Engineering Applications of Artificial Intelligence 156 (2025) 111253

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

# Coverage path planning of bridge inspection with Unmanned aerial vehicle

Xudong Li [a], Yang Chen [a,*], Zhihuan Chen [a], Zixin Huang [b]

[a] School of Artificial Intelligence and Automation, Wuhan University of Science and Technology, Wuhan, 430081, China
[b] School of Electrical and Information Engineering, Wuhan University of Technology, Wuhan, 430205, China

ARTICLE INFO

*Keywords:*
Unmanned aerial vehicle
Coverage path planning
Particle swarm optimization
Bridge inspection

ABSTRACT

To address the low efficiency of bridge inspection using Unmanned Aerial Vehicle (UAV) due to redundant inspection viewpoints and frequent sharp maneuvers, this paper proposes a novel coverage path planning model that leverages artificial intelligence methods. Firstly, to minimize viewpoint redundancy, a Collaborative Greedy Neighborhood (CGN) algorithm is introduced to optimize the distribution of viewpoints. This method preprocesses the bridge model using Voronoi diagrams and generates optimal viewpoints by fully considering constraints such as camera field of view and gimbal angles. Secondly, in solving the UAV path planning problem, factors including collision risks between the UAV and the bridge, path length, frequency of steep ascents, and sharp turns are comprehensively considered. An improved Particle Swarm Optimization (PSO) algorithm based on a Greedy Strategy (Greedy Strategy-Improved PSO, GS-IPSO) is proposed to achieve global optimization of the path. The simulation results show that compared with the viewpoint offset method, Q-learning algorithm and clustering algorithm, the proposed CGN algorithm has significant advantages in significantly reducing the number of required viewpoints. In addition, in the path planning task, the GS-IPSO algorithm can not only effectively shorten the path length, but also significantly reduce the number of sharp turns and rapid climbs and descents compared to the traditional PSO algorithm and enhanced discrete particle swarm optimization (Enhanced-DPSO) algorithm, thereby significantly improving the inspection efficiency and reliability. Finally, through experimental verification in a real-world environment, the practicality and effectiveness of the method proposed in this paper are further confirmed.

## 1. Introduction

Since the beginning of the 21st century, countries around the world have made significant progress in infrastructure construction, with the number of bridges continuously increasing. However, over time, bridge materials may undergo aging and deterioration, leading to a series of issues such as structural weakening, corrosion, and cracking (Sobczyk et al., 2024). Therefore, regular inspection and maintenance are crucial to ensure the structural safety and proper functioning of bridges. Traditional bridge inspection methods not only consume a large amount of manpower and time but also pose significant safety risks when inspecting complex terrains or hard-to-reach areas (Faris et al., 2024).

In recent years, drone technology has matured significantly. Thanks to advantages such as low cost, high flight speed, and strong operational safety, drones have been widely applied across various industries (Niu et al., 2024). Depending on different scenario requirements, drones can be equipped with various types of sensors to perform a diverse range of tasks (Nwaogu et al., 2023). In bridge inspection missions, drones equipped with high-resolution cameras, LiDAR, and other sensors can capture high-definition images and videos, helping inspectors gain a comprehensive understanding of the bridge's condition (Yindee et al., 2024). Drones can conduct comprehensive inspections of large-area, complex-structured, or environmentally hazardous bridges in a short time, significantly reducing inspection time and labor costs (Zhang et al., 2022). Additionally, the use of drones eliminates the need for inspectors to directly contact bridge structures, especially in high-risk or hard-to-reach areas, thereby greatly reducing the risk of personnel injury.

When performing bridge inspection missions, drones typically encounter Coverage Path Planning (CPP) problems (Tan et al., 2021). CPP involves designing an optimal path that traverses every accessible part of the target area or minimizes area omissions and overlapping coverage, thereby ensuring comprehensive coverage of the entire region while satisfying the robot's motion constraints and avoiding environmental obstacles. In the context of bridge path planning, this can be achieved by generating viewpoints to ensure that each area is effectively

* Corresponding author. author.
*E-mail address:* chenyag@wust.edu.cn (Y. Chen).

https://doi.org/10.1016/j.engappai.2025.111253
Received 2 January 2025; Received in revised form 26 April 2025; Accepted 19 May 2025
Available online 26 May 2025

covered. These viewpoints are then used as path nodes to plan an optimal route that visits all nodes. This process can be transformed into the classic Traveling Salesman Problem (TSP) (Hoffman et al., 2013), which seeks to find an optimal closed path that allows the drone to start from a designated point, visit each node exactly once, return to the starting point, and minimize the total cost. The schematic diagram of the bridge inspection with drone is shown in Fig. 1.

To address the TSP, various intelligent optimization algorithms have been widely applied, including Ant Colony Optimization (ACO) (Zhang et al., 2020), Genetic Algorithm (GA) (Jiacheng and Lei, 2020), Particle Swarm Optimization (PSO) (Wang et al., 2020), and Simulated Annealing (SA) (Meliani et al., 2024). These algorithms are favored for their high versatility and computational efficiency, making them effective tools for solving combinatorial optimization problems such as the TSP. Particle swarm optimization has unique advantages over other algorithms. As a swarm intelligence optimization algorithm, PSO stands out for its fast convergence and simple implementation. Compared with the ant colony algorithm, PSO does not rely on building a probability model and can find a better solution faster; compared with the genetic algorithm, the parameter setting of PSO is simpler, and the number of fitness evaluations is relatively small, which helps to reduce the computational complexity. Therefore, PSO is particularly suitable for scenarios where a better solution needs to be obtained in a short time. However, traditional PSO may still face the problem of slow convergence and easy to fall into local optimality when dealing with complex TSP problems with a large number of access points.

Liu et al. (2021) proposed a universal large-scale bridge inspection coverage path planning model based on point cloud slicing. They utilized the split and combine–consensus based bundle algorithm (SC-CBBA) and clustering characteristic of cluster growing self-organizing map (CRSOM) algorithms to complete UAV task allocation and the planning of inspection path arc sequences. This study relies on bridge point cloud data to construct the model; however, it lacks sufficient consideration of the global aspects of the bridge's overall structure. Tu et al. (2024) used an improved RRT algorithm to complete the path planning task, optimizing the robot's trajectory in narrow areas. However, they did not consider the overall coverage requirements. Chen et al. (2023) optimized viewpoints using clustering methods and proposed a ranking-oriented hybrid ant colony algorithm to solve the coverage path planning model. This study only considered the range of the field of view and did not take into account the requirements for view clarity. Choi et al. (2018) introduced a UAV three-dimensional trajectory optimization algorithm for remote sensing tasks in irregular terrains. They proposed a normal offset viewpoint generation method based on different viewing angles.

Phung et al. (2017) modeled the detection path planning problem as an extended Traveling Salesman Problem (TSP) with coverage constraints and employed the Enhanced-DPSO algorithm to solve the TSP. However, this approach did not adequately consider the adverse effects of sharp turns and rapid ascents/descents on UAV energy consumption. Bolourian et al. (Bolourian and Hammad, 2020) developed a three-dimensional UAV bridge inspection path planning method combining photodetectors and LiDAR scanners. This method comprehensively considered the potential locations of bridge deck cracks and the shortest path planning time, using a genetic algorithm and A* algorithm for joint optimization. Nevertheless, the method primarily focused on bridge deck inspection, resulting in insufficient coverage of the pier areas.

Shanthakumar et al. (2020) proposed a UAV path planning method for scenarios where GPS and optical flow navigation fail. This method achieved navigation mode switching through a supervisor and utilized a planner to calculate the UAV's optimal inspection path to minimize flight time. While this study accounted for UAV positioning issues in extreme environments, the construction of the objective function only considered time consumption as the optimization target, neglecting other factors that affect path quality.

Fahmani et al. (Fahmani and Benhadou, 2024) developed a UAV-based line inspection functional architecture with electromagnetic interference and minimum approach distance as primary constraints. They proposed a hybrid optimization method combining genetic algorithms, particle swarm optimization, and ant colony algorithms. This study took into account the safety constraints of UAVs in practical scenarios.

Jung et al. (2018) presented a three-dimensional coverage path planning method for buildings by constructing viewpoints using the voxel method and determining the shortest path with a TSP solver. This research segmented buildings through multi-layer slicing and employed a voxel grid filter on each layer to reduce the number of viewpoints before sequentially connecting the paths of each layer. However, the UAV paths planned by this method require completing one layer before moving to the next, which may result in excessively long path lengths for buildings with particularly unique shapes. Zendehdel et al. (2024) constructed a UAV control method based on control through electrooculography (EOG) eye tracking. By converting electrooculogram signals into navigation commands, they achieved precise and efficient control of the UAV, opening up a new research direction in the field of human-computer interaction and auxiliary technology. At the same time, it provided a theoretical basis and technical ideas for using human eyes to control UAVs to perform bridge inspection tasks.

To address the aforementioned shortcomings in existing research, this paper first proposes a Collaborative Greedy Neighborhood algorithm based on Voronoi diagrams. This algorithm is utilized to preprocess the external surfaces of bridge models and generate uniformly scaled triangular meshes. By integrating the UAV camera's field of view,

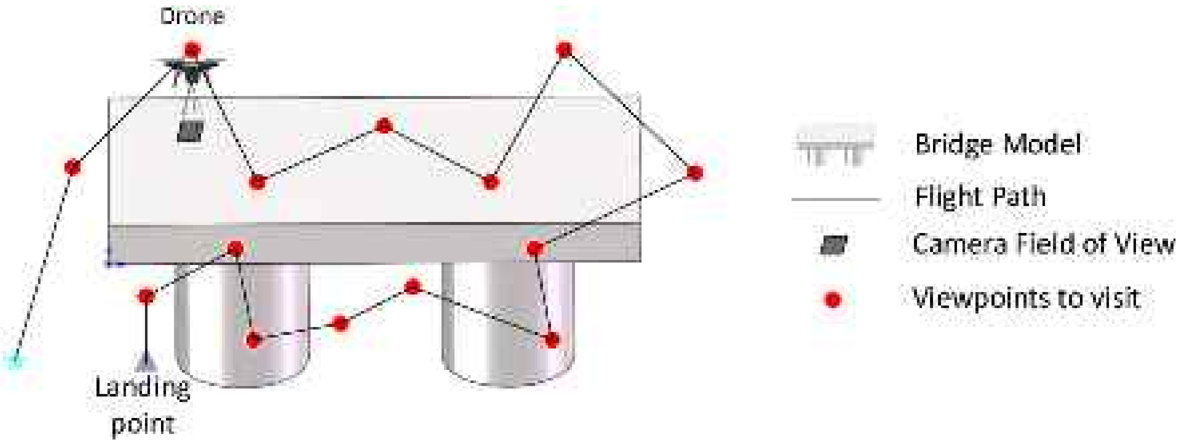


**Fig. 1.** Schematic diagram of bridge inspection with drone.

gimbal angle constraints, and coverage requirements, the CGN algorithm optimizes the number and distribution of viewpoints to ensure efficient coverage. Secondly, this paper constructs a mathematical model for UAV path planning, with path length, the number of sharp turns, and rapid altitude changes as optimization objectives. To solve this model, the Greedy Strategy-Improved Particle Swarm Optimization algorithm is proposed, thereby enhancing flight and inspection efficiency. The main contributions of this paper are as follows:

1) Proposed the CGN Algorithm to Reduce Viewpoint Redundancy. First, the external surfaces of the bridge model are preprocessed using Voronoi diagrams to generate uniformly scaled triangular meshes. Then, the number of triangles covered by each viewpoint is calculated, and the viewpoint covering the largest number of triangles is selected. Next, the central positions of the viewpoints corresponding to these triangles are recalculated, and the sets of viewpoints and triangles are updated. This process is repeated until all triangles are covered.
2) Developed a Mathematical Model for UAV Path Planning and Designed the GS-IPSO Algorithm to Solve It. Building upon traditional PSO, the Greedy Strategy-Improved Particle Swarm Optimization algorithm incorporates crossover and mutation operations from GA and integrates the Metropolis criterion from Simulated Annealing. Through the adaptive updating of factors, dynamic optimization of the particle swarm is achieved. Finally, elite particles are selected based on fitness values using the roulette wheel selection method for iterative crossover and mutation operations, generating new superior particles to replace those with lower fitness. This approach enhances both the efficiency of the solution process and the quality of the optimization outcomes.

## 2. UAV bridge inspection model

### 2.1. Bridge surface treatment

To effectively address subsequent viewpoint optimization issues, it is first necessary to reasonably preprocess the external surfaces of the obtained bridge models. This paper employs the Voronoi diagram method for processing the bridge's external surface, dividing the bridge surface into relatively uniform triangular meshes. This approach not only preserves the geometric features of the bridge but also enhances computational accuracy and mesh uniformity (Audette et al., 2011).

Define the set of triangles as $S = \{s_1, s_2, s_3, \ldots, s_n\}$, which comprises $n$ triangles. Define the vertices of the triangles as $\{A_n, B_n, C_n\}$,which $A_n = (x_{1n}, y_{1n}, z_{1n})$, $B_n = (x_{2n}, y_{2n}, z_{2n})$, $C_n = (x_{3n}, y_{3n}, z_{3n})$ Define the normal vector of the triangles as $\overrightarrow{N_n}$, As shown in Equation (1).

$$\overrightarrow{N_n} = \overrightarrow{A_nB_n} \times \overrightarrow{A_nC_n} \tag{1}$$

### 2.2. Viewpoint generation

To achieve optimal inspection results and minimize errors caused by image distortion or blurring, it is essential to ensure that the orientation of each viewpoint remains as perpendicular to the bridge surface as possible. Therefore, when the UAV arrives at a viewpoint location, the gimbal must adjust the camera to face directly toward the bridge surface, ensuring the highest quality imaging. An illustration of the camera angle is shown in Fig. 2.

To avoid the aforementioned potential image issues, the constraints of the gimbal are first determined, and the UAV inspection viewpoint position is defined as $(x_d, y_d, z_d)$, The central position of the target area corresponding to this viewpoint is $(x_t, y_t, z_t)$。 Based on this relationship, the gimbal's pitch angle (Pitch) and yaw angle (Yaw) during UAV inspection at the viewpoint are determined by Equation (2) and Equation (3), respectively.

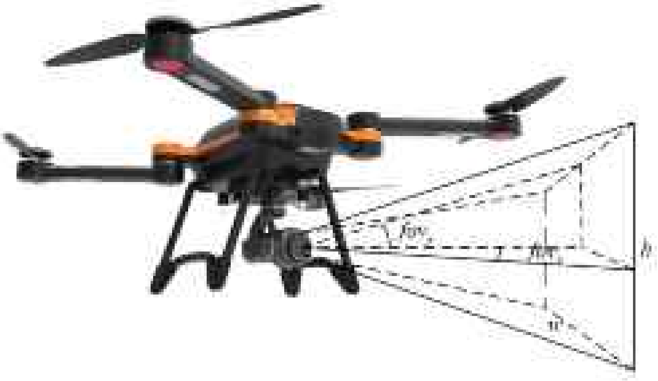


**Fig. 2.** Schematic diagram of camera angle.

$$Pitch = \arctan\left(\frac{z_t - z_d}{\sqrt{(x_t - x_d)^2 + (y_t - y_d)^2}}\right) \tag{2}$$

$$Yaw = \arctan\left(\frac{y_t - y_d}{x_t - x_d}\right) \tag{3}$$

Assume there are $N$ viewpoints. The sets of pitch and yaw angles for each viewpoint are defined as shown in Equation (4) and Equation (5), respectively. $(x_t^i, y_t^i, z_i^i)$ denotes the coordinates of the $i$ viewpoint, and $(x_d^i, y_d^i, z_d^i)$ represents the coordinates of the triangle center corresponding to the $i$ viewpoint.

$$\{Pitch_i\}_{i=1}^{N} = \left\{ \arctan\left(\frac{z_t^i - z_d^i}{\sqrt{(x_t^i - x_d^i)^2 + (y_t^i - y_d^i)^2}}\right) \right\} \tag{4}$$

$$\{Yaw_i\}_{i=1}^{N} = \left\{ \arctan\left(\frac{y_t^i - y_d^i}{x_t^i - x_d^i}\right) \right\} \tag{5}$$

Furthermore, considering that the UAV needs to maintain a certain safety distance from the bridge and the limitations of the camera's field of view, an appropriate distance must be maintained between the viewpoint and the bridge. This ensures that the images of the bridge surface captured during the inspection process have sufficient clarity and detail integrity. The definition of the relative distance between the viewpoint and the bridge surface is presented in Equation (6).

$$D = \frac{R_w \times G_{sd} \times f}{W} \tag{6}$$

$D$ is the relative distance between the viewpoint and the bridge, measured in meters (m). $R_w$ represents the camera's horizontal resolution, measured in pixels. $G_{sd}$ denotes the desired detection resolution, measured in meters per pixel (m/px). $f$ is the camera's focal length, measured in millimeters (mm). $W$ signifies the sensor width, measured in millimeters (mm).

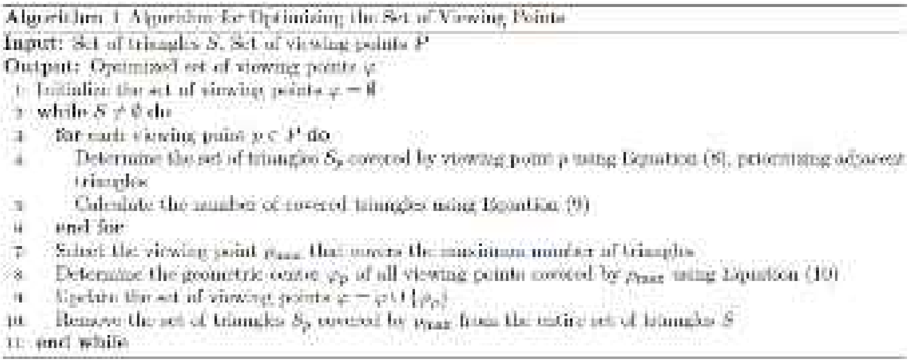
Considering the drone's imaging requirements and the geometric characteristics of the bridge surface, this study employs the previously defined triangle centroid $G_n$ in conjunction with the corresponding triangle normal vector $\overrightarrow{N_n}$ from Equation (1). After adjusting the direction of the normal vector, the centroid $G_n$ is displaced along the normal direction by a certain distance $D$ to generate the initial viewing point $P_n = \{p_1, p_2, \ldots, p_n\}$. The calculation of $P_n$ is presented in Equation (7).

$$P_n = G_n + D \cdot \overrightarrow{N_n} \tag{7}$$

### 2.3. Optimization of viewing points

In the inspection of bridge surfaces using drones, capturing images perpendicular to the bridge deck can achieve higher spatial resolution and reduced image distortion. However, this approach also presents several challenges. Specifically, the camera's field of view limits the area that can be covered in each individual capture. During the drone's flight, multiple photographs are required to cover the entire bridge surface. Nevertheless, the directly generated viewing points may not fully utilize the camera's field of view, potentially leading to excessive overlap in captured areas or the presence of uncovered blind spots.

To address the aforementioned issues, the field of view is defined as the rectangular area that the camera can cover at a specific altitude $z$. For a given viewing point and target plane, Equation (8) is used to determine whether the center point $(x, y)$ of a triangle lies within the current viewing point's field of view. A triangle is considered to be within the viewing point only if its center falls inside this rectangular area. Subsequently, the decision to retain the corresponding viewing point for the triangle is evaluated by calculating the total area of all triangles already included within the current viewing point's field of view using Equation (9). If the inclusion of the new triangle does not cause the total area to exceed the field of view area, the triangle is retained; otherwise, it is discarded. Finally, the geometric centers of all retained viewing points are calculated using Equation (10) to determine the optimal inspection viewing points, thereby maximizing coverage and enhancing inspection efficiency.



```
Algorithm 1 Algorithm for Optimizing the Set of Viewing Points
Input: Set of triangles $S$, Set of viewing points $P$
Output: Optimized set of viewing points $\varphi$
Initialize the set of viewing points $\varphi = \emptyset$
while $S \neq \emptyset$ do
    for each viewing point $p \in P$ do
        Determine the set of triangles $S_p$ covered by viewing point $p$ using Equation (8), prioritizing adjacent
        triangles
        Calculate the number of covered triangles using Equation (9)
    end for
    Select the viewing point $p_{max}$ that covers the maximum number of triangles
    Determine the geometric center $\varphi_p$ of all viewing points covered by $p_{max}$ using Equation (10)
    Update the set of viewing points $\varphi = \varphi \cup \{\varphi_p\}$
    Remove the set of triangles $S_p$ covered by $p_{max}$ from the entire set of triangles $S$
end while
```

$$\begin{cases} -\dfrac{z}{\tan\left(\dfrac{fov_x}{2}\right)} \leq x \leq \dfrac{z}{\tan\left(\dfrac{fov_x}{2}\right)} \\ -\dfrac{z}{\tan\left(\dfrac{fov_y}{2}\right)} \leq y \leq \dfrac{z}{\tan\left(\dfrac{fov_y}{2}\right)} \end{cases} \tag{8}$$

$fov_x$ denotes the camera's field of view angle in the $x$ direction, $fov_y$ denotes the camera's field of view angle in the $y$ direction, $\tan\left(\frac{fov_x}{2}\right)$ and $\tan\left(\frac{fov_y}{2}\right)$ are used to calculate the camera's field of view at a given altitude $z$.

$$\sum_n A \leq \frac{4z^2}{\tan\left(\frac{fov_y}{2}\right)\tan\left(\frac{fov_x}{2}\right)} \tag{9}$$

$\sum_n A$ denotes the total area of all triangles within the camera's field of view.

$$(x_c, y_c, z_c) = \frac{1}{\tau_c}\sum_{i=1}^{\tau_c}(x_i, y_i, z_i) \tag{10}$$

$\tau_c$ represents the total number of triangles contained within the viewing point.

To address the aforementioned issue of reducing the number of viewing points, it can be formulated as an optimization problem where $P$ represents the set of all viewing points and $S$ denotes the set of all triangles in the bridge model. This study aims to identify the minimal subset $\varphi$ of viewing points that ensures coverage of all triangles in the bridge model. The optimization objective is presented in Equation (11). To solve the optimal viewing point problem, this paper proposes a Collaborative Greedy Neighborhood algorithm, the detailed solution process of which is outlined in Algorithm 1.

$$\begin{aligned} &\min|\varphi| \\ &s.t. \forall s \in S, p \in P, s \prec p, \varphi = \{(x_c, y_c, z_c)\} \end{aligned} \tag{11}$$

### 2.4. Objective function

In the process of executing bridge inspection missions, the Unmanned Aerial Vehicle must determine the optimal sequence for visiting all visible inspection points and ultimately return to the starting location. This task can be effectively modeled as a Traveling Salesman Problem.

Let $V$ represent the set of all nodes and $E$ denote the set of edges connecting these nodes. The movement of the Unmanned Aerial Vehicle is constrained to the graph $G = (V, E)$. The objective of the Traveling Salesman Problem is to identify a path with the minimal total cost such that each viewpoint is visited exactly once and the UAV ultimately returns to the initial node.

Considering the safety concerns associated with UAV flight, it is essential to avoid collisions with inspection targets. To address this, a safety cost $L_{ij}$ is incorporated into the paths between adjacent viewpoints. Let $i$ and $j$ represent two nodes. The value of $L_{ij}$ is determined by checking whether the path between nodes $i$ and $j$ collides with the triangular mesh of the bridge surface. $L_{ij} = 1$ is defined such that if the path between nodes $i$ and $j$ does not collide with the bridge and remains within a safe range, the edge is assigned a distance cost $e_{ij}$ between the two points. Otherwise, the path is assigned an infinite cost. The path cost $e_{ij}$ is calculated as shown in Equation (12).

$$e_{ij} = \begin{cases} \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}, \text{if } L_{ij} = 1 \\ \infty, \text{otherwise} \end{cases} \tag{12}$$

During the execution of missions, frequent sharp turns by the Unmanned Aerial Vehicle not only increase energy consumption but also extend flight time. Therefore, it is crucial to minimize the number of sharp turns. To achieve this, the angle $\theta$ between three consecutive nodes is calculated to determine the occurrence of a sharp turn. When $\theta$ is less than $\frac{\pi}{2}$, it is classified as a sharp turn. The total number of sharp turns is incorporated as a component of the objective function to reduce their frequency. The number of turns $N_\theta$ is calculated as shown in Equation (13), where $T$ represents the total number of viewpoints.

$$\begin{cases} \theta = \arccos\left(\dfrac{(p_3 - p_2)\cdot(p_2 - p_1)}{\|p_3 - p_2\|\cdot\|p_2 - p_1\|}\right) \\ a_i = \begin{cases} 0, \text{if } i = 1, 2 \text{ } or \text{ } \theta \geq \dfrac{\pi}{2} \\ 1, \text{otherwise} \end{cases} \\ N_\theta = \sum_{i=1}^{T-2} a_i \end{cases} \tag{13}$$

To ensure the stability of the UAV during flight, it is essential to minimize abrupt vertical altitude changes. Such sudden changes can lead to increased energy consumption and reduced flight stability. Therefore, the number of rapid ascents and descents should be minimized. This is achieved by calculating the altitude difference between adjacent viewpoints. When the altitude difference exceeds 3 m, it is considered a rapid ascent or descent. The total number of these rapid altitude changes is incorporated as a component of the objective function to minimize such drastic vertical movements. Let $N_h$ denote the number of rapid ascents and descents, calculated as shown in Equation (14).

$$\begin{cases} h_i = \begin{cases} 0, \text{if } i = 1 \text{ or } |h_i - h_{i-1}| \leq 3 \\ 1, \text{otherwise} \end{cases} \\ N_h = \sum_{i=1}^{T-1} h_i \end{cases} \tag{14}$$

This paper proposes an objective function $J$, which accounts for the UAV's path length, the number of sharp turns, and the number of rapid ascents, each assigned with weight coefficients $\kappa_1 > 0, \kappa_2 > 0, \kappa_3 > 0$. These weight coefficients can be adjusted according to actual requirements. The objective function is defined as shown in Equation (15), where $E_{visited}$ represents the set of actual visited paths.

$$\min J = \kappa_1 \sum_{i=1} e_{i,i+1} + \kappa_2 \sum_{i=1}^{T-2} a_i + \kappa_3 \sum_{i=1}^{T-1} h_i \tag{15}$$
$$s.t. \ \forall i \in \mathrm{V}, \mathrm{e}_{i\ ,\ i+1} \in \mathrm{E}_{\text{visited}}$$

## 3. Path planning model solution

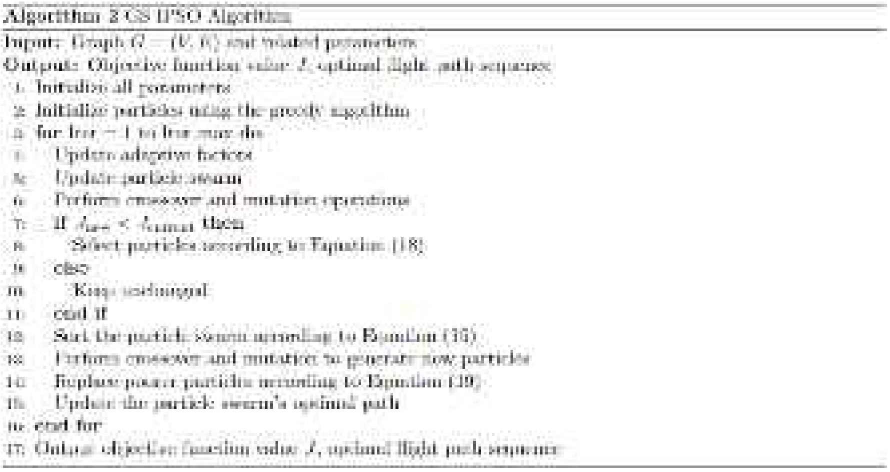
The Particle Swarm Optimization algorithm possesses unique advantages compared to other algorithms. As a swarm intelligence optimization algorithm, PSO stands out for its rapid convergence and simplicity in implementation. PSO is particularly well-suited for scenarios that demand obtaining high-quality solutions within a short timeframe. However, traditional PSO may still encounter challenges such as reduced convergence speed and a propensity to become trapped in local optima when addressing complex Traveling Salesman Problem instances that involve a large number of visit points.

To address the aforementioned issues, this paper proposes an improved Particle Swarm Optimization algorithm—Greedy Strategy-Improved PSO—for solving the bridge inspection path planning model.

Firstly, the algorithm initializes particles using a greedy strategy, thereby enhancing the quality of the initial solutions and accelerating the convergence speed of the algorithm. Subsequently, the traditional position and velocity update mechanisms of PSO are modified by incorporating crossover and mutation operations along with the Metropolis criterion (Meng et al., 2024). Additionally, an adaptive update factor is introduced to increase the diversity of initial solutions, effectively preventing the algorithm from prematurely converging to local optima.

Finally, based on the fitness values of the particles, a roulette wheel selection mechanism is employed to select high-performing particles for iterative crossover and mutation. This process generates new particles that replace low-fitness, inferior particles, thereby further enhancing the global convergence performance.

In summary, the solution process of the GS-IPSO algorithm is illustrated in Algorithm 2.



**Algorithm 2** GS-IPSO Algorithm

```
Input: Graph $G = (V, E)$ and related parameters
Output: Objective function value $J$, optimal flight path sequence
Initialize all parameters
Initialize particles using the greedy algorithm
for Iter = 1 to Iter_max do
    Update adaptive factors
    Update particle swarm
    Perform crossover and mutation operations
    if $d_{best} < d_{current}$ then
        Select particles according to Equation (18)
    else
        Keep unchanged
    end if
    Sort the particle swarm according to Equation (15)
    Perform crossover and mutation to generate new particles
    Replace poorer particles according to Equation (19)
    Update the particle swarm's optimal path
end for
Output objective function value $J$, optimal flight path sequence
```

### 3.1. Particle swarm initialization

In the GS-IPSO algorithm, to enhance the quality of the initial solutions, a greedy algorithm is employed for initializing the particle swarm. The specific process is as follows: First, a point $v \in V$ is randomly selected as the starting point of the path, where $V$ is the set of candidate points. Subsequently, in each iteration step $k$, the next point is chosen from the candidate set such that the cost $J(v_k)$ in Equation (15) is minimized. After each selection, the chosen point is removed from the candidate set until all points are incorporated into the path. Once all starting points have been utilized, a previously generated path is randomly selected as the starting point for a new path. Through this method, the initial path of each particle is close to the current local optimal solution, thereby effectively improving the quality of the initial solutions and accelerating the convergence process of the algorithm.

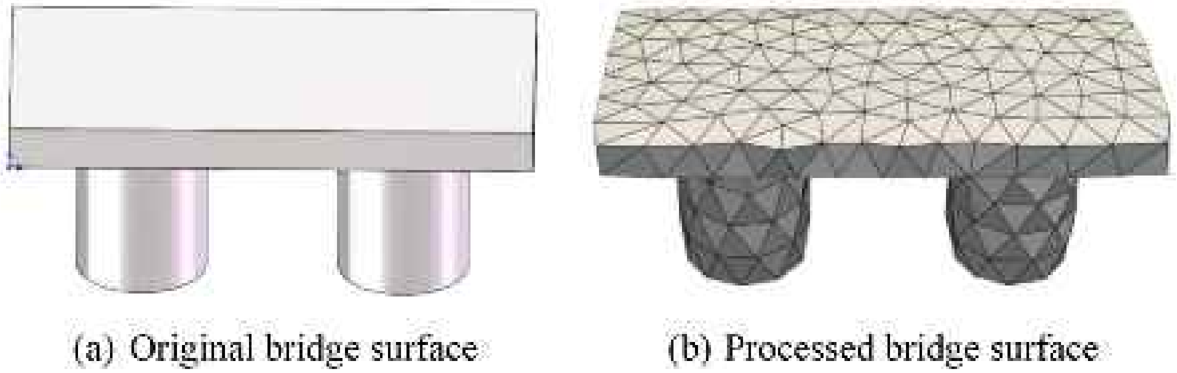

(a) Original bridge surface (b) Processed bridge surface

**Fig. 3.** Bridge meshing diagram.

**Table 1**
Camera parameters.

| Parameter Settings | Values |
|---|---|
| $f$ | 8 mm |
| $W$ | 7.4 mm |
| $R_w$ | 1920 px |
| $G_{sd}$ | 0.002 m/px |

**Table 2**
Comparison of the number of viewpoints.

| Method | Viewpoint Shift Method | Q-learning Algorithm | CGN Algorithm | K-means Clustering Algorithm |
|---|---|---|---|---|
| Number of Viewpoints | 596 | 232 | **178** | 198 |

### 3.2. Particle swarm update

In the classical Particle Swarm Optimization algorithm, the velocity and position update mechanisms of particles are not suitable for the Traveling Salesman Problem (TSP). Therefore, this paper adjusts the traditional update strategies by introducing crossover and mutation operations, while enhancing the performance of the particle swarm in complex high-dimensional path planning problems. The improved update rules are shown in Equation (16).

$$\begin{cases} V_k^{t+1} = w \otimes V_k^t \oplus r_1 \otimes \left(p_{\text{best}} \ominus X_k^t\right) \oplus r_2 \otimes \left(\alpha_1 \otimes \left(p_{\text{tmp}} \ominus X_k^t\right) \oplus \alpha_2 \otimes \left(g_{best} \ominus X_k^t\right)\right) \\ X_k^{t+1} = X_k^t \oplus V_k^{t+1} \end{cases} \tag{16}$$

$V_k^{t+1}$ and $X_k^{t+1}$ represent the velocity and position of the $k$ particle at the $t+1$ iteration, respectively. $w$ is the inertia weight, which controls the influence of the previous velocity on the current velocity. $r_1$,$r_2$ are random coefficients. $p_{\text{best}}$ denotes the particle's own historical best sequence, $p_{\text{tmp}}$ represents the particle sequence randomly selected at the current stage, and $g_{best}$ is the current global best sequence among all particles. $\ominus$ signifies the exchange and mutation operations between particles, $\oplus$ refers to the updates of velocity and position, and $\otimes$ indicates the multiplication of coefficients.

Furthermore, this paper introduces adaptive updating factors $\alpha_1$ and $\alpha_2$ to dynamically adjust the particle update mechanisms during different optimization stages. In the early stages of the algorithm, particles increase the diversity of solutions by exchanging path segments with other particles, thereby avoiding premature convergence to local optima. In the later stages, to accelerate the convergence speed, particles are more inclined to exchange with higher-performing particles to obtain superior solutions. The update rules for the adaptive factors are presented in Equation (17).

$$\begin{cases} \alpha_1 = 1 - \dfrac{t}{T_{early}}, \text{if } t \le T_{early} \\ \alpha_2 = 1 - \dfrac{t - T_{early}}{T_n - T_{early}}, \text{otherwise} \end{cases} \tag{17}$$

$t$ denotes the current iteration count, $T_{early}$ represents a threshold used to distinguish between the early and later stages of the particle swarm, and $T_n$ is the total number of iterations for the algorithm, which can be adjusted based on specific parameters.

Finally, to further enhance the diversity of particles, the algorithm proposed in this paper permits the retention of inferior solutions with a certain probability. To achieve this, the Metropolis criterion is introduced, and the replacement probability $P(J_{\text{current}} \rightarrow J_{\text{new}})$ is defined as shown in Equation (18).

$$P(J_{\text{current}} \rightarrow J_{\text{new}}) = \begin{cases} 1, \text{if } J_{\text{new}} \le J_{\text{current}} \\ \exp\left(\dfrac{J_{current} - J_{new}}{T_{temp}}\right), \text{otherwise} \end{cases} \tag{18}$$

$J_{\text{new}}$ and $J_{\text{current}}$ represent the objective function values of the new solution and the current solution, respectively. $T_{temp}$ is used to control the probability of the algorithm accepting inferior solutions, dynamically adjusting the algorithm's global search capability during different phases.

### 3.3. Particle replacement

During the optimization process, the fitness of each particle $k(k = 1, 2, \dots, n)$ is first evaluated and denoted as $J(k)$. Based on the fitness values, using $G_P$ as the threshold, the particles are divided into the "excellent particles" set $k_{best}$ and the "poor-performing particles" set $k_{poor}$. Excellent particles are given priority in being selected to participate in cyclic crossover and mutation operations.

In the cyclic crossover operation, two particles, $k$ and $l(k \ne l)$, are selected from set $k_{best}$. Two crossover points, $m_1$, $m_2(m_1 < m_2)$, are randomly chosen. The gene segments of the two particles within the interval $[m_1, m_2]$ are then exchanged, generating new offspring particles: $k_{new}$ and $l_{new}$. Subsequently, conflicts between the particles are checked.

The mutation operation is performed on the newly generated offspring particles $k_{new}$ and $l_{new}$. Two positions, $b_1$ and $b_2(b_1 < b_2)$, are randomly selected to define the mutation interval. The gene segment within this interval $[b_1, b_2]$ is then reversed. If the resulting individual exhibits improved fitness, the solution is accepted; otherwise, the mutation is discarded, and the algorithm proceeds to the next step.

Finally, the roulette wheel selection method is employed to select all particles. The probability $P_k$ of each particle $k$ being selected is defined as shown in Equation (19). Particles with higher fitness values have a greater probability of being selected for replacement, whereas particles

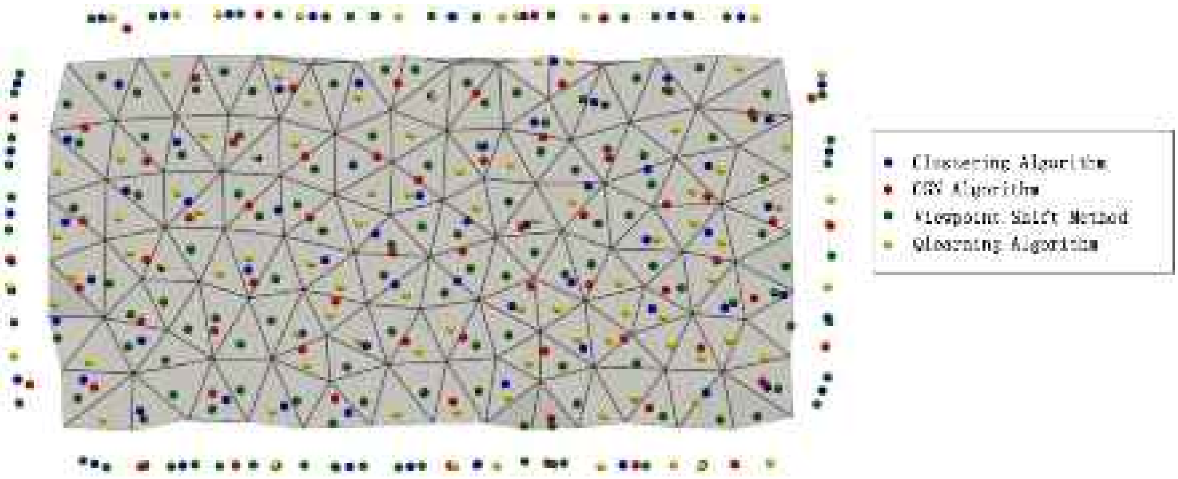


**Fig. 4.** Optimized viewpoint.

**Table 3**
Algorithm parameters.

| System Parameters | Value |
|---|---|
| *Iter_max* | 300 |
| *Particle_num* | 800 |
| *P_mutate* | 0.05 |
| *G_mutate* | 0.03 |
| $T_{temp}$ | $J(0)$ |
| $T_{early}$ | 0.6 × *Iter_max* |
| $G_P$ | 0.2 × *Particle_num* |
| $\kappa_1$ | 1 |
| $\kappa_2$ | 1 |
| $\kappa_3$ | 1 |
| $w$ | 1 |

with lower fitness values have a higher probability of advancing to the next generation.

$$P_k = \frac{J(k)}{\sum_{x=1}^{n} J(x)} \tag{19}$$

## 4. Experiment and analysis

### 4.1. Viewpoint generation

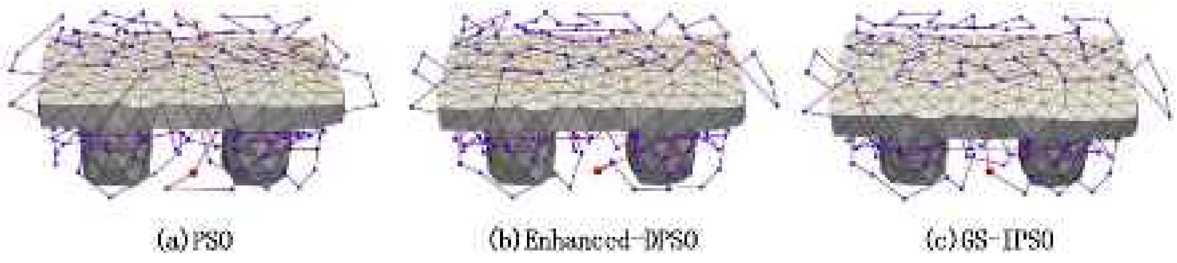
In this section, a bridge model with dimensions of 12 m in length, 6 m in width, and 5 m in height is selected, and bridge piers modeled as cylinders with a diameter of 3 m are used as simulation objects. After pre-processing the bridge surface, a total of 596 triangular meshes are generated. The meshing results of the bridge are illustrated in Fig. 3. The camera field of view parameters are set as shown in Table 1. Using Equation (2), the relative altitude $D$ between the UAV and the bridge is calculated to be approximately 2 m.

In order to verify the effectiveness of the CGN algorithm proposed in this paper, the algorithm is compared with the K-means clustering algorithm proposed in (Chen et al., 2023), the viewpoint offset method proposed in (Choi et al., 2018), and the reinforcement learning method proposed in (Beldi and Bessedik, 2023). The results in Table 2 show that when the coverage rate is 100 %, the number of viewpoints generated by the algorithm proposed in this paper is reduced by about 70 %, 10 %, and 23 %, respectively. In addition, as shown in the small figure on the right side of Fig. 4, the algorithm proposed in this paper has obvious advantages in viewpoint position layout.

### 4.2. Path planning simulation results and analysis

In order to evaluate the performance of the GS-IPSO algorithm, this paper compares the traditional PSO algorithm and the Enhanced-DPSO algorithm proposed in (Phung et al., 2017) with the algorithm in this paper. The algorithm-related parameter settings are shown in Table 3. Fig. 5 shows the path trajectory diagrams of the UAV inspection bridge model solved by the three algorithms. The purple color represents the UAV's flight path, the blue color represents the viewpoints the UAV needs to visit, and the red color marks the UAV's starting point.

As shown in Fig. 5, there is an obvious crossover phenomenon between the paths generated by the traditional PSO algorithm and the Enhanced-DPSO algorithm in the yellow circle area, while the path planned by the GS-IPSO algorithm appears to be more coherent and smoother. Specifically, the path generated by the traditional PSO algorithm has 39 sharp turns and 7 rapid ascents and descents, with a total path length of 333.664m; and after the Enhanced-DPSO algorithm is improved on this basis, the number of sharp turns is reduced to 27, the number of rapid ascents and descents is reduced to 5, and the total path length is shortened to 251.778m; in contrast, the GS-IPSO algorithm



(a) PSO (b) Enhanced-DPSO (c) GS-IPSO

**Fig. 5.** Comparison of the path trajectories.

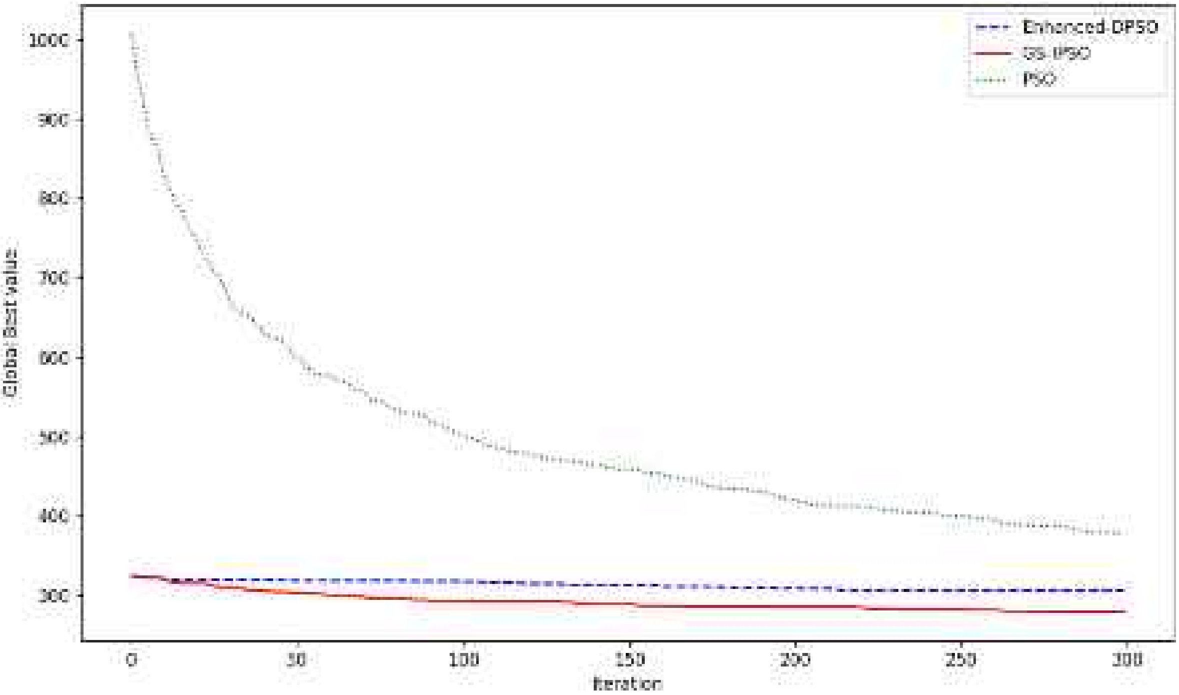


**Fig. 6.** Iteration curves of three algorithms solving the model.

**Table 4**
Algorithm performance comparison.

| Algorithm Name | $J$ | Runtime (s) |
|---|---|---|
| PSO | 374.242 | 243.274 |
| Enhanced-DPSO | 308.778 | 266.331 |
| GS-IPSO | 256.859 | 311.486 |

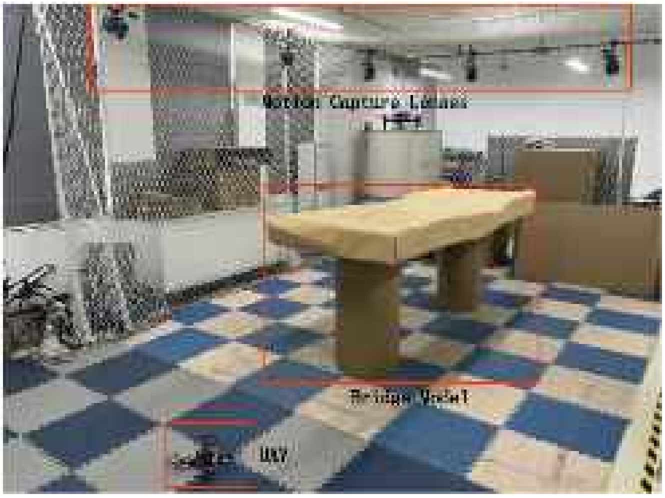


**Fig. 7.** Experimental environment.

further optimizes the path planning performance, with only 18 sharp turns, 3 rapid ascents and descents, and a total path length of 238.859m.

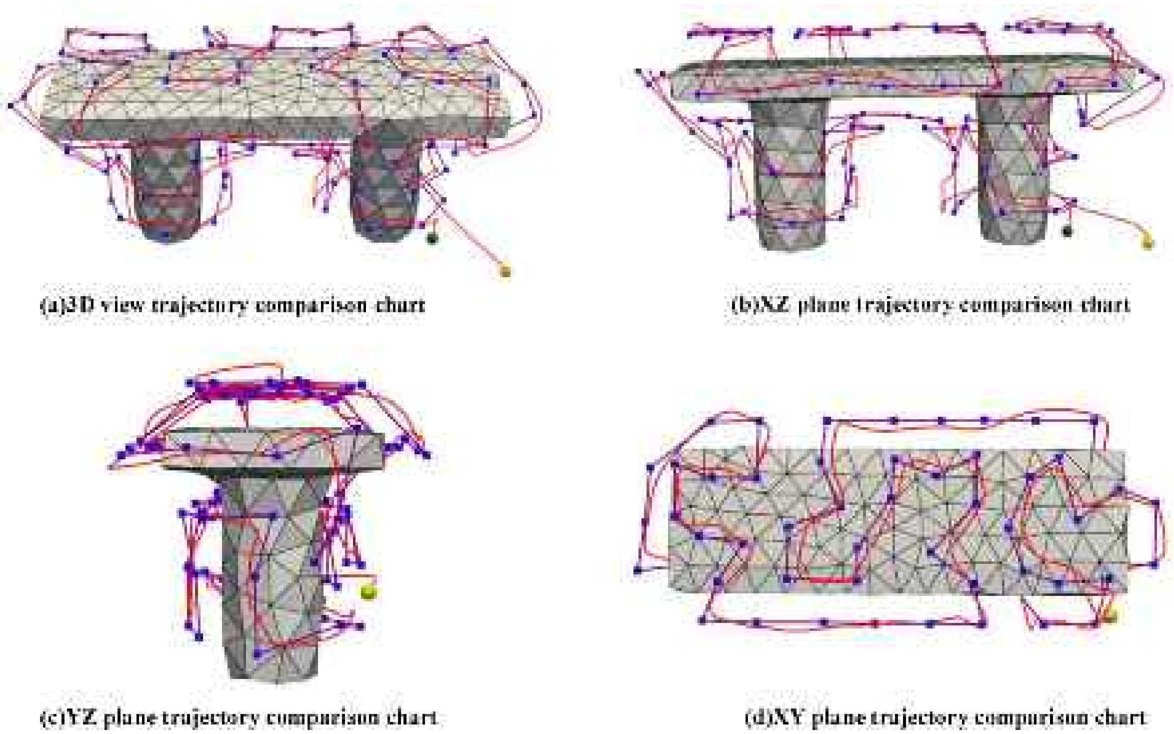
To further quantify the performance of the three algorithms, this study compared their total cost and running time during the solution process. To ensure the reliability of the results, each algorithm was run five times independently and the average value was calculated. Fig. 6 shows the comparison of the three algorithms during the total cost iteration process, while Table 4 lists the running time and total cost of each algorithm in detail. The experimental results show that the GS-IPSO algorithm has a more significant advantage in global search, can effectively avoid falling into the local optimal solution, and obtain better path planning results in a shorter running time.

### *4.3. Path planning experimental results and analysis*

To validate the effectiveness and reliability of the designed UAV path planning algorithm for bridge inspection in practical applications, this section presents an experiment conducted in a laboratory environment for the bridge coverage inspection task. The experimental setup is shown in Fig. 7. The primary goal of the experiment is to compare the actual UAV flight trajectory with the simulated trajectory. Due to space limitations, the simulated bridge model was scaled down, where the bridge deck is modeled as a rectangular structure with dimensions of 3 m in length, 1 m in width, and 0.2 m in height. The bridge piers are modeled as cylindrical structures with a diameter of 1 m.

In this experiment, the NOKOV optical 3D motion capture system was used to capture the UAV's position information and movement trajectory data. The UAV viewpoints generated in the simulation were imported into the system, and a start command was issued. The UAV took off from the starting point and performed an inspection flight around the bridge. Upon reaching each viewpoint, the UAV made a brief stop before continuing its flight until it reached the last viewpoint and completed the flight. After the UAV bridge inspection path experiment was completed, the flight data obtained in the experimental environment was imported into the simulation system for comparison and analysis with the simulation results.

In order to clearly compare the error between the actual trajectory and the simulated trajectory, this study compared the differences between the two trajectories from a three-dimensional perspective, XY plane projection, YZ plane projection, and XZ plane projection. Fig. 8 shows the comparison of the simulated trajectory and the actual trajectory of the drone, which are compared from a three-dimensional perspective, XY plane, YZ plane, and XZ plane. In the figure, the blue dots represent the points that need to be visited, the purple solid line represents the simulated flight trajectory, the red solid line represents the actual flight trajectory of the drone, the yellow dot is the take-off



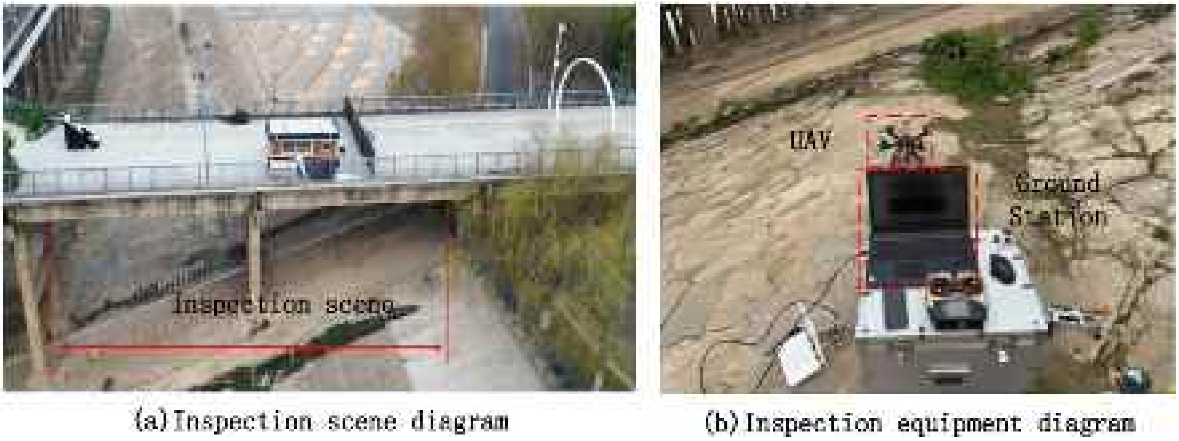
(a)3D view trajectory comparison chart

(b)XZ plane trajectory comparison chart

(c)YZ plane trajectory comparison chart

(d)XY plane trajectory comparison chart

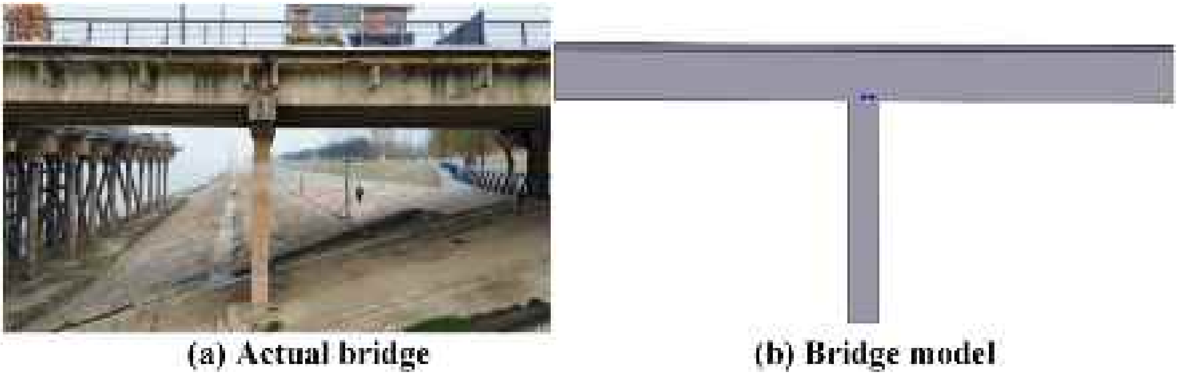
**Fig. 8.** Simulated trajectory and actual trajectory of drone.



(a)Inspection scene diagram

(b)Inspection equipment diagram

**Fig. 9.** Real world inspection scenario.



(a) Actual bridge

(b) Bridge model

**Fig. 10.** Schematic diagram of bridge model.

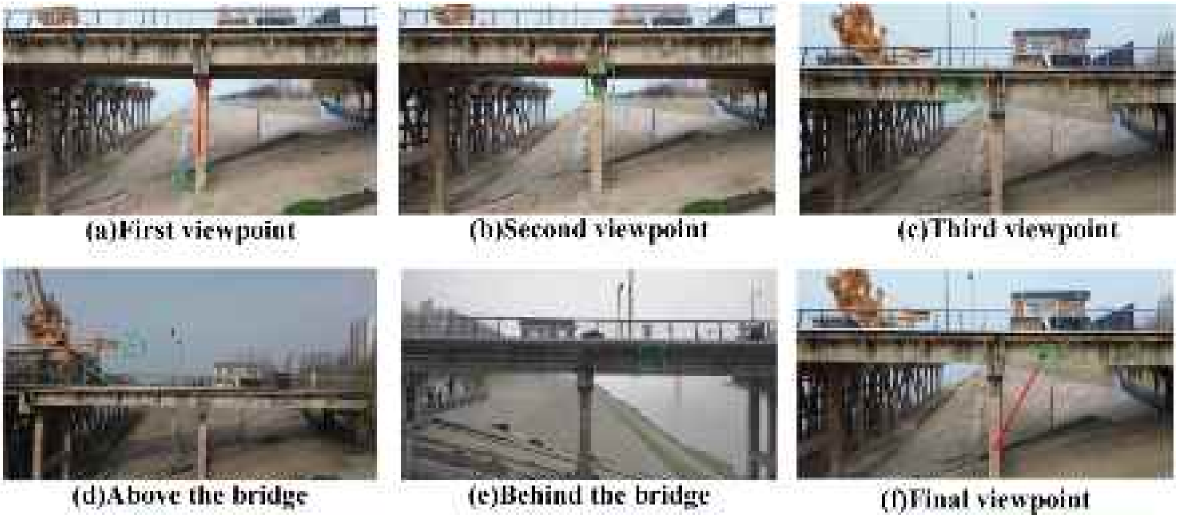


**Fig. 11.** Real-world bridge inspection.

point of the drone, and the green dot is the landing point of the drone. The results demonstrate that while a measurable discrepancy exists between the actual trajectory and the simulated trajectory - which could be attributed to unaccounted environmental interference factors and real-world measurement inaccuracies - the empirical data maintains satisfactory alignment with simulation predictions overall.

### 4.4. *Outdoor experimental verification*

In order to verify the effectiveness and reliability of the path planning algorithm proposed in this paper in the actual application environment, this study selected a part of a bridge over the Yangzi River in Wuhan as the experimental object. The bridge has typical structural characteristics and inspection requirements, which can fully reflect the performance of the system in the real bridge environment. The basic parameters of the experimental site are: bridge length 30 m, width 12 m, bridge deck thickness 1.6 m, bridge column height 8 m. The experimental scenario is shown in Fig. 9.

According to the actual parameters of the experimental site, a simplified model of the bridge was established using SolidWorks software, as shown in Fig. 10. On this basis, the reasonable generation of viewpoints was achieved based on the camera parameters and the corresponding mathematical models and calculation formulas. Considering the flight safety requirements and the limitations of hardware equipment, this study only conducted path planning verification tests on the side and top of the bridge, thus providing a theoretical and experimental basis for subsequent more complex three-dimensional path planning.

Fig. 11 shows the dynamic process of the UAV in the bridge inspection mission, showing the movement of the UAV in the viewpoint of the inspected bridge section. In the figure, the green box is used to accurately identify the real-time position of the UAV, while the red arrow clearly indicates its direction of movement.

## 5. Conclusions and future work

The CGN algorithm and GS-IPSO algorithm proposed in this paper show significant optimization effects in the path planning of UAVs for bridge inspection. The CGN algorithm optimizes the viewpoint distribution, effectively reduces the number of viewpoints and improves the coverage efficiency. Compared with the existing methods, the number of viewpoints is significantly reduced. The GS-IPSO algorithm shows better performance in path planning, reduces the number of sharp turns and sharp altitude changes, optimizes the path length, and improves the inspection efficiency and flight stability. The simulation results show that GS-IPSO has advantages over the PSO algorithm and the Enhanced-DPSO algorithm in both path optimization quality and calculation time. Finally, the feasibility and practicality of the algorithm proposed in this paper in the actual environment are verified through indoor and outdoor experiments. In summary, this research enhances the efficiency, reliability, and safety of UAV path planning for bridge inspection tasks. Additionally, it offers novel methodologies and theoretical insights for addressing UAV path planning challenges in complex environments.

Future research will focus on the optimization strategy of multiple UAVs cooperating to perform large bridge inspection tasks, in order to solve the limitation that a single UAV cannot achieve full coverage. Under the premise of knowing the parameters of the large bridge model, we intend to adopt a strategy based on task decomposition and collaborative scheduling to formalize the inspection task into a multi-traveling salesman problem. In order to comprehensively balance the inspection efficiency, energy consumption and task allocation balance, a multi-objective correlation algorithm is adopted, aiming to provide a scientific, reasonable and efficient scheduling solution for the collaborative operation of multiple UAVs through global optimization. At the same time, we also recognize that hardware limitations such as drone vibration, wind interference, and sensor noise may cause the actual shooting viewpoint to deviate from the theoretical optimal position and increase data noise, thus affecting the imaging quality and system robustness. To address these challenges, we plan to introduce robust optimization methods in future work, model hardware uncertainty as random perturbations, and combine real-time correction mechanisms to dynamically adjust the viewpoint generation strategy, thereby improving the overall performance of the system in complex environments.

## CRediT authorship contribution statement

**Xudong Li:** Writing – original draft. **Yang Chen:** Conceptualization. **Zhihuan Chen:** Writing – review & editing. **Zixin Huang:** Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by National Natural Science Foundation of

China under Grant 62173262, 62073250, 62203339, and Hubei Province Science and Technology Innovation Talent Program under Grant 2024DJC048.

## Data availability

Data will be made available on request.

## References

Audette, M., Rivière, D., Ewend, M., et al., 2011. Approach-guided controlled resolution brain meshing for fe-based interactive neurosurgery simulation[C]//Workshop on Mesh Processing in Medical Image Analysis. In: Conjunction with MICCAI 2011, pp. 176–186.

Beldi, Z., Bessedik, M., 2023. A Q learning approach for an efficient estimation of initial number of communities in social networks. Procedia Comput. Sci. 225, 3785–3794.

Bolourian, N., Hammad, A., 2020. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. Autom. ConStruct. 117, 103250.

Chen, L., Chen, Y., Yang, 2023. UAV coverage path planning for 3D structural visual inspection. J. Electron. Meas. Instrum. 37 (2), 1–10.

Choi, Y., Choi, Y., Briceno, S., et al., 2018. Three-dimensional UAS trajectory optimization for remote sensing in an irregular terrain environment[C]. 2018 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, pp. 1101–1108.

Fahmani, L., Benhadou, S., 2024. Optimizing 2D path planning for unmanned aerial vehicle inspection of electric transmission lines. Sci. Afr. 24, e02203.

Faris, N., Zayed, T., Fares, A., et al., 2024. Automated rebar recognition and corrosion assessment of concrete bridge decks using ground penetrating radar. Autom. ConStruct. 166, 105631.

Hoffman, K.L., Padberg, M., Rinaldi, G., 2013. Traveling salesman problem. Encyclopedia of operations research and management science 1, 1573–1578.

Jiacheng, L., Lei, L., 2020. A hybrid genetic algorithm based on information entropy and game theory. IEEE Access 8, 36602–36611.

Jung, S., Song, S., Youn, P., et al., 2018. Multi-layer Coverage Path Planner for Autonomous Structural Inspection of High-Rise structures[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1–9, 2018.

Liu, C., Zhang, H., Wang, C., 2021. UAV bridge laser scanning path planning based on coverage path arcs. Laser & Optoelectronics Progress 58 (8), 0828003.

Meliani, H., Ohamouddou, S., El Afia, H., et al., 2024. Robot path planning using fuzzy tsukamoto simulated annealing[C]. 2024 Mediterranean Smart Cities Conference (MSCC). IEEE, pp. 1–4.

Meng, F., Zhou, X., Piao, X., et al., 2024. Variable neighbourhood search based on Metropolis criterion for crowdsourced delivery scheduling problem in dispatch model. Comput. Oper. Res. 167, 106662.

Niu, B., Zhang, J., Xie, F., 2024. Drone logistics' resilient development: impacts of consumer choice, competition, and regulation. Transport. Res. Pol. Pract. 185, 104126.

Nwaogu, J.M., Yang, Y., Chan, A.P.C., et al., 2023. Application of drones in the architecture, engineering, and construction (AEC) industry. Autom. ConStruct. 150, 104827.

Phung, M.D., Quach, C.H., Dinh, T.H., et al., 2017. Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. Autom. ConStruct. 81, 25–33.

Shanthakumar, P., Yu, K., Singh, M., et al., 2020. View planning and navigation algorithms for autonomous bridge inspection with UAVs. Proceedings of the 2018 International Symposium on Experimental Robotics. Springer International Publishing, pp. 201–210.

Sobczyk, B., Pyrzowski, Ł., Miśkiewicz, M., 2024. Damage of a post-tensioned concrete bridge–Unwanted cracks of the girders. Eng. Fail. Anal. 160, 108178.

Tan, C.S., Mohd-Mokhtar, R., Arshad, M.R., 2021. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access 9, 119310–119342.

Tu, H., Deng, Y., Li, Q., et al., 2024. Improved RRT global path planning algorithm based on Bridge Test. Robot. Autonom. Syst. 171, 104570.

Wang, L., Liu, L., Qi, J., et al., 2020. Improved quantum particle swarm optimization algorithm for offline path planning in AUVs. IEEE Access 8, 143397–143411.

Yindee, M., Khumngoen, P., Manatchaiworakul, W., et al., 2024. Modifying a mini drone for remote drug delivery for wildlife medicine. Drone Systems and Applications 12 (1), 1–6.

Zendehdel, N., Chen, H., Song, Y.S., et al., 2024. Implementing Eye Movement Tracking for UAV Navigation[C]//International Symposium on Flexible Automation, 87882. American Society of Mechanical Engineers, V001T08A004.

Zhang, D., You, X., Liu, S., et al., 2020. Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning. IEEE Access 8, 129958–129974.

Zhang, C., Zou, Y., Wang, F., et al., 2022. Towards fully automated unmanned aerial vehicle-enabled bridge inspection: where are we at? Constr. Build. Mater. 347, 128543.