic parameter improvement approach. This method introduces dynamic enhancement factors and evaporation coefficients within the ACO framework. Consequently, each group of ants inherits pheromones from the previous group while also utilizing distinct parameters, enhancing the diversity within each group. The objective is to achieve diversified results in as few iterations as possible. The main parameters of the ACO-OPD algorithm are presented in Table II.

When selecting the next viewpoint, the UAV prioritizes viewpoints with a larger coverage increment, shorter distance, and smaller turning angles. The search continues until a feasible path that meets the coverage constraints is discovered, which concludes the search mission

$$\eta_{ij} = \begin{cases} \frac{w_1}{d_{ij}} + w_2\sigma_{ij}, & \text{if } 1 \le i < 2 \\ \frac{w_1}{d_{ij}} + w_2\sigma_{ij} + w_3\phi_{ij}, & \text{if } i \ge 2 \end{cases} \tag{26}$$

where d_{ij}, σ_{ij}, and ϕ_{ij}, respectively, represent the distance between viewpoints, the coverage area increment, and the turning angle. The increment in coverage area is defined as the area of new triangular meshes that can be covered when selecting the next viewpoint, relative to the already covered complete triangular meshes. w_1, w_2, and w_3 denote the weights assigned to the three parameters, which are set to $w_1 : w_2 : w_3 = 0.8 : 0.1 : 0.1$, to ensure that the distance factor remains dominant.

In the pheromone increment model, this study introduces a variable pheromone enhancement factor ν, which increases the pheromone concentration during the tth iteration. Its role is to enhance the pheromone on a path if the optimal solution in the current iteration is better than the previous solutions. This accelerates the deposition of pheromones on the optimal path during the positive feedback process. The pheromone enhancement factor $\nu(t)$ is defined as

$$\nu(t) = \begin{cases} \frac{Q}{L^*(t)}, & \text{if } L^*(t) < L \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where $L^*(t)$ represents the length of the optimal path during the tth iteration, while L represents the length of the optimal path up to the previous iteration. Next, the new pheromone update rule will replace the existing one, and is specifically expressed as

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\big[\Delta\tau_{ij}(t) + \nu(t)\big]. \tag{28}$$

In the first iteration, $\nu(1)$ is set to zero. If the optimal path length in the second iteration is shorter than in the previous one, the pheromone on that path is enhanced; otherwise, $\nu(2)$ remains zero, indicating that no enhancement occurs. Thus, pheromone enhancement occurs only when a new optimal solution is found. Furthermore, we incorporate an elitist ant colony rule that retains pheromone only from the top 80% ranked ants.

The pheromone evaporation coefficient ρ significantly affects the convergence speed of the algorithm. A lower ρ results in slower evaporation of residual pheromones, which can hinder convergence. In contrast, a higher ρ accelerates convergence, but increases the risk of the algorithm becoming trapped in a local optimum. To address this issue, a variable pheromone evaporation coefficient is introduced.

To strike a balance between exploration and exploitation during the search process, we define a linearly decreasing pheromone evaporation coefficient to ensure that certain dynamic variables gradually decrease during early iterations while maintaining activity through a predefined lower bound. This approach helps prevent premature convergence and inefficiencies. The decay model enhances ACO algorithm by dynamically adjusting pheromone evaporation, enabling the continuous exploration of new potential paths, as shown in

$$\rho = \max\left(0.1, 0.4 \times \left(1 - \frac{t-1}{0.3 \times t_{\max}}\right)\right) \tag{29}$$

based on [37], ρ is varied between 0.1 and 0.4.

When optimal solutions do not show significant change over two consecutive iterations, it is suspected that the solution has become trapped at a local minimum. At this viewpoint, the variable pheromone evaporation coefficient ρ is used to escape the local optimum [8], with adaptive adjustments as follows:

$$\rho(t+1) = \begin{cases} 1 - 0.9(1-\rho(t)), & \text{if } \delta < 0.001 \\ \rho(t), & \text{otherwise} \end{cases} \tag{30}$$

$$\delta = \left|\frac{\text{cost}(t) - \text{cost}(t-1)}{\text{cost}(t-1)}\right| \tag{31}$$

where δ represents the relative error between the optimal solutions obtained in two consecutive iterations. When the change in the length of the optimal path between iterations is not significant (the relative error δ is less than 0.001), the pheromone evaporation coefficient ρ for the next iteration will be updated; otherwise, it remains unchanged.

To overcome the limitations of fixed heuristic parameters, this article proposes a Beta-distribution-based ant state

Algorithm 1: ACO-OPD Algorithm

Input: 3-D mesh model, visualization matrix, candidate viewpoint set
Output: Planned global path
Initialize parameters: N_{iter}, N_{ant}, ν, ρ
for $t = 1$ *to* N_{iter} **do**
  Put all ants on the starting viewpoint
  Generate α and β based on (32) and (33)
  for $k = 1$ *to* N_{ant} **do**
    while *Ant k has not reached the end* **do**
      Update heuristic information matrix based on (26)
      Select next viewpoint for ant k using the transition probability function (21)
      Update the coverage area of candidate viewpoints in the visualization matrix
    Optimize the path using the 2-opt algorithm
    Optimize the path using the RRT* algorithm
  if *All ants have reached the end* **then**
    Calculate the path length of the $k-$th ant optimized with the minimum snap algorithm: L_{k}
    Retain the pheromones of the top 80% of ants
    Find the best path for iteration t : $L^*(t)$
    Find the best path across all iterations: L
    Calculate pheromone enhancement factor based on (27)
    Update pheromone levels on the path using (28)-(31)
return Viewpoint sequence from the optimal path

transition mechanism. Specifically, the fixed cost heuristic parameter β in the traditional ACO algorithm is replaced with a value drawn from a Beta distribution biased toward $\beta_{\max}$, and the fixed pheromone heuristic parameter α is replaced with a value drawn from a mixed distribution influenced by β, which restricted between 0 and $\alpha_{\max}$. According to the maximum values of the parameter defined in [38], we set $\beta_{\max}$ to 5 and $\alpha_{\max}$ to 3 for our implementation

$$\beta = \beta_{\max} \times \text{rand}(N_{\text{ant}}, 1)^{0.6} \tag{32}$$

$$\alpha = \alpha_{\max} - \text{rand}(N_{\text{ant}}, 1) \times \min\left(\alpha_{\max}, \frac{\alpha_{\max}^2 + \beta - \alpha_{\max}}{\alpha_{\max}}\right). \tag{33}$$

In each iteration, the parameters for these ants are randomly generated to cover a wide range of the meaningful parameter space. By dynamically adjusting α and β, this strategy aims to evaluate the algorithm's performance under different influences of pheromone and cost heuristics. This nonuniform parameter generation enables ACO algorithm to escape local optima and enhances solution diversity. The rerandomization of parameters in each iteration allows for a broader exploration of the search space, which is particularly beneficial in complex problems, ultimately leading to more comprehensive solutions.

D. Time Complexity and Convergence Analysis

The time complexities of the LKH algorithm and ACO-OPD algorithm are influenced by different factors related to their operations, each of which has different advantages and disadvantages. For LKH algorithm, the complexity is approximately $O(I \cdot \lambda \cdot n^2)$, where I is the number of iterations required to converge, λ is the average size of the candidate set, and n is the total number of cities. LKH algorithm builds an initial tour and iteratively refines it using dynamic k-opt moves, evaluating only promising edge swaps based on candidate sets. Although this method avoids exhaustive search and provides a near-optimal solution for TSP, it is not suitable for problems with dynamic constraints where the set of viewpoints may change due to task requirements or environmental variations. The local search approach used by LKH algorithm lacks the flexibility to adapt to these changes. Specifically, LKH algorithm relies on a static sequence of viewpoints, which makes it incapable of handling situations that require dynamic selection or re-evaluation of viewpoints. In contrast, ACO-OPD algorithm, developed for UAV path planning, has a time complexity of $O(N_{\text{iter}} \cdot N_{\text{ant}} \cdot |V| + N_{\text{iter}} \cdot |R|)$, where N_{iter} is the number of iterations, N_{ant} is the number of ants, $|V|$ are the candidate viewpoints, and $|R|$ is the feasible routes. The complexity of ACO-OPD algorithm reflects its iterative path finding, pheromone updating, and dynamic coverage adjustments, which make it suitable for dynamic and complex environments, such as 3-D path planning. However, ACO-OPD algorithm is generally slower than LKH algorithm because of its nature as a probabilistic algorithm that requires multiple ants and iterations to explore feasible routes and converge on optimal or near-optimal paths. Performance is further influenced by parameter tuning, which can significantly affect the quality of solution and the convergence speed. Therefore, LKH algorithm is typically more efficient for static, smaller-scale problems, as it can achieve high performance in relatively fewer iterations. On the other hand, ACO-OPD algorithm offers greater adaptability to dynamic path planning mission in 3-D environments but at the cost of higher computational requirements, making it slower and more computationally intensive, especially as the number of viewpoints and routes increases. Using a parallel pool, we can potentially reduce the time complexity, thus lowering the computational cost to some extent. However, given that the proposed method significantly reduces the overall path length, the increase in computation time is considered acceptable.

To better understand the advantages of ACO-OPD algorithm, it is crucial to analyze its performance from the perspective of convergence, stability, and accuracy. These factors play a significant role in evaluating the algorithm's effectiveness in solving complex path-planning problems. Based on [8], the algorithm is analyzed as follows.

Lemma 1: For any route $r_{ij} \in R$, the pheromone concentration $\tau_{ij}(t)$ is bounded within the range $\tau_{\min} \le \tau_{ij}(t) \le \tau_{\max}$, where $\tau_{\max} = (f^*/\rho)$.

Proof: Assuming the initial pheromone concentration on route r_{ij} is $\tau_{ij}(0) = \tau_0$, we analyze the two possible cases over time. If the route r_{ij} is never chosen, the pheromone concentration decays exponentially due to evaporation, and at iteration t, it is given by $\tau_{ij}(t) = (1-\rho)^t \tau_0$. As $t \to \infty$, the pheromone approaches a minimum threshold $\tau_{\min}$, which ensures that the probability of selecting r_{ij} does not reach zero. However, if the route r_{ij} is chosen, the pheromone increases at each iteration by a maximum of f^*, and over time, the concentration approaches the upper bound $\tau_{\max} = [f^*/\rho]$. Combining these two cases, the concentration of pheromones $\tau_{ij}(t)$ is always bounded within $\tau_{\min} \le \tau_{ij}(t) \le \tau_{\max}$, which completes the proof. ■

Theorem 1: As the number of iterations t increases, the probability of finding the optimal solution $p^*(t)$ approaches 1. For any small value ε, $p^*(t) \geq 1 - \varepsilon$, and when $t \to \infty$, $p^*(t) \to 1$.

Proof: Even in the worst-case scenario, the minimum probability $p_{\min}$ of choosing the correct route is always greater than 0. Therefore, as iterations increase, the probability of finding the optimal solution increases, and eventually converges to 1. This section analyzes the convergence, stability, and accuracy of the ACO-OPD algorithm. ■

Let r_{ij} represent the route from point i to point j in the set of all feasible routes R. The deposition of pheromones in iteration t is denoted as $f(t)$. The optimal solution after t iterations is represented by S^*, and its pheromone deposition is denoted as f^*. Since the optimal path accumulates the most pheromone over time, we have the following relationship:

$$f(t) = \rho\left[\Delta\tau_{ij}(t) + \nu(t)\right] \leq f^* \tag{34}$$

where ρ is the pheromone decay rate, controlling the rate of pheromone decay, $\Delta\tau_{ij}(t)$ represents the pheromone deposited on route r_{ij} at iteration t, $\nu(t)$ is a random perturbation factor that introduces exploration and helps prevent premature convergence.

Consider the probability of selecting the optimal route in each iteration. Even in the worst-case scenario, the minimum probability $p_{\min}$ of choosing the correct route is always greater than 0, ensuring that the optimal solution remains accessible. As iterations increase, this probability accumulates due to the Markov property in the selection process, and the algorithm converges toward the optimal solution. Consequently, as $t \to \infty$, the probability of selecting the optimal solution satisfies

$$p^*(t) \geq 1 - \varepsilon \tag{35}$$

and therefore

$$\lim_{t\to\infty} p^*(t) = 1. \tag{36}$$

These properties ensure that the algorithm converges reliably to the global optimum as iterations progress. This proves the convergence of the ACO-OPD algorithm toward the optimal solution with an increasing probability as the number of iterations grows, ensuring accuracy and stability in the solution.

V. Simulation

In this section, we analyze the results of the CPP in 3-D structures by comparing them with other heuristic methods, demonstrating the feasibility and superiority of the proposed algorithm. The details are as follows.

A. Simulation Settings

To assess the performance of the proposed algorithm, simulations were performed using a standard wind turbine model. The UAV, equipped with a camera with FoV [120°, 120°] of view along both the horizontal and vertical axes, set the viewpoint distance at 8 m, as specified in [25]. For safety during inspections, the camera detection distance from the target structure is set between 5 and 20 m, ensuring safe detection of surface defects. All modules run on an Intel Core i7-12700H CPU. To validate the proposed framework, we benchmarked it through simulations in three large and complex scenarios, Wind Turbine ($75 \times 10 \times 120$ m^3), Big Ben ($20 \times 20 \times 110$ m^3), and Christ ($10 \times 36 \times 38$ m^3). The effectiveness of ACO-OPD algorithm is evaluated by comparisons with several methods. Among them, LKH+RRT* algorithm and the structural inspection planner (SIP) are CPP approaches that do not incorporate viewpoint quantity optimization. ACO-VP algorithm and ACO-IPP algorithm represent methods that integrate the VS component of our approach into CPP. VS+SO and VS+ESO algorithm [18], on the other hand, demonstrate the application of swarm optimization algorithms to planning the coverage path while incorporating the VS strategy of our method. To highlight the advantages of dynamic parameter adjustment, we integrated the viewpoint optimization process into ACO-IPP algorithm. This allows for a direct comparison under identical viewpoint optimization conditions. Each method was tested ten times, and the average results for the number of viewpoints and path lengths are summarized in Table III. The best-performing result from these tests is illustrated in Fig. 5. By incorporating this enhancement, we aimed to assess the performance of ACO-OPD algorithm within the same optimization framework and demonstrate the effectiveness of dynamic parameter adjustment in our approach. In the implementation of ACO-OPD algorithm, the population size is set to $N_{\text{ant}} = 10$ and following the experimental setup in [8], [39], and [40], the maximum number of iterations is $N_{\text{iter}} = 100$.

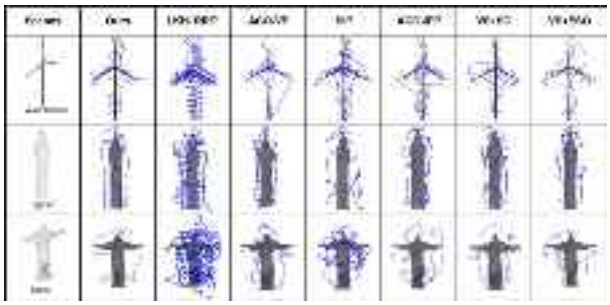

Fig. 5. Comparisons on generated path of the proposed method, LKH+RRT* [12], ACO-VP [8], SIP [14], ACO-IPP [9], VS+SO [17], and VS+ESO [18] in three scenes.

TABLE III
Coverage Planning Evaluation in Three Scenarios

Scenes	Method	Viewpoint Number	Path Length	Time
Wind Turbine	Ours	**45**	**735.65 m**	2570.97 s
	LKH+RRT*	1354	1717.51 m	**52.64 s**
	ACO-VP	48	780.25 m	3387.17 s
	SIP	1354	1611.90 m	702.88 s
	ACO-IPP	126	837.63 m	4499.15 s
Big Ben	Ours	**70**	**630.84 m**	311.75 s
	LKH+RRT*	528	1975.42 m	**37.76 s**
	ACO-VP	75	648.45 m	349.58 s
	SIP	528	784.35 m	148.37 s
	ACO-IPP	134	746.73 m	336.79 s
Christ	Ours	**80**	**337.63 m**	322.36 s
	LKH+RRT*	787	2055.34 m	**40.41 s**
	ACO-VP	84	421.46 m	362.54 s
	SIP	787	700.50 m	500.83 s
	ACO-IPP	205	1082.73 m	461.74 s

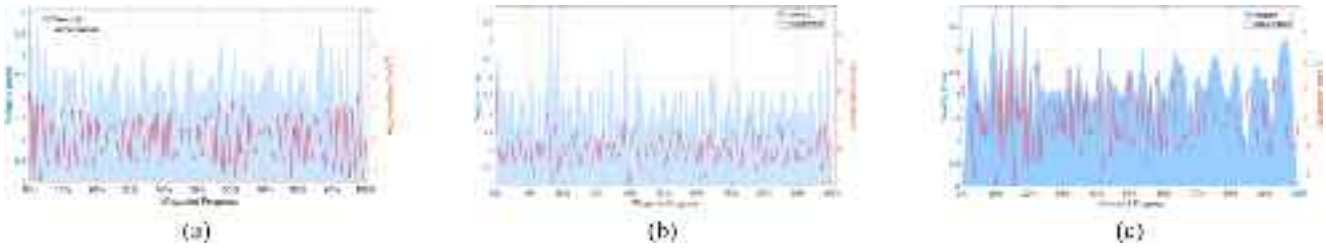

Fig. 6. (a)–(c) respectively, show the variations in trajectory velocity and acceleration when the lower limit of the average velocity for trajectory optimization is set to 2 m/s in the Wind Turbine, Big Ben, and Christ scenarios using our method.

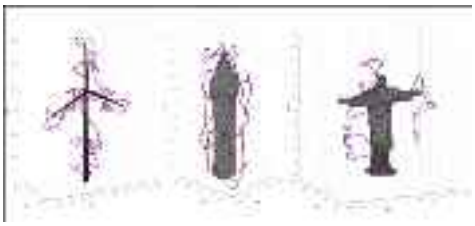

Fig. 7. Final trajectory optimization results for CPP, where the red line represents the initial path and the blue line represents the optimized trajectory.

TABLE IV
COMPARISON OF OURS AND THE OPTIMIZER METHOD WITH VS IN THREE SCENARIOS

Scenes	Method	Viewpoint Number	Path Length	Time
Wind Turbine	Ours	**45**	**735.65 m**	2570.97 s
	VS+SO	49	786.29 m	2835.58 s
	VS+ESO	**45**	756.72 m	2635.54 s
Big Ben	Ours	**70**	**630.84 m**	311.75 s
	VS+SO	72	653.71 m	346.43 s
	VS+ESO	72	641.48 m	339.78 s
Christ	Ours	**80**	**337.63 m**	322.36 s
	VS+SO	83	396.45 m	369.64 s
	VS+ESO	**80**	354.73 m	347.85 s

B. Performance Evaluation of Different Algorithms

Based on the results in Table III, for SIP algorithm and LKH+RRT* algorithm, the number of viewpoints corresponds directly to the triangular meshes in the 3-D model. This experiment achieves a significant reduction in the length of the path compared to other methods, highlighting the critical role of optimizing viewpoint numbers in addressing CPP challenges. Moreover, optimizing viewpoints also contributes to kinematically feasible paths.

Furthermore, the effectiveness of dynamic parameter adjustment extends beyond just viewpoint optimization; it also plays a significant role in improving the overall performance of the path planning process. In practical operations, the speed of flight of UAVs is a critical factor influencing energy consumption. By controlling the UAV to fly at an energy-optimal speed, the energy expenditure during inspection missions can be significantly reduced, making the selection of an appropriate flight speed essential. The ACO-VP algorithm and ACO-IPP algorithm algorithms do not incorporate dynamic optimization of the pheromone heuristic parameter α and the cost heuristic parameter β, leading to premature convergence and suboptimal path planning. In contrast, ACO-OPD algorithm dynamically adjusts these parameters, improving pheromone guidance and exploration, thereby achieving more efficient and optimized path planning. For the VS+SO algorithm and VS+ESO algorithm, due to their use of a group-based synchronous update mechanism, they cannot dynamically adjust the number of path points. By incorporating our VS strategy, these methods achieve viewpoint minimization before each iteration. However, as evidenced by the simulation results in Table IV, our proposed optimization method still demonstrates superior performance in 3-D CPP missions. This shows that with dynamic adjustment of heuristic parameters, pheromone evaporation rates and pheromone update rules, ACO-OPD algorithm can find better solutions by using pheromone information after each iteration. In particular, the pheromone provides more valuable guidance for VS, leading to more optimal outcomes.

As demonstrated in Fig. 6, trajectory optimization tends to increase the overall path length. Therefore, in our approach, incorporating the optimized path length as a cost during the path exploration process provides a more convincing comparison than simply applying trajectory optimization to the globally shortest path after path planning. According to the analysis in [41], there is an inverse relationship between the flight speed of the UAV and the energy consumption per unit distance. Within a certain range, higher speeds result in lower energy consumption per unit distance. However, trajectory planning often increases the path length, thereby increasing the risk of collisions. Therefore, optimizing speed requires a comprehensive consideration of the energy consumption model and the additional path length introduced by trajectory planning. Whether during horizontal flight or vertical ascent/descent, the energy consumption per unit distance is significantly higher when the flight speed is below 2 m/s. As shown by the acceleration and velocity variations in Fig. 7, the changes are smooth, demonstrating that the trajectory generated by the planning process is both feasible in terms of kinematics and efficient. As the velocity increases, the optimized path length decreases to some extent, as shown in Fig. 8. In minimum snap optimization, reducing the allocated time forces the trajectory to make sharper turns to meet constraints, and the nonlinearity of the problem can lead to irregular changes in path length rather than a smooth, steady decrease.

VI. CONCLUSION

This article proposes a UAV path planning method for 3-D structure coverage, aiming to improve coverage efficiency and reduce trajectory length. Building upon traditional methods, it introduces two key innovations to enhance path generation

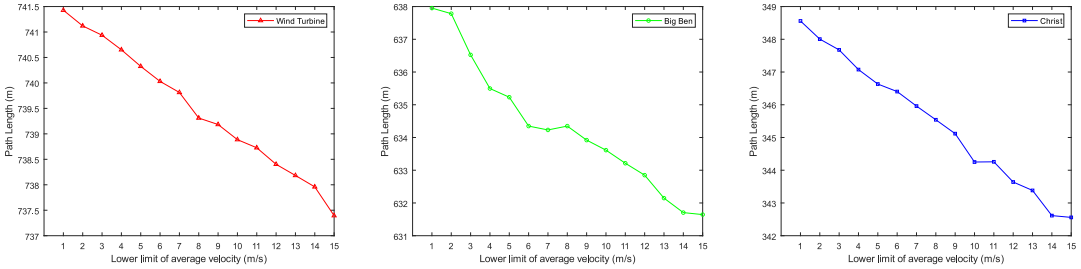

Fig. 8. Optimized path lengths obtained at different lower limits of average velocity based on the same sequence of viewpoints in the Wind Turbine, Big Ben, and Christ scenarios.

and overall UAV performance. First, existing approaches often overlook UAV flight constraints, leading to unrealistic planned paths. Moreover, uneven VS can cause frequent turns in confined spaces. To address this, an ACO-based method is developed, integrating a redesigned cost function, UAV kinematic constraints, and an improved search strategy to enhance path feasibility. Second, an improved ACO algorithm, termed ACO-OPD, is introduced with a Beta-distribution-based state transition mechanism and an adaptive pheromone regulation strategy. The proposed algorithm enhances global optimization by balancing exploration and exploitation and demonstrates strong adaptability in complex 3-D environments. Simulation results show that it outperforms traditional methods, particularly in challenging 3-D scenarios, and provides a practical and efficient solution for UAV inspection tasks over structures, such as wind turbines, buildings, and bridges. In future work, we plan to incorporate energy consumption considerations into the cost function, further optimizing both path length and energy efficiency to extend UAV operational endurance in real-world applications.

REFERENCES

[1] J. Seo, L. Duque, and J. Wacker, "Drone-enabled bridge inspection methodology and application," *Autom. Construct.*, vol. 94, pp. 112–126, Oct. 2018.

[2] J. Seo, L. Duque, and J. P. Wacker, "Field application of UAS-based bridge inspection," *Transp. Res. Rec.*, vol. 2672, no. 12, pp. 72–81, 2018. Online. Available: https://journals.sagepub.com/doi/abs/10.1177/0361198118780825

[3] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on BIM and UAV," *Autom. Construct.*, vol. 131, Nov. 2021, Art. no. 103881.

[4] X. Huang, Y. Liu, L. Huang, S. Stikbakke, and E. Onstein, "BIM-supported drone path planning for building exterior surface inspection," *Comput. Ind.*, vol. 153, Dec. 2023, Art. no. 104019.

[5] X. Li, B. Huang, B. Jia, Y. Gao, and J. Qiao, "MAS-DSO: Advancing direct sparse odometry with multi-attention saliency," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 17468–17481, Nov. 2024.

[6] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," *J. Commun.*, vol. 9, no. 9, pp. 687–692, 2014.

[7] W. Gu, D. Hu, L. Cheng, Y. Cao, A. Rizzo, and K. P. Valavanis, "Autonomous wind turbine inspection using a quadrotor," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2020, pp. 709–715.

[8] J. Li, Y. Xiong, and J. She, "UAV path planning for target coverage task in dynamic environment," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17734–17745, Oct. 2023.

[9] D. N. Bui, T. N. Duong, and M. D. Phung, "Ant colony optimization for cooperative inspection path planning using multiple unmanned aerial vehicles," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, 2024, pp. 675–680.

[10] Y. Chen, Z. Mou, B. Lin, T. Zhang, and F. Gao, "Complete coverage path planning for data collection with multiple UAVs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2024, pp. 1–6.

[11] G. Gao, Y. Mei, Y.-H. Jia, W. N. Browne, and B. Xin, "Adaptive coordination ant colony optimization for multipoint dynamic aggregation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7362–7376, Aug. 2022.

[12] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *Eur. J. Oper. Res.*, vol. 126, no. 1, pp. 106–130, 2000.

[13] C. Cao, J. Zhang, M. Travers, and H. Choset, "Hierarchical coverage path planning in complex 3d environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 3206–3212.

[14] A. Bircher et al., "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 6423–6430.

[15] C. Feng, H. Li, M. Zhang, X. Chen, B. Zhou, and S. Shen, "FC-planner: A skeleton-guided planning framework for fast aerial coverage of complex 3D scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024, pp. 8686–8692.

[16] M. Yan, C. A. Chan, A. F. Gygax, C. Li, A. Nirmalathas, and I. Chih-Lin, "Efficient generation of optimal uav trajectories with uncertain obstacle avoidance in MEC networks," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38380–38392, Dec. 2024.

[17] F. A. Hashim and A. G. Hussien, "Snake optimizer: A novel meta-heuristic optimization algorithm," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108320.

[18] L. Yao, P. Yuan, C.-Y. Tsai, T. Zhang, Y. Lu, and S. Ding, "ESO: An enhanced snake optimizer for real-world engineering problems," *Expert Syst. Appl.*, vol. 230, Nov. 2023, Art. no. 120594.

[19] W. Li, C. Wang, Y. Huang, and Y.-M. Cheung, "Heuristic smoothing ant colony optimization with differential information for the traveling salesman problem," *Appl. Soft Comput.*, vol. 133, Jan. 2023, Art. no. 109943.

[20] Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, and S. Wang, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," *J. Comput. Sci.*, vol. 25, pp. 50–57, Mar. 2018.

[21] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25546–25556, Dec. 2022.

[22] J. Chen, Y. Zhang, L. Wu, T. You, and X. Ning, "An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16842–16853, Sep. 2022.

[23] H. Gong, B. Huang, and B. Jia, "Energy-efficient 3-D UAV ground node accessing using the minimum number of UAVs," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12046–12060, Dec. 2024.

[24] C. González-Santos, M. A. Vega-Rodríguez, and C. J. Pérez, "Addressing topic modeling with a multi-objective optimization approach based on swarm intelligence," *Knowl.-Based Syst.*, vol. 225, Aug. 2021, Art. no. 107113.

[25] H. Zhu, J. J. Chung, N. R. Lawrance, R. Siegwart, and J. Alonso-Mora, "Online informative path planning for active information gathering of a 3D surface," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 1488–1494.

[26] M. Zhang, Y. Han, S. Chen, M. Liu, Z. He, and N. Pan, "A multi-strategy improved differential evolution algorithm for UAV 3D trajectory planning in complex mountainous environments," *Eng. Appl. Artif. Intell.*, vol. 125, Oct. 2023, Art. no. 106672.

[27] R. Xu, Y. Fan, Z. Li, S. Zhu, H. Shang, and A. Gao, "Time-optimal attitude planning for spacecraft with movable parts using second order cone programming," *Aerosp. Sci. Technol.*, vol. 141, Oct. 2023, Art. no. 108589.

[28] G. Legnani, I. Fassi, A. Tasora, and D. Fusai, "A practical algorithm for smooth interpolation between different angular positions," *Mech. Mach. Theory*, vol. 162, Aug. 2021, Art. no. 104341.
[29] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 9480–9487.
[30] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Three-dimensional UAS trajectory optimization for remote sensing in an irregular terrain environment," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2018, pp. 1101–1108.
[31] V. Staderini, T. Glück, P. Schneider, R. Mecca, and A. Kugi, "Surface sampling for optimal viewpoint generation," in *Proc. IEEE 13th Int. Conf. Pattern Recognit. Syst. (ICPRS)*, 2023, pp. 1–7.
[32] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-opt algorithm for the TSP," *Algorithmica*, vol. 68, no. 1, pp. 190–264, 2014.
[33] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1478–1483.
[34] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2520–2525.
[35] T. Wang, L. Wang, D. Li, J. Cai, and Y. Wang, "Monte carlo-based improved ant colony optimization for path planning of welding robot," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 7, 2023, Art. no. 101603.
[36] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, and X. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101005.
[37] L. Wu, X. You, and S. Liu, "Multi-ant colony algorithm based on cooperative game and dynamic path tracking," *Comput. Netw.*, vol. 237, Dec. 2023, Art. no. 110077.
[38] Y. Zhang, Y. Shen, Q. Wang, C. Song, N. Dai, and B. He, "A novel hybrid swarm intelligence algorithm for solving TSP and desired-path-based online obstacle avoidance strategy for AUV," *Robot. Auton. Syst.*, vol. 177, Jul. 2024, Art. no. 104678.
[39] C. Liu et al., "An improved heuristic mechanism ant colony optimization algorithm for solving path planning," *Knowl. Based Syst.*, vol. 271, Jul. 2023, Art. no. 110540.
[40] S. E. Comert and H. R. Yazgan, "A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106375.
[41] H. Gong, B. Huang, B. Jia, and H. Dai, "Modeling power consumptions for multirotor UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 7409–7422, Dec. 2023.

Jiaxin Du received the B.E. degree in automation from Huazhong Agricultural University, Wuhan, China, in 2020, and the master's degree in computer science from Inner Mongolia University, Hohhot, China, in 2025.

His research interest is UAV path planning.

Baoqi Huang (Senior Member, IEEE) received the B.E. degree in computer science from Inner Mongolia University (IMU), Hohhot, China, in 2002, the M.S. degree in computer science from Peking University, Beijing, China, in 2005, and the Ph.D. degree in information engineering from the Australian National University, Canberra, ACT, Australia, in 2012.

He is with the College of Computer Science, IMU, where he is currently a Professor. His research interests include indoor localization and navigation, wireless sensor networks, and mobile computing.

Dr. Huang was a recipient of the Chinese Government Award for Outstanding Chinese Students Abroad in 2011.

Bing Jia (Member, IEEE) received the Ph.D. degree from Jilin University, Changchun, China, in 2013.

She is with the College of Computer Science, Inner Mongolia University, Hohhot, China, where she is currently a Professor. Her current research interests include indoor localization, crowdsourcing, wireless sensor networks and mobile computing.