Biomimetic Intelligence and Robotics 5 (2025) 100228

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

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

Research Article

View planning for visual detection coverage tasks of large airplane upper surface using UAVs

Zhun Huang

Chinese Flight Test Establishment, Xi'an 710089, China

ARTICLE INFO

Article history:
Received 29 October 2024
Revised 13 February 2025
Accepted 17 February 2025
Available online 26 February 2025

Keywords:
Airplane surface inspection
View planning
Visual coverage
Combinatorial optimization
Unmanned aerial vehicle

ABSTRACT

In order to enhance the efficiency of visual inspection and effectively carry out 3D visual coverage tasks, this paper focuses on the 3D view planning problem concerning the visual coverage of an airplane's surface using unmanned aerial vehicles (UAV). Our objective is to attain a sufficiently high coverage rate with the least number of viewpoints. The contributions of this work are enumerated as follows. Firstly, the 3D model of the target aircraft is spatially extended in accordance with the depth range of the camera mounted on the drone, thereby confining the sampling range of 3D viewpoints. Next, a candidate set of viewpoints is generated through random sampling and the probabilistic potential field technique. Subsequently, we propose a novel hyper-heuristic algorithm. In this algorithm, a genetic algorithm serves as a high-level heuristic strategy, in tandem with multiple low-level heuristic operators devised for combinatorial optimization. This not only augments the capacity to seek the global optimal solution but also expedites the convergence rate, aiming to ascertain the optimal subset of viewpoints. Moreover, we devise a new fitness function for appraising candidate solution vectors in the set covering problem (SCP), strengthening the evolutionary guidance for genetic algorithms. Eventually, experimental findings on the simulated and real airplanes corroborate the efficacy of the proposed method, i.e., it markedly diminishes the requisite number of viewpoints and augments inspection efficiency.

1. Introduction

With the continuous advancement of aviation technology, the inspection of the surface of airplanes has become a critical task for ensuring their safe operation [1]. To improve the efficiency and precision of monitoring the surface condition of airplanes, automation technology is gradually being incorporated and is emerging as a prominent research direction within the domain of aviation safety [2,3]. This research focuses on the view planning problem for visual coverage tasks on the upper surface of large airplanes. It aims to generate a set of coordinates and line-of-sight directions in three-dimensional space for Unmanned Aerial Vehicles (UAV), ensuring comprehensive visual coverage of the airplane's upper surface with the minimum number of viewpoints possible, thereby enhancing inspection efficiency.

In view planning applications for inspection and surveillance, common optimization objectives usually encompass minimizing the number of viewpoints and maximizing the coverage rate of the surface area [4–6]. Scott et al. [7] propose the offset method within the "generate-test" framework. Moreover, their study converts the view planning problem into a set cover problem and solves the optimal subset of viewpoints by utilizing optimization algorithms. Nevertheless, the number of viewpoints generated by this method is relatively large, which results in increased computational costs and potentially lower coverage rates for large target objects. Choi et al. [8] propose an offset method, which incorporates normal vector offset and vertical offset techniques based on viewing angles. However, similar to previous methods, it suffers from high computational costs and is not suitable for inspecting large structures. Jing et al. [4] introduce a novel two-step computation method known as the iterative random sampling method. This method generates candidate viewpoints through random sampling in the surrounding space of the target object and solves the set cover problem using a greedy search algorithm. Compared with the offset method, it enhances the coverage rate of the target object's surface and reduces computational costs. Zheng et al. [9] propose a clustering-based view planning method, which generates viewpoints by clustering the triangular mesh of the target object's surface. This method exhibits high coverage efficiency but necessitates addressing the issue of viewpoint positioning correction. Wu et al. [10] propose a continuously differentiable sampling-based multi-directional view planning method, which takes the pitch and yaw angles of the gimbal as constraints. By constructing a continuously differentiable view quality objective function for the viewpoint collection, they optimize the candidate viewpoint set, thereby reducing the number of viewpoints.

E-mail address: hzCFTE@163.com.

https://doi.org/10.1016/j.birob.2025.100228

Vanessa et al. [11] propose a novel method for automatically generating viewpoints, which selects viewpoints by solving an integer linear programming problem and omits the preprocessing of the target. Wang et al. [12] proposed a parallel deep reinforcement learning method for view point sampling that utilizes Fibonacci numerical integration within a sphere. This method is capable of overcoming the local non-uniformity present in traditional methods. However, it does not adapt well to the surfaces of irregularly shaped objects. Kong et al. [13] proposed a genetic algorithm for view point selection, which is based on the alternating evolution strategy and aims at optimizing both the number of view points and the quality of views. Unlike the particle swarm optimization algorithm, which is prone to local optima, the genetic algorithm, an evolutionary algorithm based on the mechanism of natural selection and inheritance, may yield a better effect in solving the set covering problem [14]. Meta-heuristics are frequently employed to enhance search and optimization algorithms. This method is characterized by universality and flexibility and can be applied to multiple fields, including decision-making and logistics. Meta-heuristics consist of two parts: high-level heuristics and low-level heuristics. High-level heuristics select and apply different low-level heuristics based on the problem characteristics. Therefore, it is more efficient compared with traditional heuristics [15]. Zhang et al. [16] proposed a Q-learning-based meta-heuristic evolutionary algorithm for combinatorial optimization problems and achieved favorable results. Zhang et al. [17] proposed a meta-heuristic algorithm based on genetic programming and embedded the simulated annealing algorithm within the low-level heuristics to enhance the local search ability. The effectiveness of this algorithm was demonstrated through experiments on the benchmark set.

Due to the complexity of airplane structures and the critical nature of their applications, this paper enhances the algorithm for view planning based on the "generate-test" framework [18–20] by building upon the random sampling method, which is a model-based approach, i.e., a rough geometric model of the target object is available before planning. This method involves a two-step computational process: first, generating a large set of candidate viewpoints, and then transforming the problem into a combinatorial optimization problem to select the optimal subset of viewpoints through combinatorial optimization techniques. The proposed work comprises three main components: the generation of candidate viewpoints, visibility evaluation, and combinatorial optimization. First, the constraint space for candidate viewpoints generation is established based on the 3D model of the target airplane and camera parameters. Candidate viewpoints and directions are generated by utilizing random sampling and potential field methods. Then, the coverage of the candidate viewpoints is calculated based on the geometric model of the target object, and the problem is transformed into a combinatorial optimization problem. Finally, a genetic algorithm-based hyper-heuristic algorithm is proposed to conduct the combinatorial optimization and solve the optimal subset of viewpoints. The main contributions of this research are as follows:

- We propose an enhanced random sampling method for generating candidate viewpoints and innovatively put forward a genetic-based hyper-heuristic algorithm to conduct combinatorial optimization, aiming to identify the optimal subset of viewpoints that fulfill the coverage requirements.
- We introduce a novel fitness function to evaluate the candidate solutions for the set covering problem (SCP), with the aim of enhancing the evolutionary guidance of the combinatorial optimization algorithm.

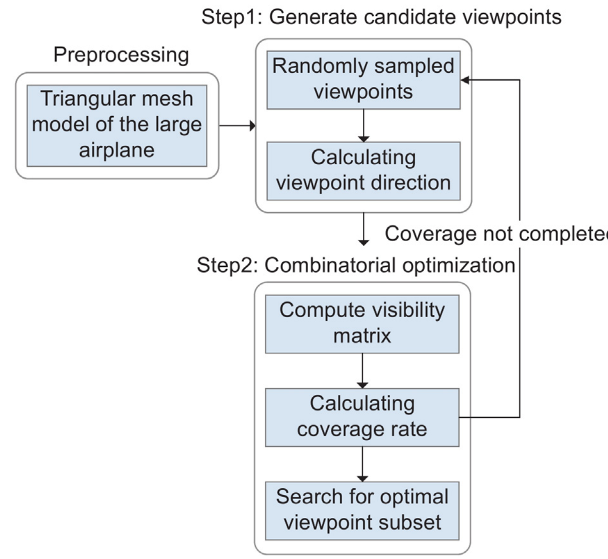


Fig. 1. Research framework of our method.

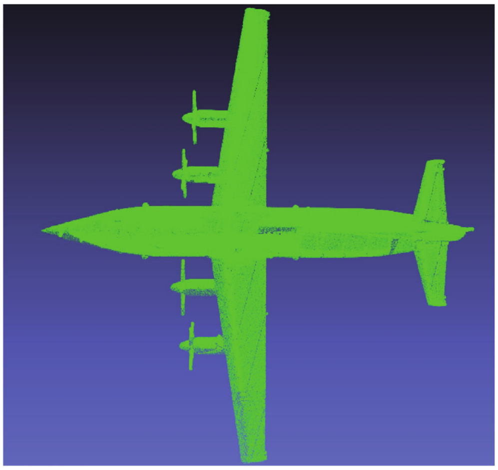
2. Method

This paper presents an enhanced iterative random sampling method [4] to address the view planning problem by employing a two-step computational framework. In the first step, candidate viewpoints are generated through a random sampling method. In the second step, the visibility matrix is computed based on the candidate viewpoints, thereby converting the problem into a combinatorial optimization problem and searching for the optimal subset of viewpoints, as depicted in Fig. 1. An iterative process ensures that if the candidate viewpoints generated by the random sampling algorithm are insufficient to achieve visual coverage of the airplane surface, new candidate viewpoints are continuously generated.

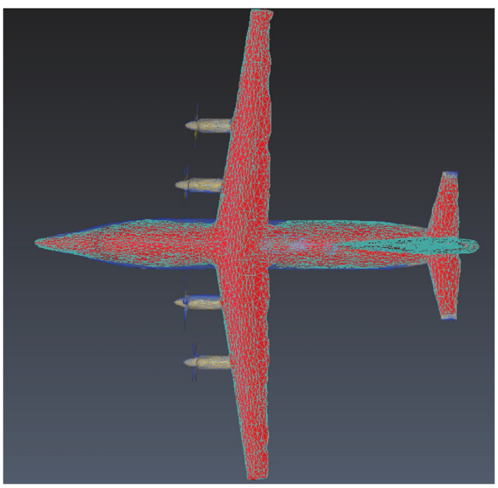
To generate high-quality viewpoints and enhance coverage efficiency, we define the constraint space for viewpoint sampling within the random sampling method based on the 3D model of the target airplane and camera parameters. Based on the genetic algorithm, a novel hyper-heuristic optimization algorithm is proposed to solve the set cover problem. Four types of heuristic operators are designed based on neighborhood structures to strengthen the crossover and mutation stages of the genetic algorithm, thereby enabling the discovery of higher-quality viewpoint subsets and achieving full coverage with fewer viewpoints.

2.1. 3D model preprocessing and candidate viewpoints generation

To generate the viewpoints for the airplane, the 3D mesh model derived from the point cloud is required. Therefore, the preprocessing of the 3D point clouds of the airplane is essential, which mainly comprises three steps: (1) Point cloud data acquisition, which can be accomplished using specialized equipment, for instance, a 3D laser scanner. (2) Processing the point cloud data through filtering and noise reduction, and performing uniform sampling to reduce the data volume. (3) Triangulating the point cloud model of the airplane to divide its surface into several nearly identical triangles. As illustrated in Fig. 2, the resulting triangular mesh model after processing serves as the input for the algorithm, where the red section represents the surface of the airplane that requires visual coverage in the viewpoint planning task.

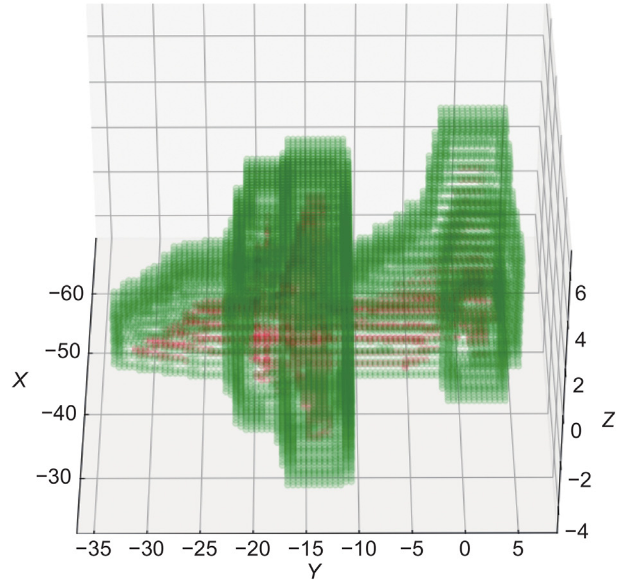


(a) Original point cloud model

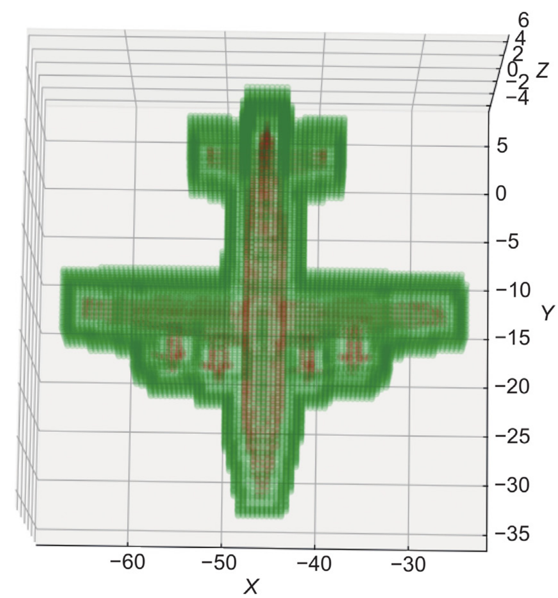


(b) Triangular mesh model

Fig. 2. Preprocessing for a 3D airplane model to generate a triangular mesh model based on the original point cloud.



(a) Side view of voxel dilation



(b) Top view of voxel dilation

Fig. 3. A schematic diagram using voxel expansion.

Furthermore, generating candidate viewpoints involves defining the constraint space for viewpoint sampling, where the camera's effective visual depth lies between the minimum and maximum visual depths. In the random sampling method, the sampling space is determined by binary voxel expansion in three-dimensional space [21], which is defined as follows:

$$A \oplus B = \{a + b | a \in A, b \in B\} \tag{1}$$

where A is the target airplane model in this paper, and B is the expansion operation. Since the visibility of the camera is constrained by the Field of Detection (FOD), two expansions are applied to the target object: first, the maximum FOD of the camera, followed by its minimum FOD. The random sampling method performs two voxel expansions on the target object, utilizing the maximum and minimum depth of the camera as the expansion distances. The difference between the two expansions defines the sampling space, where candidate viewpoints are randomly generated.

In this paper, we enhance this step of the random sampling method. Based on computational comparisons, we discover that when the position of the generated viewpoint is approximately 0.85 times the maximum depth from the target object, the overall quality of the candidate viewpoints is higher, thereby leading to more efficient visual coverage. When the viewpoint is positioned farther from the target object, the coverage effect generally deteriorates. Therefore, we modify the expansion distances to 0.85 times the maximum depth and the minimum depth, utilizing the difference between the two expansions as the sampling space. As shown in Fig. 3, the red section represents the voxelized airplane model, while the green section represents the result of voxel expansion.

Candidate viewpoints are randomly generated within the constraint space. For each batch of randomly generated candidate viewpoints, it is essential to calculate an appropriate line-of-sight direction for each viewpoint position, which determines the Unmanned Aerial Vehicle's (UAV) position and the camera's shooting direction when performing the coverage task. In this paper, the line-of-sight direction $\boldsymbol{V}$ is calculated using the probabilistic

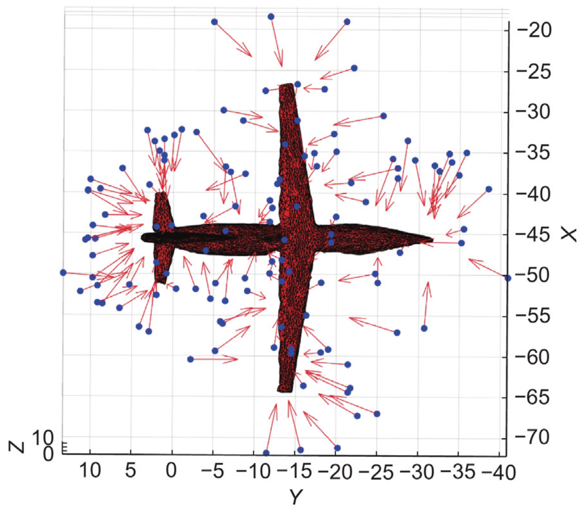


Fig. 4. Generated candidate viewpoints.

potential field method [4] as follows:

$$\boldsymbol{V} = \frac{\sum_{i=1}^{N} \frac{\boldsymbol{p}_{\mathrm{vp}} - \boldsymbol{p}_{\mathrm{mesh}_i}}{\left\| \boldsymbol{p}_{\mathrm{vp}} - \boldsymbol{p}_{\mathrm{mesh}_i} \right\|^3}}{\left\| \sum_{i=1}^{N} \frac{\boldsymbol{p}_{\mathrm{vp}} - \boldsymbol{p}_{\mathrm{mesh}_i}}{\left\| \boldsymbol{p}_{\mathrm{vp}} - \boldsymbol{p}_{\mathrm{mesh}_i} \right\|^3} \right\|} \tag{2}$$

$$\text{for all} : \left\{ \boldsymbol{p}_{\mathrm{mesh}_i} \mid \left\| \boldsymbol{p}_{\mathrm{vp}} - \boldsymbol{p}_{\mathrm{mesh}_i} \right\| < FOD_{\max} \right\}$$

where $\boldsymbol{p}_{\mathrm{vp}}$ is the direction vector for the candidate viewpoint, $\boldsymbol{p}_{\mathrm{mesh}_i}$ is the position of the ith triangular mesh, and the camera's maximum view depth $FOD_{\max}$ is also taken into consideration, i.e., only the triangular meshes within the reasonable view depth are included for the calculation of the viewpoint direction. Fig. 4 illustrates the candidate viewpoints and their corresponding directions generated by the random sampling method.

2.2. Visibility assessment

After generating candidate viewpoints by means of the random sampling method, the visibility assessment is carried out to calculate the visibility matrix, thereby converting the view planning problem into a set cover optimization problem. An optimization solution is derived from the visibility matrix.

The visibility matrix is a binary matrix of size $m \times n$, where m is the number of candidate viewpoints and n is the number of triangular meshes. If a triangular mesh is determined to be visible from a viewpoint, the corresponding element in the visibility matrix is set to 1; otherwise, it is set to 0. For a triangular mesh on the target surface to be considered visible by the camera, the following conditions must be fulfilled:

- The triangular mesh must lie within the camera's field of depth (FOD).
- The triangular mesh must fall within the camera's field of view (FOV).
- The angle between the viewing direction of the triangular mesh and the mesh's normal vector, i.e., the viewing angle, should lie within a certain range.
- The triangular mesh must not be occluded by other elements, which can be computed using a ray-triangle intersection test algorithm [22].

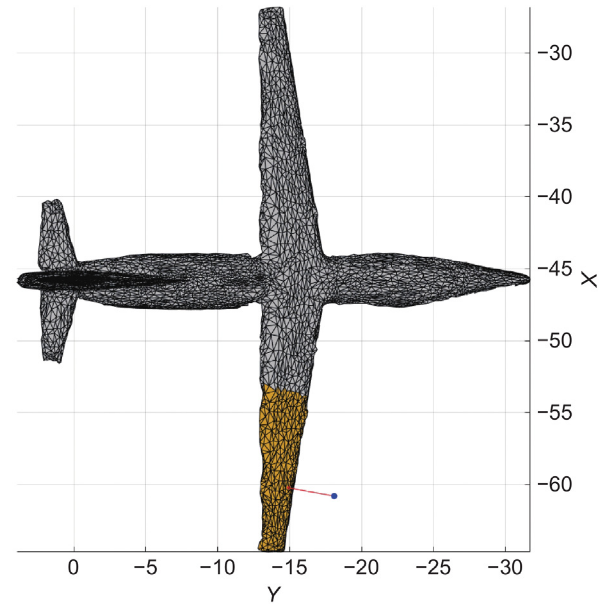


Fig. 5. A visible mesh (yellow) and its viewpoint (a blue point).

As an example, Fig. 5 illustrates the calculation of the visible triangular region for a specific viewpoint, with the yellow triangular meshes representing the visible portions of the airplane surface from that viewpoint.

After computing the visibility matrix, the view planning problem is transformed into a set cover problem [23], which is to find the smallest subset of viewpoints that can cover all the triangular meshes. The goal is defined as:

$$\begin{aligned} &\min \sum_{i=1}^{m} x_i, \quad \text{where } x_i \in \{0, 1\} \\ &\text{s.t.} \sum_{j=1}^{n} \left(\left(\sum_{i=1}^{m} x_i A_{ij} \right) \geqslant 1 \right) \geqslant \delta \cdot n \end{aligned} \tag{3}$$

where x_i denotes the ith candidate viewpoint, with $x_i = 1$ indicating selection and $s_i = 0$ indicating non-selection; $\boldsymbol{A}$ is the $m \times n$ visibility matrix; δ denotes the predefined target coverage ratio, which is set to 1 in this case, i.e., each triangular mesh needs to be covered once.

2.3. Combinatorial optimization

The set cover problem can be solved using a novel combinatorial optimization method. In this paper, we propose a hyper-heuristic algorithm to deal with this problem. Hyper-heuristic algorithms are conceptual models designed to solve complex optimization problems [24], which primarily manage and manipulate a series of low-level heuristics (LLHs) through a high-level heuristic strategy (HLH) to achieve optimization in the solution space. Compared to the traditional particle swarm optimization (PSO) [25] and genetic algorithms (GA) [26], the proposed hyper-heuristic algorithm exhibits better performance and a faster convergence rate.

Traditional heuristic algorithms, such as genetic algorithms, directly operate within the problem domain to generate solution vectors. Unlike traditional heuristic algorithms, the hyper-heuristic algorithm designed in this paper does not directly operate within the problem domain. Instead, it uses an evaluation function to select the low-level heuristic operators with relatively

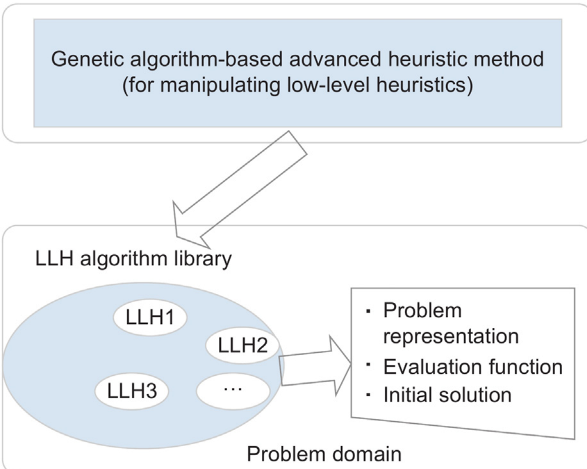


Fig. 6. The proposed hyperheuristic algorithm based on a genetic algorithm.

high fitness from the LLH algorithm library to generate low-level heuristic vectors. Each element represents the selected low-level heuristic operator, which can be applied to generate the solution vector, as shown in Fig. 6.

In this paper, we employ a genetic algorithm as a high-level heuristic strategy responsible for operating and managing a set of low-level heuristic operators. These low-level heuristic operators are designed based on neighborhood structures and aim to quickly solve problems by improving the fitness of solution vectors. This feature aims to achieve faster convergence speed and greater robustness.

2.3.1. High-level heuristic

The high-level heuristic algorithm designed in this study is based on genetic algorithms and mainly comprises fitness evaluation, initialization, selection, crossover, and mutation [27].

(1) **Initialization**. The initialization process involves initializing the problem solution. Both the low-level heuristic vector and the solution vector need to be initialized, and the initial values can be randomly generated as the initial solution.

(2) **Fitness evaluation**. For the set cover problem, this paper proposes a novel fitness calculation method to provide a more effective evolutionary direction for the optimization algorithm. The fitness value is proportional to the superiority of the individual, where the individual refers to any solution vector and its corresponding low-level heuristic vector.

Before calculating fitness, the number of triangular meshes that have been covered at least once is first determined:

$$n_{\text{cover}} = \sum_{i=1}^{n} \mathbb{I}\left((x \cdot \boldsymbol{A})_i \geq 1\right) \tag{4}$$

where x is the solution vector, which is a binary matrix where $x_i = 1$ indicates that the ith viewpoint is selected, otherwise, $x_i = 0$. $\boldsymbol{A}$ is the visibility matrix. The indicator function $\mathbb{I}$ returns 1 if the condition is true and 0 if it is false. The fitness calculation formula for the current solution vector is then:

$$f = \begin{cases} m - \sum_{i=1}^{m}, & if \quad n_{\text{cover}} = n \\ \frac{n_{\text{cover}}}{n}, & if \quad n_{\text{cover}} < n \end{cases} \tag{5}$$

where m is the number of candidate viewpoints. When $n_{\text{cover}} = n$, it indicates that all triangular meshes have been covered, and fewer viewpoints lead to higher fitness. We consider fitness evaluation in cases where the coverage requirements are not satisfied, rather than simply setting it to zero. The aim is to retain more solutions, enhance population diversity, and increase the likelihood of finding the global optimal solution.

(3) **Selection**. In genetic algorithms, individuals with higher fitness are more likely to be selected and replicated. In this stage, the roulette wheel selection method is utilized:

$$P_i = \frac{f_i}{\sum_{i=1}^{k} f_i} \tag{6}$$

where P_i is the selection probability, f_i is the individual fitness, and k is the population size. During this process, the low-level heuristic vectors and the solution vectors need to be replicated synchronously.

(4) **Crossover and mutation**. In this stage, traditional genetic algorithms perform crossover and mutation on solution vectors. However, the high-level heuristic algorithm in this paper only operates on the low-level heuristic vector without considering the solution vector. This algorithm adopts a segmented crossover strategy, which involves randomly selecting two points in the low-level heuristic vector and then exchanging the elements between these two points with the adjacent low-level heuristic vectors. The mutation strategy employs a "one-point" mutation strategy, where one element in the low-level heuristic vector is randomly selected and then randomly mutated to another element.

(5) **Generating new solution vectors**. After updating the low-level heuristic vector using crossover and mutation, new solution vectors are generated. Each element in the low-level heuristic vector represents a low-level heuristic operator, which is used to update the solution vector. If the element is 0, no operator is selected.

2.3.2. Low-level heuristics

Low-level heuristic algorithms play a key role in connecting high-level heuristic algorithms with problem solutions. The design of this part significantly impacts the effectiveness of hyper-heuristic algorithms [28,29]. In this study, four simple and efficient low-level heuristic operators are designed based on neighborhood structures to operate on solution vectors, specifically as follows:

- LLH1 (random mutation operator): We randomly select an element in the solution vector and inverts it.
- LLH2 (segmented crossover operator): For a given solution vector $\boldsymbol{s}_1$, we select an adjacent solution vector $\boldsymbol{s}_2$, randomly generate two index values, and swap the elements between these two index values. After the swap, a new solution is generated.
- LLH3 (random scattered crossover operator): Similar to the adjacent dispersed crossover operator, here we randomly select another solution vector $\boldsymbol{s}_2$ for crossover for a given solution vector $\boldsymbol{s}_1$.
- LLH4 (multi-segment crossover operator): For a given solution vector $\boldsymbol{s}_1$, we select an adjacent solution vector $\boldsymbol{s}_2$, randomly generate multiple pairs of index values, and swap the elements between these pairs of index values. After the swap, a new solution is generated.

3. Experiments

The view planning requires the UAV to perform exhaustive surface inspection of the target structure while minimizing the number of viewpoints to improve inspection efficiency. To validate the effectiveness of the proposed combinatorial optimization algorithm, comparative experiments are conducted. The two target object models used in the experiment are shown in Fig. 7, where the target object 1 is a real airplane model and the target

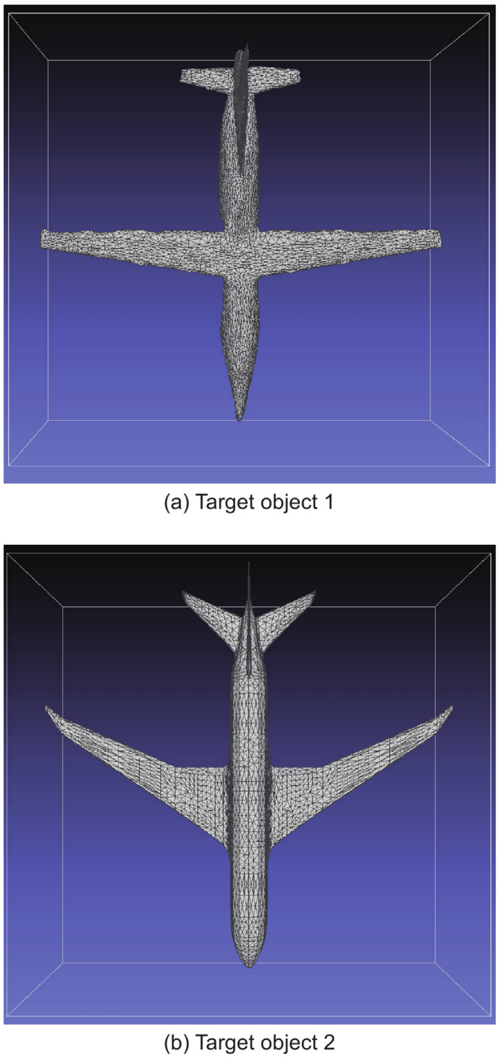


Fig. 7. The mesh model of upper surface of the airplane, where the target object 1 is a real airplane model and the target object 2 is a simulated airplane model with a different surface structure.

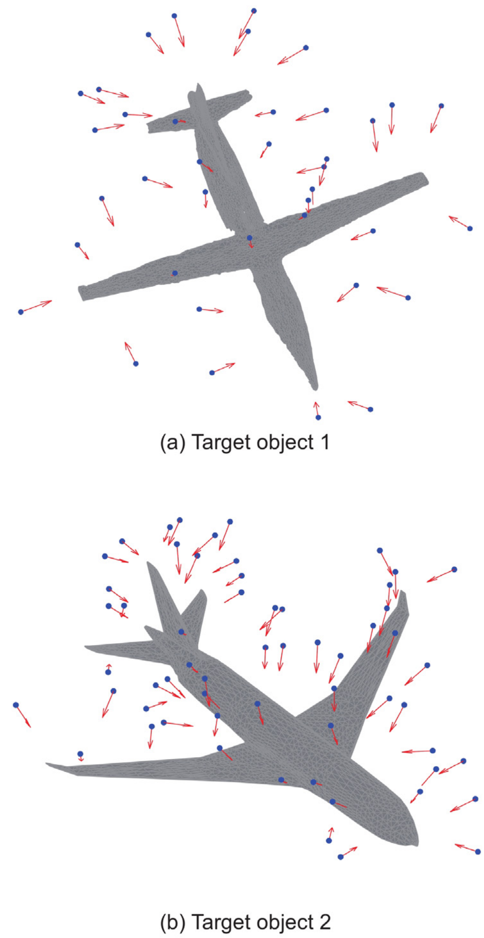


Fig. 8. Generated viewpoints using 3D target mesh models.

object 2 is a simulated airplane model with a different surface structure.

The parameters set for the view planning include a maximum and minimum view depth of 5m and 12m, respectively, a field of view angle of 80 deg, and a coverage requirement of 100%.

3.1. Comparison of random sampling methods

To compare with the original random sampling method, this paper conducts a comparative experiment based on the two airplane models shown in Fig. 7. For fairness, both methods use the same view planning parameters, with a greedy search algorithm as the common optimization method to find the optimal subset of viewpoints. Each method is run 10 times, and the average number of candidate viewpoints and optimal viewpoints is recorded as the final results.

As shown in Table 1, for two airplane models, our method reduces the number of viewpoints by 3.30% and 3.62%, respectively. The original random sampling method generates a significant portion of candidate viewpoints that are too far from the surface of the target object, thereby reducing coverage efficiency. We improve the generation method of the sampling space to address this issue. The results indicate that the proposed method with combinatorial optimization generates fewer candidate viewpoints and fewer optimal viewpoints, suggesting that the quality of the generated viewpoints and coverage efficiency are both higher. Fig. 8 shows an example of the 3D visualization results from the proposed method for planning viewpoints.

3.2. Comparison of combinatorial optimization algorithms

To verify the effectiveness of the proposed hyper-heuristic algorithm, we compare it with the Genetic Algorithm (GA), Adaptive Particle Swarm Optimization (APSO), and Random Key Genetic Algorithm (RKGA) [19]. We use the 10 sets of candidate viewpoints obtained by our method, as shown in Table 1, as the input for these algorithms and compare the number of final viewpoints in the optimization results. Since combinatorial optimization involves a degree of randomness, we run each set of results five times using each of the four optimization algorithms and take the average as the final result, with the greedy algorithm optimization results serving as the initial solution. The final results are shown in Tables 2 and 3.

As shown in Table 2, the comparison of the number of viewpoints in the optimal subset shows that APSO tends to get stuck in local optima. Even after a sufficient number of iterations, it fails to find the global optimum, leading to suboptimal performance compared to the greedy search algorithm used in the original random sampling method. The mutation strategy of the Genetic Algorithm helps it effectively escape local optima, allowing GA,

Table 1
Comparison of candidate viewpoints and optimal viewpoints.

(a) Results for the target object 1

Groups	Candidate viewpoints		Optimal viewpoints	
	Original	Ours	Original	Ours
1	125	**107**	**37**	38
2	**120**	132	39	**37**
3	135	**118**	39	**37**
4	128	**122**	44	**42**
5	**126**	142	**47**	**47**
6	148	**114**	45	**41**
7	**127**	134	**42**	44
8	118	**111**	**43**	**43**
9	137	**124**	45	**41**
10	**111**	125	43	**40**
Averages	127.5	**122.9**	42.4	**41**

(b) Results for the target object 2

Groups	candidate viewpoints		optimal viewpoints	
	Original	Ours	Original	Ours
1	159	**156**	60	**59**
2	**150**	154	64	**61**
3	159	**139**	**60**	62
4	157	**135**	**56**	62
5	**160**	**160**	71	**61**
6	168	**143**	**61**	63
7	156	**147**	69	**56**
8	**142**	159	**62**	66
9	**150**	190	67	**59**
10	196	**142**	65	**63**
Averages	159.7	**152.5**	63.5	**61.2**

Table 2
Comparison of experimental results using different optimization algorithms.

Groups	APSO	GA	RKGA	GA-HH
1	38.0	**36.0**	**36.0**	**36.0**
2	37.0	**36.0**	**36.0**	**36.0**
3	37.0	**35.0**	**35.0**	**35.0**
4	42.0	39.2	39.2	**39.0**
5	47.0	**43.0**	**43.0**	**43.0**
6	41.0	37.4	37.2	**37.0**
7	44.0	**39.0**	**39.0**	**39.0**
8	43.0	**40.0**	**40.0**	**40.0**
9	41.0	38.0	37.8	**37.4**
10	40.0	**38.0**	**38.0**	**38.0**
Averages	41.00	38.16	38.12	**38.04**

Table 3
Comparison of iteration counts.

Groups	GA	RKGA	GA-HH
1	42.6	31.2	**26.4**
2	**16.6**	17.8	18.2
3	43.2	**42.2**	43.6
4	47.0	44.8	**33.2**
5	66.8	69.0	**47.4**
6	123.0	110.2	**47.8**
7	89.0	93.0	**72.0**
8	68.8	65.0	**43.2**
9	81.0	**68.4**	**68.4**
10	46.0	55.0	**33.8**
Averages	62.40	59.66	**43.40**

RKGA, and GA-HH to select the smallest optimal viewpoint subset from the candidate viewpoints. In contrast, out of 50 experiments, GA, RKGA, and GA-HH encounter local optima 8, 6, and 2 times, respectively, indicating that GA-HH is more likely to find the global optimum, since the proposed hyper-heuristic algorithm uses a genetic-based high-level heuristic strategy to manage multiple low-level heuristics. It selects different low-level heuristic methods based on the specific problem and its stage, aiding in the discovery of the global optimum. Compared to traditional heuristic algorithms, GA-HH achieves more robust and efficient results.

Fig. 9. A DJI M30 drone is used to inspect a real aircraft surface based on the generated viewpoints.

Table 3 presents the comparison of the number of iterations required for the three combinatorial optimization algorithms to converge. The results show that GA-HH converges significantly faster than the other two algorithms, since GA-HH incorporates a diversified search strategy by combining the advantages of various low-level heuristics and adjusting the search direction based on the quality of the current solution to accelerate convergence, whereas traditional heuristic methods often rely on predetermined rules and lack flexibility. Finally, we demonstrate our algorithm using a DJI M30 drone to inspect a real aircraft surface using the target model 1 as shown in Fig. 9. Based on the viewpoint data obtained from the GA-HH algorithm, we find that the average execution time of the DJI M30 drone is 10.29 min, and the average path length is 326.75 m.

4. Conclusion

This paper investigates the UAV view planning problem for visual coverage tasks on the upper surface of large airplanes. Based on random sampling method, improvements are made to enhance the quality of the generated viewpoints and coverage efficiency. Additionally, a genetic algorithm-based hyper-heuristic (GA-HH) is proposed to solve the combinatorial optimization problem, enabling a more effective selection of an optimal viewpoint subset from the candidate viewpoints to improve inspection efficiency. Experimental results show that the improved random sampling method reduces the required number of viewpoints by 3.46%. Furthermore, the proposed GA-HH reduces the probability of getting stuck in local optima and the number of iterations to convergence by 66.67% and 27.25%, respectively, compared with advanced heuristic algorithms. The experimental results demonstrate that the proposed methods can significantly reduce the required number of viewpoints, and the proposed hyper-heuristic algorithm is more robust and efficient compared to traditional heuristic methods.

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.

Acknowledgment

This research was supported by the Trade Union Employee Innovation Foundation of China (2022270024).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.birob.2025.100228.

References

[1] D. Jiajia, G. Xiaoxi, W. Jun, Coverage path planning method of unmanned aerial vehicle for aircraft surface detection task, J. Mech. Eng. 59 (16) (2023) 243–253.
[2] D.A. Rodriguez, C.L. Tafur, P.F.M. Daza, J.A.V. Vidales, J.C.D. Rincón, Inspection of aircrafts and airports using UAS: a review, Results Eng. 22 (2024) 102330.
[3] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, G. Cai, Aircraft inspection using unmanned aerial vehicles, in: International Micro Air Vehicles, Conferences and Competitions Proceedings, 2016, pp. 1–7.
[4] W. Jing, J. Polden, W. Lin, K. Shimada, Sampling-based view planning for 3D visual coverage task with Unmanned Aerial Vehicle, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2016, pp. 1808–1815.
[5] S. Chen, Y. Li, N.M. Kwok, Active vision in robotic systems: A survey of recent developments, Int. J. Robot. Res. 30 (11) (2011) 1343–1377.
[6] M. Maboudi, M. Homaei, S. Song, S. Malihi, M. Saadatseresht, M. Gerke, A review on viewpoints and path planning for UAV-based 3-D reconstruction, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 16 (2023) 5026–5048.
[7] W.R. Scott, Model-based view planning, Mach. Vis. Appl. 20 (1) (2009) 47–69.
[8] Y. Choi, Y. Choi, S. Briceno, D.N. Mavris, Three-dimensional UAS trajectory optimization for remote sensing in an irregular terrain environment, in: 2018 International Conference on Unmanned Aircraft Systems, ICUAS, IEEE, 2018, pp. 1101–1108.
[9] Y. Zheng, G. Liu, Y. Ding, G. Tian, A new clustering-based view planning method for building inspection with drone, IEEE Robot. Autom. Lett. 9 (11) (2024) 9781–9788.
[10] W. Hua, X. Xiaoshun, B. Xiaoqing, Multi-viewpoint planning for unmanned aerial vehicles based on continuous differentiable sampling, J. Comput. Integr. Manuf. Syst. 30 (3) (2024) 1161.
[11] V. Staderini, T. Glück, P. Schneider, R. Mecca, A. Kugi, Surface sampling for optimal viewpoint generation, in: 2023 IEEE 13th International Conference on Pattern Recognition Systems, ICPRS, IEEE, 2023, pp. 1–7.
[12] Y. Wang, T. Peng, W. Wang, M. Luo, High-efficient view planning for surface inspection based on parallel deep reinforcement learning, Adv. Eng. Inf. 55 (2023) 101849.
[13] F. Kong, F. Du, D. Zhao, Station-viewpoint joint coverage path planning towards mobile visual inspection, Robot. Comput.-Integr. Manuf. 91 (2025) 102821.
[14] H. Razip, N. Zakaria, Genetic algorithm with approximation algorithm based initial population for the set covering problem, in: Proceedings of International Conference on Communication and Computational Technologies, ICCCT 2021, Springer, 2021, pp. 59–78.
[15] T. Dokeroglu, T. Kucukyilmaz, E.-G. Talbi, Hyper-heuristics: A survey and taxonomy, Comput. Ind. Eng. 187 (2024) 109815.
[16] F. Zhao, Y. Liu, N. Zhu, T. Xu, et al., A selection hyper-heuristic algorithm with Q-learning mechanism, Appl. Soft Comput. 147 (2023) 110815.
[17] H.-B. Song, J. Lin, A genetic programming hyper-heuristic for the distributed assembly permutation flow-shop scheduling problem with sequence dependent setup times, Swarm Evol. Comput. 60 (2021) 100807.
[18] C.S. Tan, R. Mohd-Mokhtar, M.R. Arshad, A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms, IEEE Access 9 (2021) 119310–119342.
[19] P. Gospodnetic, D. Mosbach, M. Rauhut, H. Hagen, Viewpoint placement for inspection planning, Mach. Vis. Appl. 33 (1) (2022) 2.
[20] W. Jing, K. Shimada, Model-based view planning for building inspection and surveillance using voxel dilation, medial objects, and random-key genetic algorithm, J. Comput. Des. Eng. 5 (3) (2018) 337–347.
[21] R.M. Haralick, S.R. Sternberg, X. Zhuang, Image analysis using mathematical morphology, IEEE Trans. Pattern Anal. Mach. Intell. (4) (1987) 532–550.
[22] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, J. Graph. Tools 2 (1) (1997) 21–28.
[23] A.S. Ferreira, A. Pozo, R.A. Gonçalves, et al., An ant colony based hyper-heuristic approach for the set covering problem, Adv. Distrib. Comput. Artif. Intell. J. 4 (1) (2015) 1–21.
[24] E.K. Burke, M.R. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward, A classification of hyper-heuristic approaches: revisited, Handb. Metaheuristics (2019) 453–477.
[25] T.M. Shami, A.A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M.A. Summakieh, S. Mirjalili, Particle swarm optimization: A comprehensive survey, IEEE Access 10 (2022) 10031–10061.
[26] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools Appl. 80 (2021) 8091–8126.
[27] J.E. Beasley, P.C. Chu, A genetic algorithm for the set covering problem, European J. Oper. Res. 94 (2) (1996) 392–404.
[28] J. Jianping., C. Kun., C. Wang, Set covering problem based on improved genetic algorithm, Math. Pract. Theory 24 (5) (2012) 120–126.
[29] G. Koulinas, L. Kotsikas, K. Anagnostopoulos, A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, Inform. Sci. 277 (2014) 680–693.