COVERAGE PATH PLANNING FOR AUTONOMOUS STRUCTURAL INSPECTION USING UAV

TONG HO WANG

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

Coverage Path Planning for Autonomous Structural Inspection Using UAV

Tong Ho Wang

A thesis submitted in partial fulfilment of the requirements for the degree

of

Doctor of Philosophy

Aug 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____________ (Signed)

Tong Ho Wang (Name)

Abstract

With the emerging concept such as low-altitude economy, Unmanned Aerial Vehicles (UAVs) are becoming a hot topic in the research field, and different applications for UAVs have been proposed. With its unique mobility, one of the emerging civil applications of UAVs is autonomous inspection. UAVs can freely navigate around the inspection target and collect data needed for different inspection purposes. In order to perform the autonomous mission, a planning algorithm is required to plan a suitable inspection path for the inspection mission. In this thesis, different aspects of inspection path planning are studied, and different algorithms are proposed to improve these aspects.

First, the scalability of the inspection path planning is studied. A mixed viewpoint generation technique is proposed to improve the scalability of the path planning algorithm such that suitable inspection path can be generated for different inspection purposes. The proposed algorithm utilizes two kinds of viewpoints. The revolving viewpoints ensure that uniform coverage is achieved while the gap-filling viewpoints ensure that the coverage requirement is satisfied. A Multi-layered Angle-distance Traveling Salesman Problem (ML-ADTSP) is also proposed such that a spiral inspection path can be obtained, guaranteeing the kinodynamic feasibility of the inspection path. Using the proposed approach, suitable inspection paths can be generated for different inspection purposes, ranging from crack detection to 3D reconstruction.

Second, to improve the efficiency of the inspection task, a multi-UAV system for inspection purposes is proposed. A multi-objective multiple Traveling Salesman Problem (MO-MTSP) is proposed to compute the optimized inspection path for different UAVs. The NSGACO hybrid algorithm is proposed to solve the combinatorial optimization problem. The algorithm allows the benefit of Non-dominated Sorting Genetic Algorithm (NSGA-II) and Ant Colony Optimization (ACO) to be realized throughout the iteration, improving the quality and diversity of the solutions obtained. The proposed problem also treats the location of the depot as an optimization variable, and a migration mechanism is proposed to search for the optimal depot location that aligns with the optimization objectives. The proposed algorithm allows the deployment of autonomous inspection systems that can perform repeatable

inspection missions, which can be applied in construction safety inspection or aircraft inspection, where frequent inspections are required.

Lastly, a parameterization-based path planning algorithm is proposed to improve the quality of the inspection task. The algorithm makes use of the parameterized surface of the 3D model to directly plan a sweep path, and dimensions-related complications can be ignored during the planning process. A barycentric-based transformation and optimization-based re-planning module are used to transform the path into 3D while minimizing the distortion during the process. To further demonstrate the scalability of the planning approach, it is used with the multi-UAV system proposed, and simulation results verify the performance of the proposed algorithm. The proposed algorithm with improved inspection quality can be applied in aircraft inspection where the quality of the task is the first priority.

List of Publications

[1] H. W. Tong, B. Li, H. Huang and C. Wen, "UAV Path Planning for Complete Structural Inspection using Mixed Viewpoint Generation," 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Singapore, 2022, pp. 727-732, doi: 10.1109/ICARCV57592.2022.10004359.

[2] H. W. Tong, B. Li, H. Huang and C. Wen, "Coverage Path Planning for Autonomous Aircraft Inspection Using UAVs," AIAA SciTech Forum, 2025, doi: 10.2514/6.2025-0216.

[3] H. W. Tong, B. Li, H. Huang, and C.-Y. Wen, "Multi-Layer Path Planning for Complete Structural Inspection Using UAV," Drones (Basel), vol. 9, no. 8, p. 541, 2025, doi: 10.3390/drones9080541.

[4] H. W. Tong, B. Li, H. Huang, and C.-Y. Wen, "Multi-UAV Coverage Path Planning for Autonomous Aircraft Inspection," J. of Air Transportation, pp. 1-12, 2026 , doi: 10.2514/1.D0540.

[5] Z. Tan et. al, " Multi-UAV coverage path planning: Enabling automated inspection of complex structures,", 11TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS), Rome, Italy, 2025, doi: 10.13009/EUCASS2025-252.

Acknowledgement

This thesis marks the culmination of my four-year journey as a research student, filled with a range of emotions. First, I would like to express my heartfelt gratitude to Dr. Boyang Li for accepting me as his research student. Dr. Li has offered invaluable support throughout my studies and has continually inspired me to become a better researcher. Whenever I faced doubts, he provided guidance to illuminate my path, and whenever I made questionable decisions, he challenged me to steer me back in the right direction.

I would also like to thank Prof. Chih-yung Wen and Dr. Hailong Huang for their additional guidance and support during my journey.

Lastly, I am deeply grateful to my family for their unwavering support. Without their encouragement, completing this four-year journey would have been impossible.

Table of Contents

List of abbreviations

ACO	Ant Colony Optimization
ACS	Ant Colony System
ACBIO	Ant Colony Binary Iterative Algorithm
AOI	Area of Interest
BF	Best Fit Crossover
BSO	Brainstorming Optimization
CPP	Coverage Path Planning
C2C	Cloud-to-Cloud
DGA	Dynamic Genetic Algorithm
ER	Edge Recombination Crossover
ETSP	Euclidian Traveling Salesman Problem
GPU	Graphics Processing Unit
GA	Genetic Algorithm
HGA	Hardware-based Genetic Algorithm
Hv	Hypervolume Indicator
IWO	Invasive Weed Optimization
IPGA	Improved Partheno-Genetic Algorithm
ML-ADTSP	Multi-layered Angle-Distance Traveling Salesman Problem
MO	Multi-objective
MTSP	Multiple Traveling Salesmen Problem
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NP	Nondeterministic Polynomial Time
NSGACO	Non-domination Sorting Genetic Ant Colony Optimization
OL-ADTSP	Open Loop Angle-Distance Traveling Salesman Problem
OX	Order Crossover

PGA	Partheno-Genetic Algorithm
PMX	Partially-mapped Crossover
PSO	Particle Swarm Optimization
QTSP	Quadratic Traveling Salesman Problem
RRT	Rapidly Exploring Random Tree
SCP	Set Cover Problem
SC-VRP	Set Covering Vehicle Routing Problem
SPEA2	Strength Pareto Evolutionary Algorithm
SOS	Symbiotic Organisms Search
TSP	Traveling Salesman Problem
TSPLIB	Traveling Salesman Problem Library

List of figures

List of tables

1 Introduction

This section introduces the background for the thesis, discussing the emerging applications and challenges of Unmanned Aerial Vehicles (UAVs) in the low-altitude economy. Finally, the research objectives are summarized.

1.1 Background

Fig. 1.1 Examples of aging buildings in Hong Kong.

With advancements in UAV technology, the applications of UAVs have grown exponentially. Newly proposed economic policies, such as Urban Air Mobility in the United States and the Low-Altitude Economy in China, have significantly increased the demand for UAV-related research. There is particularly high demand in civil applications such as search and rescue, infrastructure inspections, and surveillance [1]. While UAVs have been effectively utilized for search and rescue in recent years, their application in autonomous inspection remains limited. Modern cities like New York and Hong Kong face challenges related to aging buildings. As of 2017, the median age of residential buildings in New York City reached 90 years [2], while Hong Kong, a younger city, has an average building age of 34.4 years [3]. To ensure public safety, these aging structures require regular maintenance, and inspections are essential for early detection of potential safety hazards. Currently, most infrastructure inspections are performed manually, requiring trained personnel to examine the exterior of structures for defects, which is both time-consuming and cost ineffective. In contrast, autonomous vehicles offer superior mobility and can navigate around structures freely, dramatically improving inspection efficiency. As a result, sensor-equipped UAVs hold great potential in the field of autonomous inspection.

In addition to infrastructure inspections, other types of inspections can also benefit from the use of autonomous vehicles. Specifically, autonomous inspections can be applied to aircraft inspection. Similar to infrastructure inspections, aircraft inspection involves a repetitive manual process that is time-consuming, which can significantly increase ground time. By automating this process, the efficiency of inspections can be improved, streamlining the pre-flight inspection procedure, reducing aircraft ground time, and minimizing delays caused by inspection tasks.

To enable autonomous inspection using UAVs, several fundamental problems must be addressed, particularly the Coverage Path Planning (CPP) problem. The CPP involves planning a path that effectively covers an area of interest (AOI) while avoiding obstacles [4]. This problem has numerous applications in robotics, including search and rescue, surveillance, and vacuum cleaning. One of the popular approaches in solving CPP is to divide it into two subproblems: the view planning problem and the viewpoint assignment problem. The view planning problem focuses on identifying optimal viewpoints that cover the AOI, while the assignment problem aims to assign the viewpoints to the UAV such that it can navigate through the viewpoints in an efficient manner. When designing algorithms to solve the CPP for inspection-related applications, several factors must be considered:

1) Efficiency: The computed inspection path should optimize the UAV's traveling distance to maximize the inspection efficiency.
2) Inspection quality: As inspection is a safety-related task, the quality of the inspection should be the first priority. The computed inspection path should provide promising inspection quality that allows for a reliable inspection.
3) Generalizability: The inspection path planning algorithm should be applicable to different inspection techniques and structures with different geometries.

1.2 Research Objectives

This thesis facilitates the integration of UAVs into the low-altitude economy by advancing autonomous inspection through efficient and feasible algorithmic solutions.

First, the thesis addresses the research gap where existing CPP algorithms lack scalability, limiting their applications. These algorithms, regardless of their categories, are designed specifically for a particular inspection technique, with strictly defined parameters. Optimized for these specific parameters, their performance deteriorates when applied to incompatible scenarios. To address this gap, the thesis proposes a mixed viewpoint generation approach, enabling scalable inspection path generation suitable for various inspection scenarios.

Second, the thesis aims to improve the inspection efficiency of CPP algorithms. Specifically, the generated paths for inspecting high-rise buildings are often too long for a single UAV to navigate within battery limits. Autonomous inspection can benefit from utilizing multiple UAVs and distributing the inspection tasks evenly among them, thereby enhancing inspection efficiency. To address this issue, the thesis proposes a multi-UAV inspection system equipped with a deployable charging station, which enables efficient and repeatable inspection missions.

Third, the thesis addresses the research gap where existing CPP algorithms lack focus on inspection quality. While these algorithms primarily concentrate on improving the efficiency of inspection tasks, little attention has been given to the quality of the inspections themselves. Given that inspection is a safety-related task, ensuring high quality should be the top priority. A quality inspection task must guarantee that defects are successfully detected during the process. To address this issue, the thesis proposes a parameterization-based planning algorithm that ensures the resulting inspection path closely follows the contours of the inspection target, preventing inconsistent viewing distances and thereby enhancing inspection quality.

The research objectives of the thesis can be summarized as follows:

1) Scalability: The CPP solver must generate adaptable inspection paths for diverse techniques (e.g., crack detection, photogrammetry), eliminating the need for technique-specific algorithms.
2) Efficiency:
    a. Multi-UAV Coordination: Inspection efficiency is enhanced through collaborative multi-UAV systems.

b. Charging Station Optimization: For persistent missions (e.g., aircraft inspection), charging station placement is optimized to minimize downtime.

3) Inspection Quality: While efficiency is critical, defect detection reliability remains paramount. The CPP solver ensures sufficient coverage and resolution to guarantee high-quality inspections.

1.3 Organization of the thesis

The rest of the thesis is organized as follows:

- Chapter 2 reviews the literature related to the research topic and points out some of the research gaps which this research work is focused on.
- Chapter 3 introduces the CPP algorithm that aims to improve the scalability of the autonomous inspection task.
- Chapter 4 proposes a new CPP solver that utilizes parameterization and aims to improve the overall inspection quality.
- Chapter 5 introduces a heuristic algorithm that aims to solve the multi-UAV inspection path planning problem. It also concerns about the charging station optimization to further optimize the efficiency of the inspection task.
- Chapter 6 concludes the thesis and provides discussion on the future work.

2 Literature Review

The topic of CPP spans across multiple problems. This section focuses on reviewing literature related to view planning and path planning for inspection-related applications. Literature on Multiple Traveling Salesmen Problem (MTSP) solver will also be discussed. The literature related to CPP is summarized in **Fig**. 2.1.

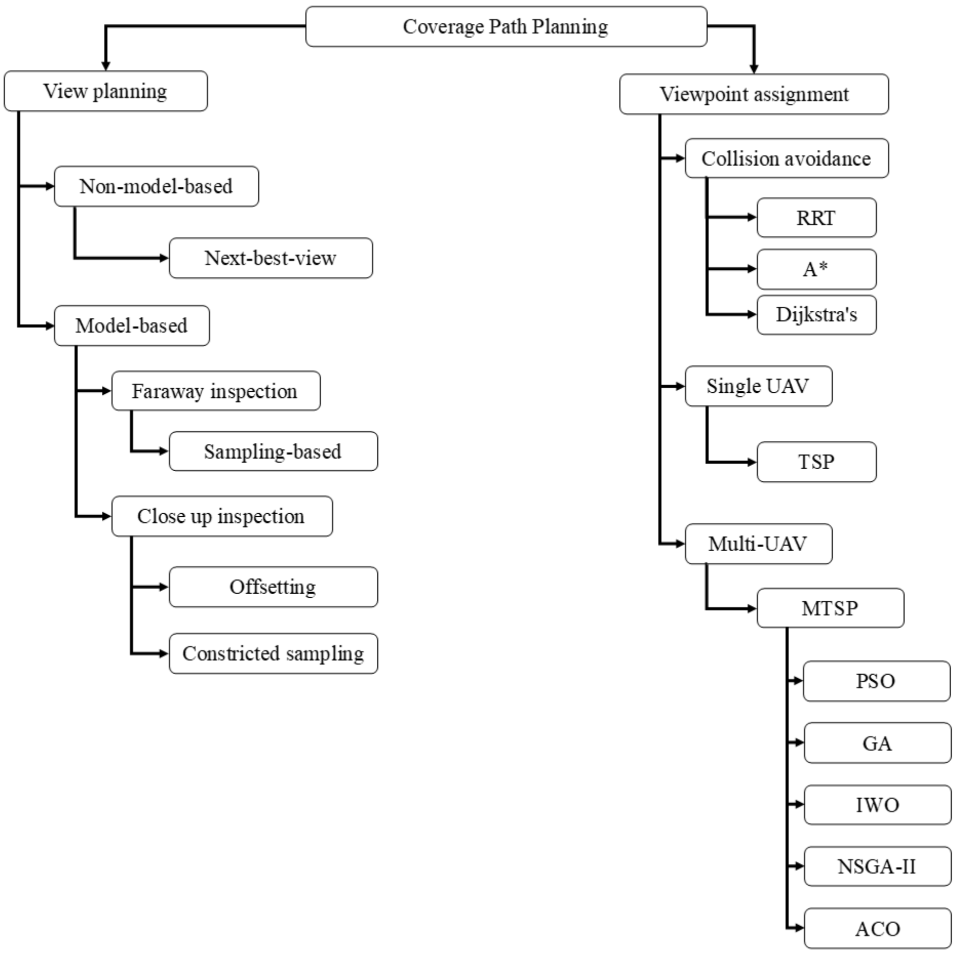


Fig. 2.1 Summary of related literature.

2.1 Inspection techniques

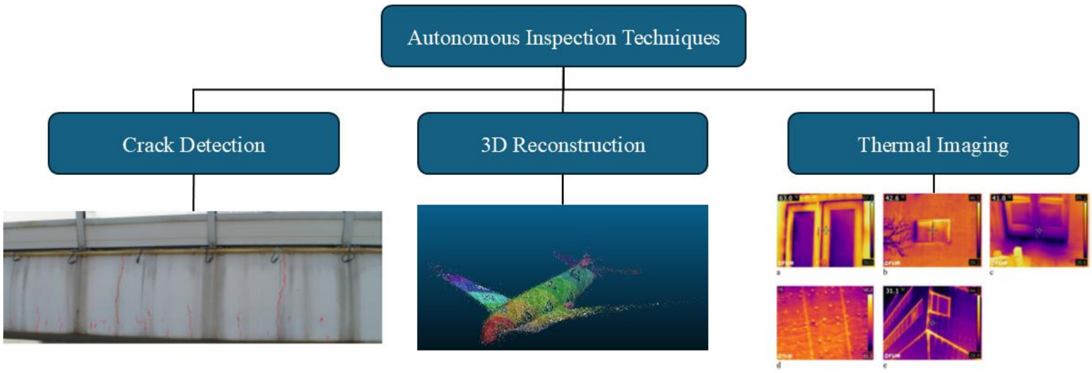


Fig. 2.2 Categorization of different autonomous inspection techniques.

Before exploring various inspection path planning algorithms, it is essential to discuss the related inspection techniques, as these algorithms are designed based on specific methods. Tarek and Alice [5] summarized different inspection techniques used with unmanned vehicles, categorizing them into three main types: crack detection, photogrammetry, and thermography, as illustrated in **Fig.** 2.2. The required viewing distance varies by inspection method. For example, close-up inspections like crack detection necessitate an inspection distance of approximately 3 meters [6] to enable effective defect detection. Crack detection utilizes artificial intelligence to detect potential cracks or other defects in the captured images. The method requires the captured images to be of high quality, and the defects concerned should be clearly visible in the images, under the determined resolution. Hence, the images should be taken close to the inspection target, and a consistent viewing distance should be maintained along the planned path. Moreover, motion blur of the captured images should also be taken into account and the camera's shutter speed should match the flight speed of the UAV.

In contrast, methods such as photogrammetry require a distance of up to 25 meters [7] to capture images with sufficient features for the reconstruction process. This type of inspection technique focuses on identifying large defects such as deformation by comparing the reconstructed model with the reference model. The technique requires adequate features to be captured across images for featuring matching. Hence, the images should be taken faraway from the inspection target such that several features can be included. The features captured should be clearly presented in the images, with a consistent size across images to prevent scale ambiguity. As a result, similar to crack detection, a consistent viewing distance should also be maintained along the planned path.

Lastly, thermal imaging is used to detect thermal leakage in the inspection target. As thermal patterns usually spread across a larger area, the viewing distance requirement is less strict compared to other inspection techniques. The viewing distance requirement for thermal imaging can range from 5m [8] to 20m [9] depending on the size of the inspection target. The captured thermal image can also be used to reconstruct the thermal model of the target to better identify locations with thermal leakage. However, unlike other inspection techniques, thermal imaging requires the images to be taken at a specific time of day to achieve a more reliable result.

Therefore, the coverage path planning algorithm must accommodate the inspection distance requirements of different techniques while maintaining a consistent inspection distance to ensure optimal inspection results.

To generate an effective coverage path for inspection tasks, it is necessary to compute a set of viewpoints that provide adequate coverage of the inspection target. To ensure the effectiveness of the inspection path, the minimal number of viewpoints should be selected. Consequently, solving the view planning problem is essential. There are two primary approaches to the view planning problem: model-based and non-model-based approaches [10]. In the model-based approach, viewpoints are generated using prior information about the inspection target. In contrast, non-model-based approaches employ a "next-based view" method, generating viewpoints without prior knowledge of the inspection target.

2.2 Single Agent Coverage Path Planning

2.2.1 Non-model-based planning

For non-model-based approaches, Song and Jo [20] proposed an online planning method that emphasizes the completeness of inspection by determining the next-best view to maximize information gain. The authors of [21] focus on planning techniques that satisfy traveling distance and observability constraints, enabling the algorithm to produce a more efficient inspection path. However, due to the exploratory nature of next-best-view methods, these planning approaches are not thoroughly optimized, often resulting in longer travel distances. Additionally, the next-best-view approach generates viewpoints dynamically as the coverage mission progresses, which can negatively impact overall inspection quality. Therefore, this thesis will concentrate on the model-based approach, which ensures guaranteed inspection quality.

2.2.2 Close-up inspection

This research focuses on model-based methods, which provide improved inspection quality over non-model-based approaches by generating viewpoints tailored to the inspection target. Model-based approaches can be further categorized based on their designated inspection methods. View planning methods designed for crack detection and similar tasks are classified as close-up inspection methods, while those intended for 3D reconstruction are categorized as faraway inspection methods. The difference in viewing distance necessitates distinct inspection methods.

In close-up inspection methods, algorithms typically relate closely to the geometry of the inspection target. For instance, Alexis et al. [11] proposed a planning algorithm that generates viewpoints by offsetting the surface points of the inspection target. To reduce excessive computation, the level of detail is refined at each iteration, adding more viewpoints as the process continues. Although this results in an inspection path that closely follows the target's contour, the complexity of the path is not considered. Additionally, inferior viewpoints generated early in the iteration may compromise the overall quality of the inspection path, as these viewpoints are based on a downscaled model. Scott [12] introduced a sampling-based approach where viewpoints are generated through random sampling near the inspection target. For each surface point, a viewpoint is sampled that optimally observes that point. After the sampling phase, the Set Cover Problem (SCP) is solved to obtain an optimal set of viewpoints that satisfy coverage requirements. By limiting the sampling space to areas near the inspection target, this algorithm ensures that the resulting inspection path closely follows the target's contour, which is advantageous for the inspection process. However, the computation time increases exponentially with the complexity of the inspection target, rendering this approach less suitable for inspecting complex structures such as aircraft.

References [22-24] propose various strategies for the autonomous inspection of aircraft. In [22], the algorithm begins by discretizing the free space around the inspection target, with each discretized position representing a potential viewpoint. The yaw angle of each viewpoint is also discretized. These viewpoints are then filtered based on collision, distance, and coverage requirements. A modified A* algorithm, biased towards reconstruction accuracy, is employed to compute the optimal inspection path. Given the extensive computational demands of this method, GPU power is utilized to accelerate the calculations. To further enhance reconstruction quality, an adaptive viewpoint sampling approach is introduced in [23]. This method generates more viewpoints in areas with less coverage while reducing viewpoints in well-covered regions. This adaptive strategy improves overall reconstruction quality while minimizing unnecessary computation. Finally, in [24], an additional heuristic function is employed to optimize coverage accuracy, travel distance, and turning angle. Brendan and Franz [13, 14] also proposed a sampling-based approach aimed at generating a sweep path for the inspection of complex structures. In this method, viewpoints are first sampled in the vicinity of surface points, and the visibility of the sweep path is subsequently estimated. To ensure 100% coverage, gap-filling viewpoints are generated to enhance the original sweep path. The sweep path closely follows the contour of the target, resulting in promising inspection quality. However, due to the sampling nature of this method, it suffers from poor scalability, and the performance of the sweep path degrades as the inspection distance increases.

In summary, sampling-based techniques are popular for close-up inspections due to their flexibility in designing inspection paths, as they decouple the view planning problem from the overall CPP. However, these methods face several fundamental challenges. First, the approach is computationally expensive. By dividing the CPP into two subproblems, multiple optimization problems must be solved, specifically the SCP and the Traveling Salesman Problem (TSP). Additionally, there is a dilemma between increasing the sampling size to enhance overall coverage and reducing the sampling size to decrease computation time.

2.2.3 Faraway Inspection

On the other hand, for view planning algorithms designed for faraway inspections, Tarbox and Gottschlich [15] proposed a spherical dilation approach. In this method, candidate viewpoints are generated along the surface of a sphere that encapsulates the inspection target, and appropriate viewpoints are selected. However, this approach is tailored for product inspection on production lines and does not account for potential view obstructions. Additionally, the method has scaling limitations that restrict its applicability. In references [16-19], various sampling-based view planning techniques are explored. A random sampling approach is introduced in [16], where a sampling space is established to meet inspection criteria while satisfying safety constraints. Viewpoints are then sampled within this space, and the SCP is solved to obtain the optimal set of viewpoints. This concept is further elaborated in [17], where the sampling space is generated by sampling the medial object of the dilated inspection target, resulting in a higher coverage ratio per viewpoint. To leverage UAV capabilities, [18] investigates path primitive sampling instead of traditional viewpoint sampling. Since the UAV is equipped with a camera capable of video recording, coverage is computed based on the path traveled between two waypoints rather than on discretized viewpoints. This approach maximizes UAV potential, achieving additional coverage while significantly reducing the traveling distance of the inspection path. Finally, the idea is extended to multi-agent scenarios, where a Set Covering Vehicle Routing Problem (SC-VRP) is formulated to select optimal path primitives and assign them to individual UAVs.

In [26], the convex hull of the inspection target at various heights is first obtained, and waypoints are computed along this convex hull. A multi-objective optimization problem is then solved to determine the inspection path. While this approach offers an intuitive method for solving the CPP problem, it may struggle as the complexity of the inspection target increases. Reference [27] introduces a dynamic inspection path planning approach that leverages object detection. After identifying the turbine blade, waypoints are generated parallel to the blade's curvature. Although this research presents a comprehensive method for inspecting wind turbines, it lacks generalizability. Moreover, the UAV must operate at a distance from the target for the detection algorithm to function effectively, resulting in limited scalability. Lastly, in [25], a multi-UAV coverage path planning algorithm is proposed. This method employs a slicing technique to extract geometric information from the inspection target. Waypoints are generated at each level, and an optimization problem is formulated to obtain an optimal, collision-free inspection path that connects all viewpoints.

In summary, sampling and dilation are among the most popular techniques for faraway inspections, offering straightforward solutions to the CPP. Unlike close-up inspections, sampling for faraway inspections has less restrictions and viewpoints are generated freely within a large sampling space. However, these approaches have limitations. Specifically, the dilation technique may struggle to generate viewpoints in clustered areas as the offsetting distance decreases. Additionally, the quality of the inspection is not guaranteed, as the sampling approach can produce viewpoints with varying viewing distances, which can adversely affect inspection quality. While it is possible to limit the viewing distance by contracting the sampling space, this adjustment increases the likelihood of algorithm failure.

2.3 Multiple Traveling Salesmen Problem

After addressing the view planning problem, the next step is to solve the viewpoint assignment problem. The primary objective is to obtain a collision-free path that navigates through all viewpoints. Various optimization objectives can also be employed to tailor the inspection path for specific purposes.

One widely chosen optimization objective is to minimize the overall traveling distance, which enhances the efficiency of autonomous inspection tasks. To obtain the shortest path that navigates through all viewpoints, the TSP can be solved. The TSP is a combinatorial optimization problem that seeks the shortest path visiting each waypoint exactly once and returning to the origin. It is classified as NP-hard, meaning that no known algorithm can solve it in polynomial time. Consequently, various heuristic algorithms have been developed to find near-optimal solutions. This literature review will primarily focus on the MTSP [27], a variation of the TSP that presents increased complexity while being reducible to the TSP. Unlike the conventional TSP, the MTSP typically involves multiple optimization objectives that often conflict with one another, adding to the problem's complexity. For example, one variation aims to minimize both the total traveling distance and the disparity in traveling distances among the salesmen, as some applications prefer a balanced workload. However, minimizing overall traveling distance may lead to assigning all waypoints to a single salesman, resulting in a significant workload imbalance. Therefore, the heuristic algorithm must search for solutions that balance these conflicting objectives.

The weighted sum technique [28] transforms the multi-objective (MO) problem into a mono-objective problem by assigning weighting factors to each objective. While this approach is straightforward, it results in a solution that represents only one specific combination of the weighting factors, limiting the diversity of possible solutions. Additionally, the design of parameters introduces further challenges. An alternative approach involves searching for the complete tradeoff curve for the optimal solution, known as the Pareto Optimal Front. This method provides users with the entire tradeoff curve, allowing them to select a specific solution that best suits their application, thereby enhancing flexibility in the decision-making process. The optimal tradeoff curve is characterized by a set of non-dominated solutions. A solution x is said to dominate solution y if both of the following conditions are satisfied [29]:

1. $Solution\ x\ is\ no\ worse\ than\ y\ in\ all\ objectives.$
2. $Solution\ x\ is\ strictly\ better\ than\ y\ in\ at\ least\ one\ objective.$

The Genetic Algorithm (GA) and its variants have been widely applied to solve the MTSP. In [30], the Non-dominated Sorting Genetic Algorithm II (NSGA-II) is employed to address the bi-objective MTSP. This algorithm utilizes a two-part chromosome representation to effectively capture the solutions, while a crossover operator based on a hardware-based genetic algorithm (HGA) is used during the global search process. This operator leverages both decoded and undecoded chromosomes to generate new solutions, enhancing the overall diversity of the solution set. Alves and Lopes [27] proposed two GAs to solve the bi-objective MTSP. Initially, a mono-objective approach is employed, adjusting the weighting factors to obtain different sections of the Pareto Optimal Front. Subsequently, the Strength Pareto Evolutionary Algorithm II (SPEA2) is applied for the MO-MTSP. Numerical studies indicate that the mono-objective GA achieves a more complete Pareto Optimal Front; however, additional parameter tuning is necessary for further optimization.

Lu et al. [31] proposed a two-step algorithm for solving the MTSP. This formulation aims to minimize both the total distance and the disparity between each salesman by dividing the MTSP into two subproblems. Initially, K-means clustering is used to group the waypoints, followed by solving the

TSPs within each group. While this approach provides an intuitive solution to the complex optimization problem, decoupling the objectives makes it challenging to adjust the weighting between them, limiting the algorithm's flexibility. To enhance the performance of GAs, other heuristic algorithms have been combined with the GA. In [34], Invasive Weed Optimization (IWO) is paired with the Partheno-Genetic Algorithm (PGA) to tackle the single-objective MTSP with multiple depots. The reproduction mechanism of the IWO leverages information from slightly worse solutions. Numerical studies demonstrate that this combined algorithm outperforms other intelligent algorithms. Zhou et al. [33] introduced an Improved Partheno-Genetic Algorithm (IPGA), a variation of the GA that does not utilize a crossover operator. This algorithm has been shown to outperform Particle Swarm Optimization (PSO).

In addition to GAs, ACO is a popular method for solving the MTSP. Dorigo and Gambardella [34] introduced the Ant Colony System (ACS) for addressing the single TSP. This algorithm employs local pheromone updates, which enhance performance and accelerate convergence. It also exhibits promising parallelization properties, further improving its effectiveness. Zhan et al. [35] applied ACO to solve the MTSP in multi-UAV search-and-rescue applications. The coverage area is first discretized according to flight constraints, and the single-objective MTSP is subsequently formulated and solved using ACO. The cooperative task allocation problem is tackled in [36] by formulating the MO-MTSP, aiming to minimize the overall distance traveled while balancing the workload among salesmen. The multi-objective ACS employs a probabilistic city allocation method to achieve this balance. Numerical studies indicate that this algorithm outperforms conventional NSGA-II and Multi-Objective Particle Swarm Optimization (MO-PSO) in terms of convergence rate and solution quality. To further enhance the convergence rate, an adaptive pheromone update strategy is proposed in [37]. A decomposition-based MO-ACO is employed to improve the diversity of the Pareto Front for the MO problem. Lastly, in [38], an MO-ACO with variable neighborhood descent is introduced to solve the bi-objective MTSP, resulting in improved solution quality and convergence rate.

While GAs and ACO are among the most popular algorithms for solving the MTSP, other methods have also demonstrated promising results. The modified Brainstorming Optimization (BSO) algorithm was introduced in [39], which mimics the human brainstorming process to address combinatorial optimization problems. Peng et al. [40] proposed a hybrid PSO algorithm, using PSO as the main framework while incorporating elements of GA. Additional approaches have been proposed to solve the MO-MTSP. References [41-42] address MTSP by subdividing the problem. A partition algorithm is first employed to divide the waypoints in a balanced manner, resulting in separate TSPs that are subsequently solved. Local search algorithms, such as the Lin-Kernighan algorithm [43] or tabu search, can be applied to further enhance the quality of the solutions.

Although GAs and ACO are popular heuristic algorithms for solving the MTSP, they both have limitations. For GAs, performance heavily relies on the quality of the initial solution, as all subsequent solutions are derived from it. A well-performing initial solution can accelerate convergence and lead to better outcomes. However, a poor initial solution can mislead the iterations, preventing the algorithm from generating a promising solution [44]. Thus, GAs can benefit from informed initialization using heuristic functions to capture promising solutions from the outset. In contrast, ACO constructs solutions based on pheromone information, which is reinforced at each iteration. This reliance on pheromone data can overwhelm the evolutionary process, potentially trapping the algorithm in local optima. Inaccurate parameter tuning can exacerbate this stagnation behavior [45]. Consequently, powerful local search algorithms are often employed alongside ACO to mitigate this issue. Various hybrid algorithms have been proposed to enhance the performance of both GAs and ACO.

One popular approach to improving ACO is through parameter optimization. Mahi et al. [46] proposed using PSO to optimize ACO parameters, incorporating the 3-opt algorithm to enhance the local search capability. Numerical studies indicate that this hybrid algorithm outperformed other algorithms in solving the TSP. To further simplify the parameter optimization process, [47] introduced the symbiotic organisms search (SOS) method, which requires no input parameters. Unlike the approach in [46], this method eliminates the need for additional parameters, and the results show that the hybrid algorithm can converge to better solutions. Lastly, [48] utilized GA to optimize ACO parameters, yielding promising results.

Apart from parameter optimization, other hybrid algorithms have been employed to enhance the performance of standalone algorithms. One popular approach to creating hybrid algorithms is to execute them sequentially, allowing both algorithms to contribute to the search and leverage their respective advantages. In [44], a hybrid algorithm combines ACO with PGA to solve the MTSP with multiple depots. This two-step algorithm first utilizes PGA to determine depot locations and city divisions, initializing the solutions. ACO is then executed to solve the TSP for each group of cities. This hybrid approach has been shown to outperform other evolutionary algorithms in solving the MTSP. A multi-step algorithm proposed in [49] involves the sequential execution of GA, ACO, and the 2-opt algorithm. Specifically, GA initializes the solution for ACO, which then performs the remaining search. The 2-opt algorithm is applied during ACO to enhance its local search capability. Two hybrid algorithms have been proposed in [50], combining GA and ACO to form two sequential algorithms: ACO-GA and GA-ACO. Their corresponding performances have been analyzed. A parallelization approach that integrates GA and ACO is presented in [51], achieving improved computational efficiency. In [52], the Dynamic Genetic Algorithm with Ant Colony Binary Iterative Optimization (DGA-ACBIO) is introduced to solve the TSP for UAV-related agricultural applications. The DGA is used in conjunction with the Ant Colony Binary Iterative Algorithm (ACBIO) sequentially. Simulation results indicate that this algorithm improves both searching efficiency and solution quality. Lastly, Dong et al. [53] proposed a hybrid algorithm based on parallelization, where GA and ACO are executed concurrently with information exchanged between them. This allows each algorithm to leverage its strengths while sharing benefits through information exchange. However, this parallel approach is computationally expensive.

While the sequential fusion approach offers a straightforward method for combining multiple algorithms, it has a significant limitation. By combining algorithms serially, the benefits of each algorithm cannot be fully utilized throughout the hybrid process. For instance, in a GA-ACO sequential hybrid algorithm, the global search capability of GA is only effective during the first half of the iterations, while heuristic information is utilized only in the latter half. To address this limitation, alternative approaches for creating hybrid algorithms have been proposed. Gao et al. [54] introduced a GA-ACO hybrid algorithm that employs an ACO-inspired crossover operator. At each iteration, solutions are generated using the state transition rule of ACO. During the crossover operation, paths with higher pheromone information are selected from the parent solutions to form the offspring solution.

By incorporating pheromone information, this approach helps preserve promising combinations. Additionally, the 2-opt algorithm is used to further enhance the local search capability of the algorithm. Dong and Ruan [55] employed a subtour library approach to combine GA and ACO. During the crossover and mutation operations, the pheromone intensity of each path is first computed. Paths with higher pheromone levels are retained, and genetic operations are performed accordingly, ensuring that these paths are not destroyed. This approach helps preserve the building blocks for the optimal solution. However, ACO is known for its high computational cost, as pheromone information is computed iteratively, requiring substantial memory to store this data. To address this issue, Chitty [56] proposed Partial-ACO, which reduces the computational load by calculating only part of the pheromone information during the crossover operation.

With the emergence of artificial intelligence, deep learning is becoming an increasingly popular approach to solving path planning problems. Zhang et al. [57] proposed a multi-agent deep reinforcement learning method to address the optimization problem by balancing exploration and exploitation throughout the iterations. Hu et al. [58] introduced a distributed cooperative deep Q-learning algorithm to achieve coverage paths while minimizing mission time, path overlaps, and collisions. Similarly, [59] developed a double Q-learning algorithm to generate coverage paths that optimize both distance and turning costs.

To summarize the literature review, several research gaps have been identified. First, existing literature primarily focuses on enhancing the efficiency of the inspection task while neglecting inspection quality. Although current inspection path planning techniques can meet coverage requirements, the quality of the inspection can be inconsistent. While the planning algorithm may generate paths that provide spatial coverage of the inspection target, the data collected may not meet the standards required by corresponding inspection techniques, potentially leading to ineffective defect detection. Therefore, there is a need for inspection path planning algorithms that prioritize improving inspection quality and enhancing defect detection to address this research gap.

Second, one common feature in existing MTSP formulations is that the depot location is predetermined and not included in the optimization process. However, it is important to recognize that the depot location also impacts the objective value. Existing literature often overlooks the significance of the depot location, highlighting a research gap that needs to be addressed. The depot location should be integrated into the objective function, and heuristic algorithms can be employed to search for near-optimal solutions.

3 Multi-Layer Path Planning for Complete Structural Inspection using UAV

This chapter focuses on enhancing the generalizability of inspection path planning algorithms for structural inspection using UAVs. The proposed algorithm emphasizes the scalability of the viewpoint generation process, offering practical solutions for various inspection distance requirements and eliminating the need for additional view planning procedures. Furthermore, the path planning algorithm aims to minimize path complexity by reducing overall turning angles, thereby smoothing the inspection process.

3.1 Notation list

The notations used in this section are defined as follows:

h	View cone height
D_{min}	Minimum viewing distance
θ	Field of view
l	Slicing interval
μ	Overlapping rate
$\boldsymbol{N}(x, y, z)$	Coordinates of the 3D model
$\boldsymbol{C}$	Camera matrix
$\boldsymbol{B}$	Viewpoint base
L	Perimeter
W	View cone width
$\boldsymbol{K}$	Intrinsic matrix
$\boldsymbol{R}$	Rotation matrix
t	Translation vector
x_{img}, y_{img}	Image plane coordinates

$x_{world}, y_{world}, z_{world}$	World coordinates
$\boldsymbol{M}$	Visibility matrix
$\boldsymbol{Surf}(x, y, z)$	Unscanned surface patches
$\overrightarrow{\boldsymbol{n_{sp}}}$	Normal vectors of unscanned surface patches
$\overrightarrow{\boldsymbol{n_{vp}}}$	Viewing direction
n_{rp}	Number of revolving viewpoints
n_{gf}	Number of gap filling viewpoints
n_n	Number of surface patches
x_j, p_{ij}, q_{ijk}	Decision variable
ε	Coverage requirement
w_1, w_2	Weighting factor
d_{ij}, d_{ijk}	Distance
a_{ijk}	Turning angle
N_G	Number of generation
N_P	Population size
N_L	Chromosome length
N_l	Number of layer
P_C	Crossover rate
P_M	Mutation rate
Φ	Objective balancing ratio
c_a	Turning angle cost
c_d	Distance cost

3.2 Motivation

As summarized in the literature review, existing inspection path planning algorithms are designed for specific inspection techniques. For instance, offsetting techniques are primarily suitable for close-up inspections, such as crack detection [60-63], while sampling techniques are typically used for distant inspections, such as reconstruction [64-66]. Since these algorithms are tailored to specific techniques, their scalability is limited. Although sampling techniques can be applied to close-up inspections, their performance is often degraded.

Consequently, different algorithms are required for various types of inspections, complicating the implementation of UAVs in inspection applications. Therefore, there is a need for an inspection path planning algorithm capable of generating suitable inspection paths for diverse purposes, simplifying UAV implementation. Additionally, execution difficulty must be considered; the inspection path should avoid extreme maneuvers that could challenge UAV mobility. To address this, kinodynamic constraints should be applied to smooth the inspection path, allowing commercial UAVs to perform inspections more effectively. The main contributions of this work are summarized as follows:

1. This work proposes a mixed viewpoint generation approach that creates a scalable viewpoint set suitable for various inspection purposes.
2. Additionally, this work introduces the ML-ADTSP, which generates kinodynamically feasible inspection paths to facilitate trajectory optimization in future processes while improving path smoothness.
3. Numerical simulations are conducted to extensively analyze the effects of incorporating kinodynamic constraints during preliminary path planning.

3.3 Problem Statement

This work considers the following scenario: Given a known 3D model of the inspection target and user-specific inspection distances, the goal is to generate a complete inspection path that the UAV can navigate to accomplish the inspection task. The objective is to unify different view-planning techniques, allowing suitable viewpoints to be generated for various inspection purposes using the proposed algorithm, thereby eliminating the need for additional planning algorithms. To achieve this, a mixed viewpoint generation approach is employed, consisting of uniform dilation and sampling. To filter out redundant viewpoints, the Set Cover Problem (SCP) is solved to select an optimal viewpoint set that meets the coverage requirements. The inspection path is then obtained by addressing the Traveling Salesman Problem (TSP). However, conventional TSPs focus solely on minimizing overall traveling distance, which may result in paths that are kinodynamically challenging for UAVs to execute. To address this, a bi-objective optimization problem, the Multi-layered Angle-Distance Traveling Salesman Problem (ML-ADTSP), is formulated. Specifically, the generated viewpoints are divided into different layers based on their altitude. The Angle-Distance TSP, which aims to minimize both traveling distance and turning angle, is solved locally within each layer. Finally, the path segments from each layer are connected to form a spiral inspection path, reducing traversal difficulty.

3.4 Methodology

The flowchart of the proposed planning algorithm is summarized in **Fig**. 3.1. Starting with the 3D model of the inspection target, a convex hull is applied to simplify the model by removing complex concave features. Slicing is then performed at various levels to extract the geometrical features of the target, followed by the generation of revolving viewpoints through uniform dilation. Next, the visibility of the generated viewpoints is computed. To ensure complete coverage of the inspection target, additional gap-filling viewpoints are created to supplement the revolving viewpoints. The SCP is then solved to minimize the total number of viewpoints used, and the complete inspection path is obtained by solving the proposed ML-ADTSP.

The proposed algorithm utilizes two sets of viewpoints to achieve a scalable viewpoint generation approach. The revolving viewpoints are generated based on the geometry of the target, providing uniform coverage and serving as the skeleton of the resulting inspection path. This limits the variation in viewing distance and improves the quality of the acquired data. In contrast, the gap-filling viewpoints offer unique coverage by deviating from the target's geometry, ensuring that the coverage requirements are met.

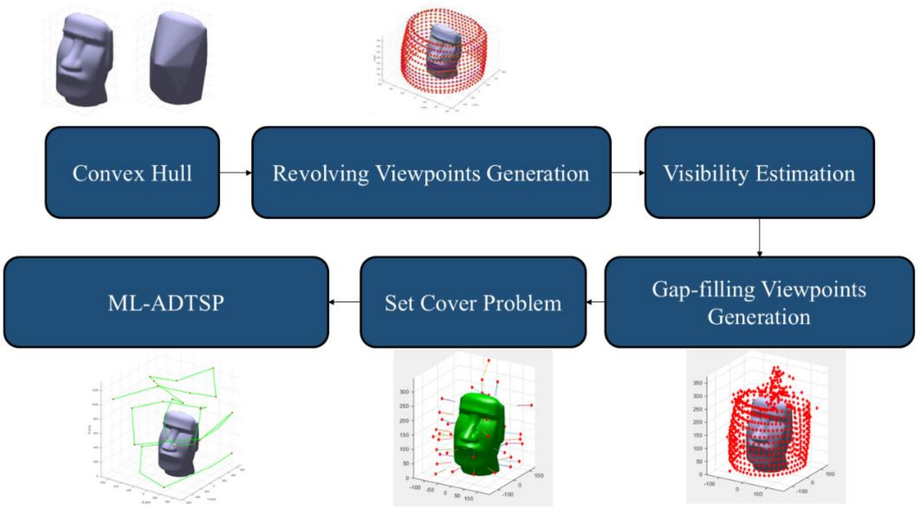


Fig. 3.1 Flow chart of the proposed algorithm.

3.4.1 Model Preprocessing

Before computing the revolving viewpoints, the geometry of the inspection target must be simplified. The 3D model may contain convex or concave features that could increase the risk of intrusion if the UAV closely follows the contours. Therefore, a convex hull is first applied to eliminate these features while preserving the overall geometry of the target, resulting in a smooth and featureless model.

3.4.2 Revolving Viewpoint Generation

With the processed model, the geometry information of the target can be extracted. The model is first divided into different layers based on altitude, and geometry information is obtained through a slicing procedure. To facilitate a more flexible slicing process, the slicing interval should be adjustable according to inspection requirements. A smaller slicing interval yields improved coverage, while a larger interval reduces the number of viewpoints. In the proposed method, the slicing interval is determined by the overlapping rate of the view plane. The view plane of a camera positioned D_{min} away from the inspection target can be illustrated in **Fig**. 3.2. With a field-of-view of θ, a camera placed D_{min} away will have a vertical viewing plane of $h = 2D_{min}\tan\left(\frac{\theta}{2}\right)$. If the slicing interval l equals to h, there will be no overlap. An overlapping rate μ is added to h such that $l = \frac{2}{\mu}D_{min}\tan\left(\frac{\theta}{2}\right)$. By adjusting μ, the slicing interval can be tailored to meet inspection requirements. After defining the slicing interval, geometry information of the inspection target at different levels is extracted by obtaining the intersections at each level. An additional 2D convex hull is applied to these intersections to create a polygon that encloses all points. This polygon is then dilated by the viewing distance to form an outline from which the revolving viewpoints will be generated. The slicing process is summarized in Algo. 1.

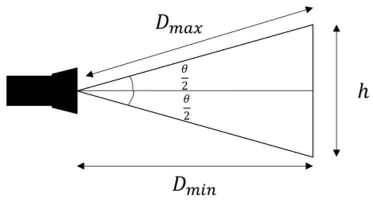


Fig. 3.2. Vertical view plane of a camera.

Algorithm 1 Viewpoint based generation

Input: Surface patches of the 3D model $N(x, y, z)$, camera matrix C, camera minimum viewing distance D_{min}, overlapping rate μ.

Output: Viewpoint base $\boldsymbol{B}$.

$\theta \leftarrow \mathrm{FOV}(C)$
$\Delta l \leftarrow \mathrm{FindInterval}(D_{min}, \mu, \theta)$
$l_1 \leftarrow \min_z(\boldsymbol{N}) + \Delta l$
while $l_i < \max_z(\boldsymbol{N})$ **do**
    $Int_i(x, y, z) \leftarrow \mathrm{Intersection}(\boldsymbol{N}, l_i)$
    $\boldsymbol{B}_i \leftarrow \mathrm{Convexhull}\big(Int_i(x, y, z)\big)$
    $\boldsymbol{B}_i \leftarrow \mathrm{Dilate}(D_{min}, \boldsymbol{B}_i)$
    $l_{i+1} \leftarrow l_i + \Delta l$
end while
return $\boldsymbol{B}$

After determining the vertical overlap, the horizontal overlap at each level should also be calculated. Similar to the vertical overlap, the horizontal overlap should be adjustable for more flexible path generation. To reduce the number of input parameters, the overlapping rate μ is used again to determine the number of viewpoints generated at each level. To simplify the calculation, a viewing cone is employed to estimate the vision of the camera, such that the viewing width w is set equal to the height of the viewing plane. If the perimeter of the viewpoint based $\boldsymbol{B}_i$ at level i is L, minimum horizontal overlap is achieved when the number of viewpoints is $n = \frac{L}{w}$. By substituting w with $\frac{2}{\mu} D_{min} \tan\left(\frac{\theta}{2}\right)$,

$n = \frac{\mu L}{2D_{min}\tan\left(\frac{\theta}{2}\right)}$ can be obtained. Thus, by increasing μ, both the vertical and horizontal overlap can be increased. Since the revolving viewpoints are generated along the dilated contour of the inspection target, collision checking is not required, which helps reduce overall computation time. As the revolving viewpoints aim to provide uniform coverage of the inspection target, all of the revolving viewpoints will be pointing toward the centroid of the inspection target at their corresponding altitude, computed using the convex hull of the inspection target.

Unlike the sampling approach, the revolving viewpoint generation process is deterministic. Since the subsequent viewpoint generation relies on these revolving viewpoints, a deterministic process provides a reliable foundation for further viewpoint generation, preventing the subsequent steps from being misguided. Additionally, this deterministic process offers consistent computation times, resulting in more predictable performance.

After generating the revolving viewpoints, their visibility information is evaluated using the pinhole camera model [67]. This model provides a simple yet realistic estimate of the vision of a high-definition camera. To compute the visibility, the camera's characteristics are first obtained, which can be expressed by the camera matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \boldsymbol{K} \times [\boldsymbol{R}|t] \tag{3.1}$$

where $\boldsymbol{K}$ is the camera intrinsic matrix which is obtained through camera calibration. It represents the parameters of the camera such as focal length and the location of the optical center. $[\boldsymbol{R}|t]$ is the camera extrinsic matrix that represents the camera's position and orientation in the 3D world. The location of the surface patches of the inspection target is represented by their centroid. The projection of the centroid onto the camera's image plane can be obtained through:

$$s\left[x_{img}\ y_{img}\ 1\right]^T = \boldsymbol{K} \times [\boldsymbol{R}|t] \times [x_{world}\ y_{world}\ z_{world}\ 1]^T \tag{3.2}$$

If the projection is within the size of the camera's sensor, the surface patch will be visible to the camera. To ensure the quality of the image captured, the viewing distance is limited. The surface patch must lie within the viewing cone of the camera, satisfying $D_{min} \leq D \leq D_{max}$. To further enhance the quality of the images, the viewing angle δ is also limited. Ideally, the camera should be perpendicular to the surface patch to minimize distortion. If the viewing angle is too large, the image may exhibit significant distortion, making it difficult to clearly observe the surface. If the centroid satisfies all these constraints, it will be considered visible to the camera, and the visibility information will be recorded in the visibility matrix $\boldsymbol{M}$, which is based on the original model of the inspection target.

3.4.3 Gap-fill Viewpoint Generation

While the revolving viewpoints provide uniform coverage of the inspection target, their coverage is limited. Specifically, uniformly distributed viewpoints may not adequately cover concave areas, as they are computed using the convex hull of the inspection target. Therefore, an additional set of viewpoints is needed to supplement the revolving viewpoints. The gap-filling viewpoints are designed to provide unique coverage, particularly for areas that the revolving viewpoints cannot reach. After evaluating the visibility of the revolving viewpoints, the unscanned areas are identified by locating zero elements in the visibility matrix. Using the locations of the unscanned surface patches and their surface normals, candidate viewpoints are generated through offsetting. The offset coordinates represent the locations of the gap-filling viewpoints, while the opposite vector of the surface normal indicates the corresponding viewing direction. However, for complex inspection targets, a simple offset approach may lead to undesirable viewpoint locations that could cause collisions. Therefore, randomness is introduced into the dilation process. Random noise is added to the offsetting vector, ensuring that the candidate viewpoints are not directly perpendicular to the surface patch while still covering the intended surfaces. This process is repeated for each unscanned surface. The visibility information of the gap-filling viewpoints is then computed, retaining only those viewpoints that provide additional coverage and are collision-free. Since the proposed planning algorithm aims to generate suitable viewpoints for various inspection methods, the coverage requirements of these methods are also considered in the planning process. For inspection techniques such as photogrammetry, multiple overlaps are necessary to provide sufficient features for the feature matching process. Consequently, additional viewpoints are generated for surface patches that do not meet the coverage requirements. The generation of gap-filling viewpoints continues until the visibility for each surface patch satisfies the coverage criteria. Algo. 2 summarizes the gap-filling viewpoint generation process.

Algorithm 2 Gap-filling viewpoint generation

Input: Unscanned surface patches $\boldsymbol{Surf}(x, y, z)$, surface patches of 3D model $\boldsymbol{N}(x, y, z)$, minimum viewing distance D_{min}, normal vectors of unscanned surface patches $\overrightarrow{\boldsymbol{n_{sp}}}$

Output: Gap-filling viewpoints $\boldsymbol{G_{vp}}(x, y, z)$, viewing direction $\overrightarrow{\boldsymbol{n_{vp}}}$

for $s_i \in \boldsymbol{Surf}$ **do**
    while $M_i < \varepsilon$ **do**
        while true **do**
            $G_{vp_i} \leftarrow \text{RandomSampling}(D_{min}, s_i, \overrightarrow{\boldsymbol{n_{sp_i}}})$
            $\overrightarrow{\boldsymbol{n_{vp_i}}} \leftarrow \text{OppoVector}(\overrightarrow{\boldsymbol{n_{sp_i}}})$
            if $\text{VistEst}(G_{vp_i}, \overrightarrow{\boldsymbol{n_{vp_i}}}, \boldsymbol{N})$ and $\text{ColliCheck}(G_{vp_i})$ **then**
                $\boldsymbol{G_{vp}} \leftarrow \text{Append}(\boldsymbol{G_{vp}}, G_{vp_i}, \overrightarrow{\boldsymbol{n_{vp_i}}})$
                $M_i = M_i + 1$
                break
            end if
        end while
    end while
end for
return $\boldsymbol{G_{vp}}, \overrightarrow{\boldsymbol{n_{vp}}}$

The generation process can be generalized as a guided random search. Consequently, it is not deterministic, meaning that different viewpoints may be generated in different executions. The computational complexity is proportional to the complexity of the inspection target, as more time is required to generate executable viewpoints for a complex structure.

3.4.4 Set Cover Problem

After generating the gap-filling viewpoints, the coverage requirements must be met. To eliminate redundant viewpoints, the SCP is solved. The SCP aims to obtain the minimum number of viewpoints while satisfying these coverage requirements. The integer programming formulation of the problem can be expressed as follows:

$$\min \sum_{j=1}^{n} x_j,$$

$$\text{s.t. for } V_i \in V, \sum_{j=1}^{n} M_{ij} \geq \varepsilon, \tag{3.3}$$

$$\text{where } x_j \in 0,1$$

The objective function aims to minimize the number of selected viewpoints, while the constraints ensure that the coverage requirements are satisfied. Specifically, every surface of the inspection target should be covered ε times. The final constraint ensures the integrality of the problem.

3.4.5 Multi-layered Angle-Distance Traveling Salesman Problem

After obtaining the optimal set of viewpoints, a complete inspection path must be established to connect these viewpoints. The most straightforward approach is to formulate and solve the TSP. However, TSP focuses solely on minimizing traveling distance and ignores other factors that may impact the performance of the inspection task. For instance, the overall turning angles of the inspection path significantly affects its smoothness. Large turning angles can result in a jittery path, challenging the mobility of the UAV. Sudden turns, such as a 180° turn in the inspection path, require the UAV to come to a complete stop before changing its heading. Such maneuvers necessitate a sharp change in acceleration, challenging the UAV's maneuverability. As a result, the flight path may become less stable, with the UAV potentially overshooting and struggling to follow the designed path closely. Additionally, rapid altitude changes should be avoided, as vertical movement consumes more energy compared to horizontal movement, as pointed out in [68-70]. As a result, the resulting inspection path should be a continuous spiral that allows for smooth altitude transitions. To address these considerations, the Multi-layered Angle-Distance Traveling Salesman Problem (ML-ADTSP) is proposed to generate a continuous spiral inspection path, enhancing both vertical and horizontal smoothness.

The ML-ADTSP can be divided into three steps. First, the viewpoints are categorized into different layers based on their altitude, limiting the altitude change for each path segment. Second, the Open Loop Angle-Distance Traveling Salesman Problem (OL-ADTSP) is solved locally within each layer.

By grouping the viewpoints into smaller sets, the number of viewpoints for each optimization problem is reduced, which in turn decreases the computation time. Lastly, after obtaining the open loop path segments, they are connected to create a continuous spiral inspection path.

To create the inspection path segment at each layer, the OL-ADTSP must be solved. The optimization problem aims to minimize both the traveling distance and the turning angle of the path segment. The integer linear programming formulation of the problem can be expressed as follows:

$$\min w_1 \sum_{i=1}^{n} \sum_{j=1,\, j\neq i}^{n} d_{ij}\, p_{ij} + w_2 \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{i\neq j\neq k, k=1}^{n} a_{ijk} q_{ijk} \tag{3.4}$$

$$\sum_{i=1, j\neq i}^{n} p_{ij} = 1, \quad \forall j \in V \tag{3.5}$$

$$\sum_{j=1, i\neq j}^{n} p_{ij} = 1, \quad \forall \mathrm{i} \in V \tag{3.6}$$

$$\sum_{i\neq j \in S} p_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset \tag{3.7}$$

$$p_{ij} \in 0,1 \tag{3.8}$$

$$q_{ijk} \in 0,1 \tag{3.9}$$

where w_1 and w_2 are the weights for the combined distance and turning angle cost, respectively. d_{ij} is the distance between viewpoint i and j while a_{ijk} is the turning angle between three viewpoints, which can be calculated by:

$$a_{ijk} = \arccos\left(\frac{p_j - p_i}{|p_j - p_i|} \cdot \frac{p_k - p_j}{|p_k - p_j|}\right) \tag{3.10}$$

The turning angle considered is the yaw angle of the UAV, which is independent of the camera's yaw, as the camera is controlled independently by a gimbal. Constraints (3.5) and (3.6) ensure that each viewpoint is visited only once. Constraint (3.7) is the subtour elimination constraint, which guarantees that a complete tour is obtained. Constraints (3.8) and (3.9) ensure the integrality of the problem.

However, since the traveling distance is computed using two consecutive viewpoints, while the turning angle uses three, two variables are needed for the optimization problem. To simplify the formulation, q_{ijk} can be used to replace p_{ij}, allowing the objective function to be reduced to:

$$\min w_1 \sum_{i=1}^{n} \sum_{i \neq j, j=1}^{n} \sum_{i \neq j \neq k, k=1}^{n} d_{ijk} q_{ijk} + w_2 \sum_{i=1}^{n} \sum_{i \neq j, j=1}^{n} \sum_{i \neq j \neq k, k=1}^{n} a_{ijk} q_{ijk} \tag{3.11}$$

$$\text{subject to} \quad p_{ij} = \sum_{(i,j,k) \in \mathrm{V}} q_{ijk} = \sum_{(k,i,j) \in V} q_{kij}, \qquad (i,j) \in A \tag{3.12}$$

Constraint (3.12) limits that for a selected arc$(i,j) \in A$, there must exist a viewpoint $k \in V$ such that the path (i,j,k) is included in the resulting inspection path. Additionally, there must be another viewpoint $k' \in V$ such that the path (k',i,j) is also included in the inspection path. As proven in [71], this ADTSP model is a formulation for the Quadratic Traveling Salesman Problem (QTSP).

Since the formulated problem is NP-hard, a heuristic algorithm is employed to search for a near-optimal solution. Specifically, the Genetic Algorithm (GA) [72] is selected. As mentioned in the literature review, GA offers fast computation times while providing flexibility in algorithm design. The genetic operators are modular and can be adapted according to specific requirements. In addition to commonly used genetic operators such as crossover and mutation, the GA used for solving the ADTSP also incorporates elitism, which preserves the best-performing solutions and prevents them from being lost during subsequent genetic operations. The flowchart of the search process is illustrated in **Fig**. 3.3. The algorithm begins by randomly generating a set of solutions, which are then evaluated for performance. Elitism is applied to store the currently best-found solutions to protect them from being degraded by later genetic operations. The crossover operation is executed based on the probability P_c, aiming to combine the useful features of parent solutions to create superior offspring. The mutation operation, serving as the local search component, is executed based on the probability P_M. The iteration continues until the maximum number of generations N_G is reached.

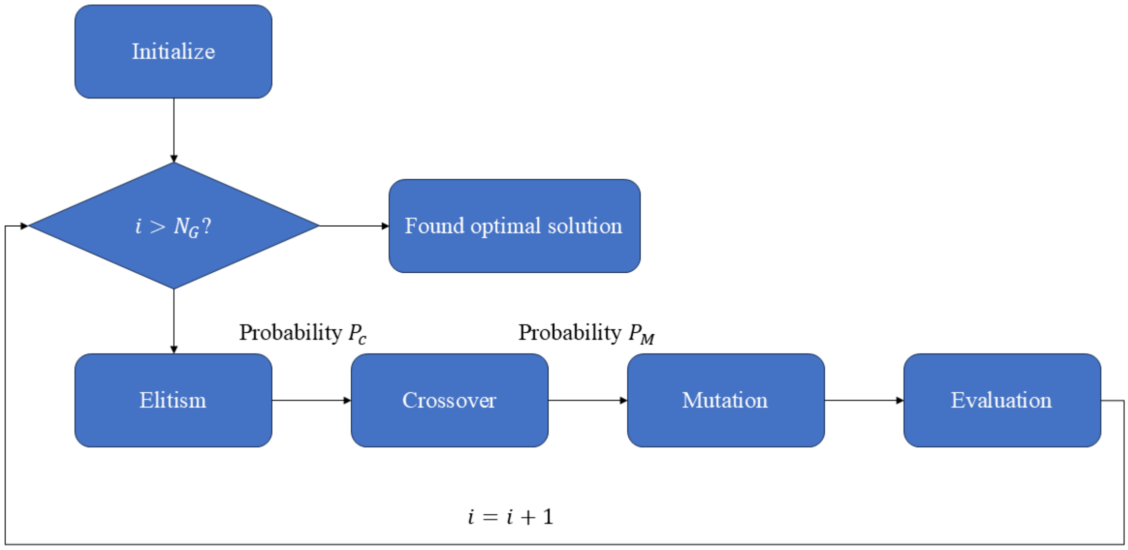


Fig. 3.3. Flow chart of the proposed GA.

Before the crossover operation, candidate solutions must be selected from the population pool. To ensure a strong selection bias toward promising solutions, roulette wheel selection is employed. In this selection process, candidate solutions with greater fitness values have a higher probability of being chosen, ensuring the quality of the parent solutions and preventing subsequent generations from being misled.

After selecting the appropriate parent solutions, the crossover operation is performed. This operation is a crucial component of the Genetic Algorithm (GA) as it contributes to the algorithm's evolution process. In [73], various genetic operators used to solve the TSP are discussed, concluding that the Edge Recombination (ER) and Order Crossover (OX) provide the best performance within a reasonable computation time. ER generates offspring by selecting the common features of the parent solutions, ensuring those features are passed on to the child solution. Similarly, OX achieves this by preserving a section of the parent solution and transferring it to the child. While both methods facilitate the evolution process, they do not consider the actual performance of each combination, relying only on estimates of promising features. Thus, these operations do not guarantee improvement.

To accurately identify promising features within the parent solutions, their corresponding fitness must be evaluated. Therefore, the Best Fit Crossover (BF) is proposed in this work. BF first selects a random viewpoint as the starting point, then searches for potential connections from the parent solutions.

Different combinations are evaluated, and the combination with the highest fitness value is selected. By assessing the fitness of various combinations, promising features can be correctly identified. Additionally, BF enables the exploration of other genetic features that could enhance solution quality. A numerical study comparing the performance of the three crossover operators shows that BF outperformed both OX and ER. The results are summarized in Table 3.1.

Table 3.1 Performance comparison of different crossover operators.

	OX	ER	BF
Best tour length (m)	129	125	125
Average tour length (m)	184	150	150
Generations needed	/	26	13

The optimal solution for the problems used in the numerical study is 150 m, and only the Edge Recombination (ER) and Best Fit (BF) crossover methods are able to converge to it. While both ER and BF reach the optimal solution, BF demonstrates a superior convergence rate, achieving the solution in fewer generations. Consequently, BF outperforms both Order Crossover (OX) and ER.

To prevent the algorithm from becoming trapped in local optima, mutation operations are employed to enhance the local search capability of the Genetic Algorithm (GA). To ensure a thorough local search, three mutation functions are utilized in this work, and their corresponding details are summarized as follows:

1) Swap mutation: Swap the positions of two randomly selected viewpoints to eliminate potential path crossings in the solution.
2) Scramble mutation: Randomize the sequence of viewpoints, except for a selected section, which helps explore previously unsearched areas of the solution space.
3) Inverse mutation: Invert the sequence of a selected section to remove potential path crossings.

For the proposed multi-objective TSP, it is essential to determine the weighting of each objective, as it will affect the performance of the resulting inspection path. If $w_1 = 1$ and $w_2 = 0$, the convectional Euclidean TSP is obtained. Conversely, if $w_1 = 0$ and $w_2 = 1$, the angle TSP [74] can be obtained, where the objective is to minimize only the total turning angle. Therefore, there should exist a ratio Φ

that allows the objectives can be balanced and be minimized simultaneously. The ratio Φ can be represented as:

$$\Phi = \frac{w_2 c_a}{w_1 c_d + w_2 c_a} \tag{3.13}$$

where c_a and c_d are the angle and distance cost respectively. Numerical study shows that a ratio of 50-70% allows for a more balanced objective.

Before discussing the next procedure, the time complexity in solving the ADTSP is discussed. The time complexity of solving the ADTSP depends on the designed GA. It can be represented as $O(N_G \times N_P \times s)$, where s is a constant representing the time required for fitness evaluation, crossover, and mutation operations. In the proposed approach, the viewpoints are divided into different layers, allowing the ADTSP to be solved locally. This division reduces the complexity of each subproblem, thereby decreasing the overall time required to compute the complete inspection path.

After solving the ADTSP at each layer, the path segments are connected to form the resulting inspection path. As the objective of the ADTSP is to minimize the traveling distance and the total turning angle, the inter-layer connection should also align with this objective. **Figure** 3.4 illustrates the scenario of two path segments ab and cd, and there are four possible connections between the two segments. The traveling costs of all combinations are calculated, and the combination with the lowest cost is selected.

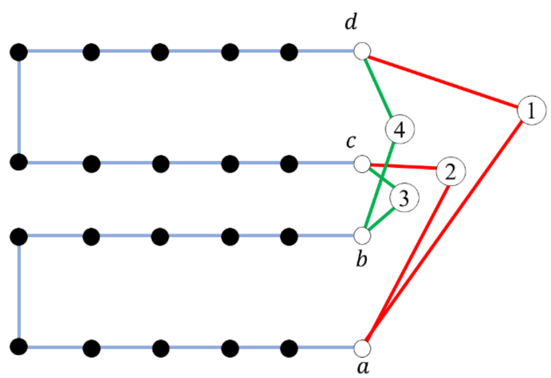
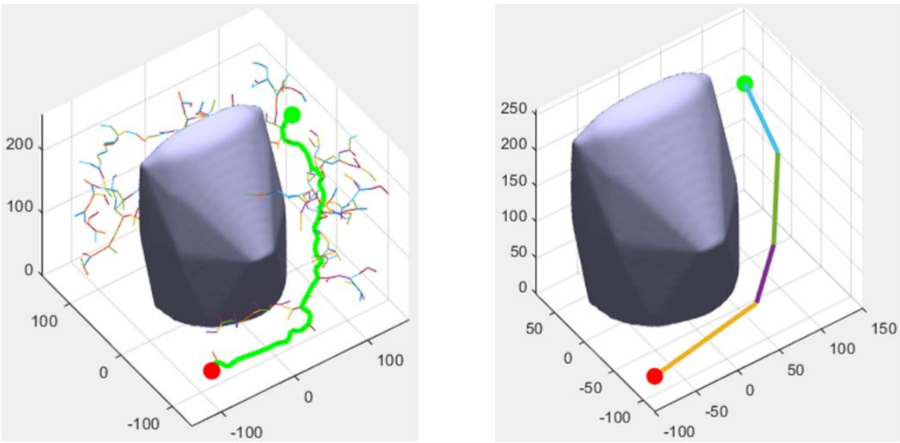


Fig. 3.4. Inter-layer connections between two path segments.

The ADTSP focuses solely on minimizing the optimization objective, neglecting other aspects of the inspection path. Consequently, the inspection path obtained through the ADTSP may contain infeasible segments, necessitating additional collision checking. After generating the inspection path, collision checking is performed to identify these segments. To obtain a collision-free path, the Rapidly Exploring Random Tree (RRT) algorithm is utilized. Although the solutions obtained by RRT may not be optimal compared to other planning algorithms, such as A*, RRT offers greater versatility and robustness. As a sampling-based local path planning algorithm, RRT does not require pre-discretization of the planning area, allowing for easy application to various geometries. However, the RRT path may exhibit rapidly changing headings and redundant waypoints, resulting in a jittery path, contradicting the optimization objectives. To address this, a path simplification module is introduced to smooth the updated path. Using the RRT path, the algorithm begins by selecting the starting waypoint. It then performs collision checking on all possible combinations with the starting point. The furthest point that can be reached without collision is selected as the next point, while all intermediate waypoints are discarded. This iteration continues until the destination point is reached. **Figure** 3.5 illustrates the effect of the RRT path smoothing approach, showing a significant reduction in the number of waypoints and a smoother resulting path compared to the original.



(a) (b)

Fig. 3.5. RRT path simplification. (a) Original Path, (b) Simplified path.

3.4.6 Time complexity

The proposed inspection path planning algorithm consists of several key components: viewpoint generation, visibility estimation, the SCP, and the ML-ADTSP. Among these, visibility estimation and the ML-ADTSP are the most computationally intensive. For visibility estimation, the time complexity can be expressed as $O(n_{rp}n_n + n_{gf}n_n)$, representing the sum of the estimation processes for both revolving and gap-filling viewpoints. Visibility information is computed for each viewpoint on every surface patch of the inspection target, represented by n_n. The ML-ADTSP is solved using GA with a time complexity of $O(N_G N_P N_L)$. Since the ML-ADTSP is solved at each layer, the time complexity for the overall path planning process is $O(N_l N_G N_P N_L)$, where N_l is the number of layers. Thus, the overall time complexity of the proposed algorithm can be summarized as $O(n_{rp}n_n + n_{gf}n_n + N_l N_G N_P N_L)$.

3.5 Simulation Results and Discussion

Numerical simulations are conducted to evaluate the performance of the proposed method. The scalability of the approach is first examined. To demonstrate the capability of the proposed method in generating suitable inspection paths for various inspection purposes, models of different sizes are selected as inspection targets, and different inspection techniques are simulated. Model 1 is the Bank of China Tower in Hong Kong, which has a height of 367 m, and Model 2 is Big Ben, with a height of 96 m. Both crack detection and photogrammetry techniques are simulated on these inspection targets. The simulations mimic the performance of the DJI Phantom 4 Pro V2, and the parameters of the onboard camera are obtained through camera calibration using a checkerboard pattern. In the simulation environment, the inspection target is assumed to be located in an open space, thus ignoring potential collisions with other structures. The onboard camera has three degrees of freedom, and its orientation is independent of the UAV, as it is controlled by a gimbal. The corresponding simulation configurations are summarized in Table 3.2, while the simulated performances are detailed in Table 3.4 and illustrated in **Fig**. 3.6. In addition, the parameters used in the GA for solving the ADTSP are summarized in Table 3.3.

Table 3.2 Simulation configurations.

Inspection Method	Crack Detection	Photogrammetry
Max. viewing distance (m)	[10, 15]	[20, 25]
Field of view	84°	84°
Overlapping rate, μ	15	20
Viewing angle, δ	60°	60°
Coverage requirement, ε	1	3

Table 3.3 Genetic Algorithm Configurations.

Parameters	Value
Number of generations, N_G	500
Population size, N_P	300
Elitism Rate, P_E	0.05
Crossover Rate, P_C	0.7
Mutation Rate, P_M	0.3

Table 3.4 Simulation Results.

	Number of viewpoints		Traveling distance (m)	
Inspection Method	Model 1	Model 2	Model 1	Model 2
Crack Detection	290	87	5996	2202
Photogrammetry	87	47	3676	1768

The simulation results demonstrate the proposed method's effectiveness in generating suitable inspection paths for various inspection purposes. For the Bank of China Tower, which features complex concave shapes, the uniformly distributed revolving viewpoints are insufficient to cover all areas. Consequently, additional gap-filling viewpoints are required, leading to a looser inspection path. In contrast, Model 2, with its more uniform structure, benefits from the revolving viewpoints, providing better coverage and requiring fewer gap-filling viewpoints. If only the revolving viewpoints are used, the resulting inspection path may not offer sufficient coverage to meet inspection quality standards. Conversely, relying solely on gap-filling viewpoints may produce an erratic inspection path with poor quality. Thus, the results highlight the importance of employing a mixed viewpoints generation technique to ensure adequate coverage. This proposed method allows the coverage requirements of different inspection techniques to be met while maintaining high information quality.

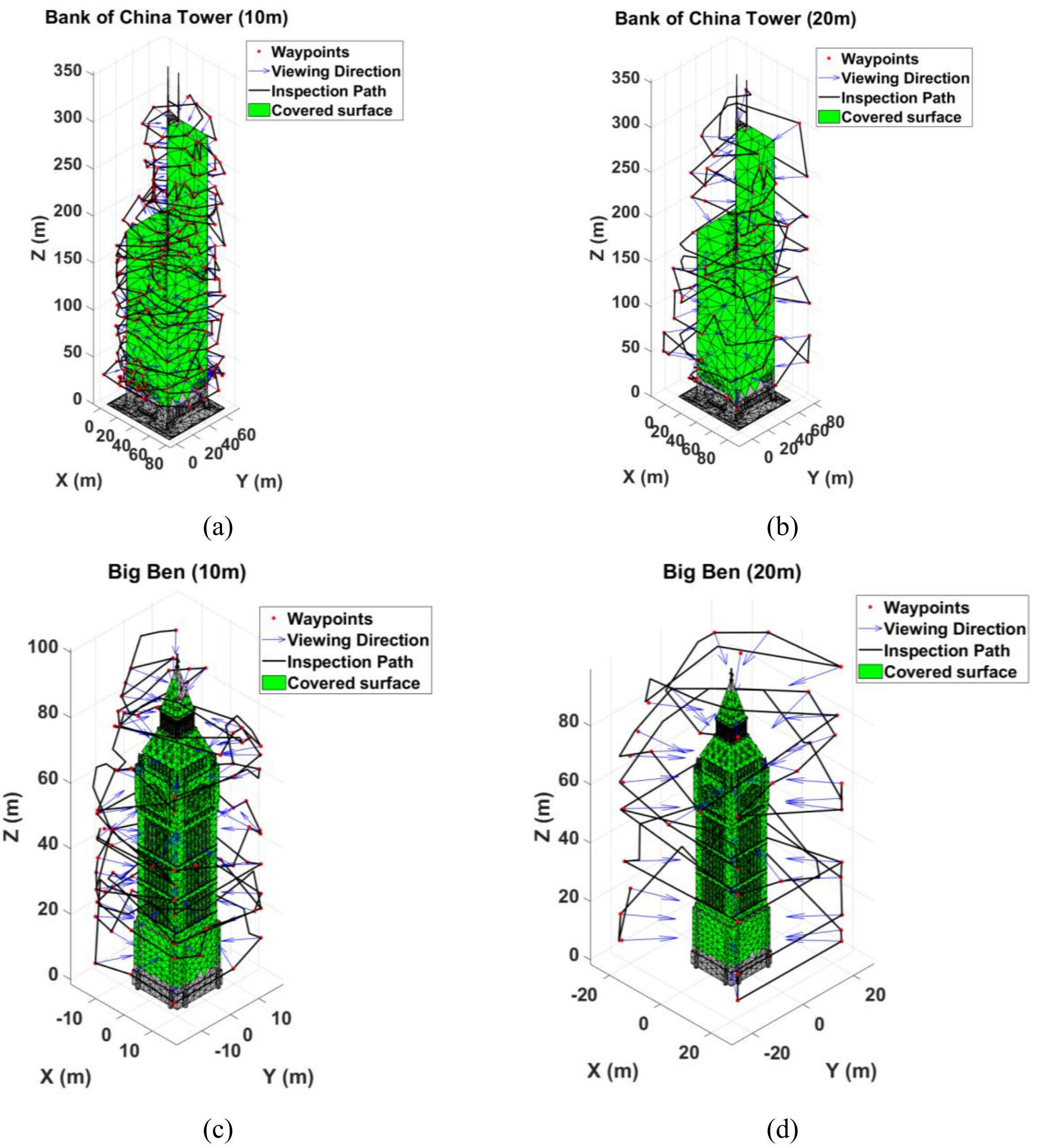
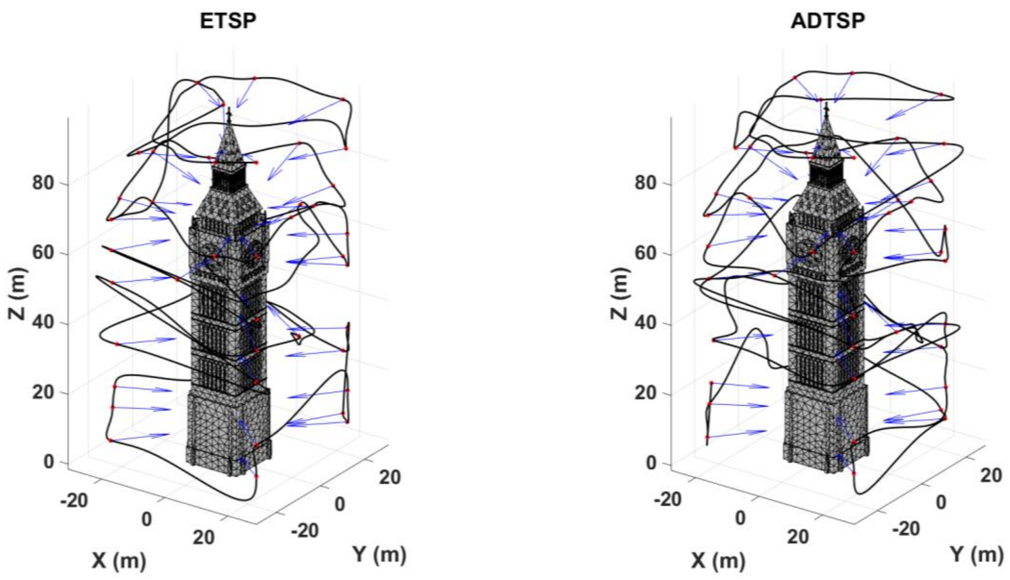


Fig. 3.6. Generated inspection paths. (a) Crack detection for Model 1. (b) Photogrammetry for Model 1. (c) Crack detection for Model 2. (d) Photogrammetry for Model 2.

The effects of the ADTSP are then studied. To study the effects of path smoothening, the performances of the resulting minimum snap trajectory are simulated. The ADTSP is also compared to the ETSP to demonstrate the kinodynamic improvement. The scenario of photogrammetry on the Big Ben is used for numerical simulation, with the UAV executing the inspection mission at a targeted velocity of 1 ms^{-1}. The simulation results are summarized in Table 3.4, **Fig**. 3.7 and **Fig**. 3.8.

Table 3.4 Simulated results of the minimum snap trajectories.

	ML-ETSP	ML-ADTSP	% Change
Mission Time (s)	**1442**	1768	+22%
Mean Turning Angle (rad)	1.48	**1.28**	-15%
Mean x-acceleration (ms^{-2})	0.06	**0.055**	-9%
Mean y-acceleration (ms^{-2})	0.051	**0.05**	-2%
Mean z-acceleration (ms^{-2})	0.058	**0.057**	-1.7%



(a) (b)

Fig. 3.7. Simulated trajectories. (a) ETSP. (b) ADTSP.

As the traveling distance and turning angles are conflicting objectives, a trade-off is observed in the simulated results. Compared to the ETSP, which solely aims to minimize traveling distance, the ADTSP results in an increased traveling distance of 326 m. However, while the traveling distance increases, the average turning angles are significantly reduced. The ADTSP decreases the average turning angle by 15%, effectively eliminating sharp turns in the inspection path. By minimizing extreme maneuvers, the UAV can maintain a more consistent velocity, reducing the need for sudden acceleration. It is evident that the ETSP exhibits higher mean acceleration in all directions, which poses additional challenges during execution. Consequently, by minimizing the average turning angle, the UAV can navigate through viewpoints more comfortably, reducing traversal difficulty. **Figure** 3.8 reveals that the ADTSP provides significant improvements in areas with high viewpoint density. In these locations, the ETSP tends to connect nearby viewpoints without considering the implications of sharp turns. This results in an inspection path with abrupt turns, generating sudden and large accelerations, increasing traversal difficulty. While dense viewpoint distribution is often unavoidable, especially in close-up inspection scenarios, the ADTSP is essential for enhancing traversal smoothness.

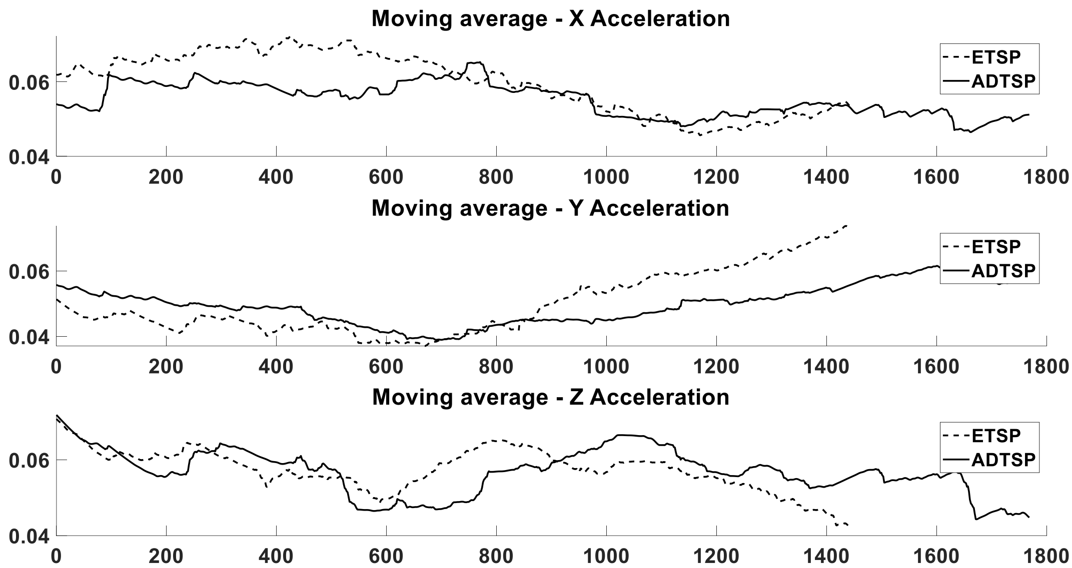


Fig. 3.8 Comparison of the simulated accelerations.

The numerical simulations were conducted at a relatively low traversal speed, and significant improvements in terms of path smoothness were observed. In time-constrained inspection scenarios where the UAV must operate at higher speeds, the impact of sharp turns becomes even more pronounced. This further emphasizes the importance of the ADTSP in such contexts. While the computed inspection path may require additional low-level planning before execution, the simulations highlight the necessity of considering kinodynamic constraints during the preliminary planning process.

3.6 Conclusion

This work addresses the scalability aspect of inspection path planning. The proposed method utilizes two sets of viewpoints to generate suitable inspection paths for various inspection purposes. To enhance the smoothness of the inspection path, the ML-ADTSP is introduced. The generated viewpoints are first divided into multiple layers based on their altitude. The ADTSP, which aims to minimize both traveling distance and turning angle, is then solved locally at each layer using GA. These path segments are subsequently connected to form a continuous spiral inspection path. Numerical simulations demonstrate that the ADTSP significantly reduces traversal difficulties, facilitating the execution of the inspection mission.

4 Migratory Genetic Algorithm with Pheromone Information for Autonomous Coverage Path Planning using UAVs

In the previous chapter, the scalability of the viewpoint planning process for autonomous inspection using UAVs was discussed. However, the efficiency of the task is limited since only one UAV is deployed. While scalability is a crucial aspect of the application, task efficiency is equally important. In this chapter, a multi-agent approach is proposed to enhance the efficiency of the autonomous inspection task. By utilizing multiple UAVs, different sections of the target can be inspected simultaneously, significantly reducing the overall inspection time.

4.1 List of notations

The notations used in this section are defined as follows:

$\boldsymbol{F}$	Combined objectives
f_1	Combined traveling distance
f_2	Maximum traveling distance
m	Number of UAVs
n	Number of waypoints
$x_{kij}, o_{kip}, o_{kpj}$	Decision variables
d_{ij}, d_{ip}, d_{pj}	Distance
V	Waypoint set
p	Depot
P	Population size
G_1	Number of generations of the NSGACO
G_2	Number of generations of the migration process
L	Length of the chromosome
r_c	Crossover rate
r_m	Mutation rate

α	Relative importance of pheromone trail
β	Relative importance of heuristic function
ρ	Pheromone evaporation rate
Q	Pheromone update constant
$\boldsymbol{P}$	Pheromone matrix
$\boldsymbol{D}$	Heuristic matrix
$\tau(i,j)$	Pheromone trail
$\eta(i,j)$	Heuristic function
C	Unvisited waypoints
$q, p(i,j)$	Probability
F	Objective value
λ	Pheromone balancing factor

4.2 Motivation

In a single-agent scenario, obtaining the inspection path requires solving the Traveling Salesman Problem (TSP). This combinatorial optimization problem can be generalized to the Multiple Traveling Salesman Problem (MTSP), where the optimal paths for all m agents must be determined. Like the TSP, the MTSP is also NP-hard, necessitating heuristic approaches to find near-optimal solutions. In the multi-agent inspection scenario, the primary objective is to reduce overall inspection time, which can be interpreted as minimizing both the total traveling distance and the maximum distance traveled by any agent. Therefore, a multi-objective (MO) variant of the MTSP needs to be solved. Additionally, in realistic scenarios, determining the dispatch location is crucial, as it significantly affects overall travel distance. Thus, an additional heuristic approach is required to address this extended problem.

This paper proposed an MTSP with an unknown depot. While the MTSP is a well-discussed problem in the literature, the discussion on depot optimization remains limited. In the formulated problem, it considers the problem of depot optimization, and a migration system that utilizes pheromone intensity is proposed to search for the optimal depot location that can improve the deployment efficiency of the inspection system.

To solve the formulated optimization problem, a hybrid algorithm consists of Genetic Algorithm (GA) and Ant Colony Optimization (ACO) is proposed. The proposed algorithm leverages the benefits of improved global search capability from the GA and the use of heuristic information from the ACO to achieve the improvement in both solution quality and diversity, and the performance of the algorithm is validated through numerical studies.

The main contributions of this work are as follows:

1) Heuristic Information Design: This work develops unique heuristic information that facilitates system migration and aids in the search for the optimal depot location.
2) Hybridization Approach: A hybridization of Genetic Algorithms (GA) and Ant Colony Optimization (ACO) is proposed, enabling the benefits of both algorithms to be leveraged during the search process while minimizing computational workload.
3) Performance Validation: The proposed algorithm's performance is validated using benchmark problems, demonstrating improvements in both solution quality and diversity.

4.3 Preliminary

In this section, the problem formulation and background knowledge on GA and ACO will be discussed.

4.3.1 Problem formulation

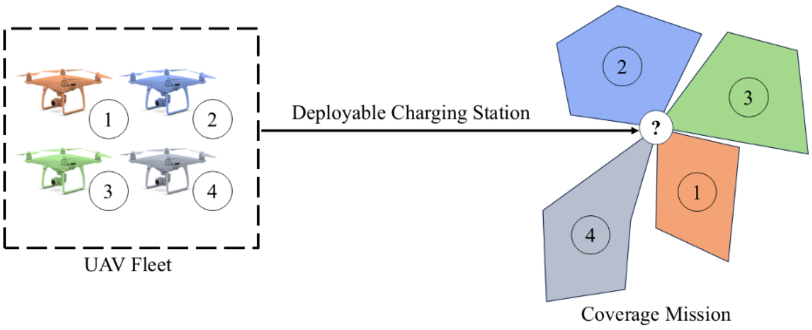


Fig. 4.1 Illustration of the inspection mission with 4 UAVs.

The scenario considered in this work is illustrated in **Fig.** 4.1. A group of m UAVs and a deployable charging station are utilized to create an autonomous system for a repeatable inspection mission. The inspection mission consists of a set of n viewpoints generated prior to the path planning task, which will be assigned to different UAVs. Before assigning the viewpoints, the location of the charging station must be determined. For simplicity, the charging station's location is selected from the points of interest, with selection criteria aligned to the optimization objectives. After deploying the charging station, the UAVs will be dispatched from it to travel through their assigned viewpoints. Upon completing their missions, the UAVs will return to the charging station. This scenario can be represented by the MTSP with an unknown depot. The corresponding integer linear programming problem can be formulated as follows:

$$\min \boldsymbol{F} = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x})] \tag{4.1}$$

$$f_1(\boldsymbol{x}) = \sum_{k=1}^{m} \sum_{i=1, i\neq p}^{n} \sum_{j=1, j\neq p}^{n} x_{kij} d_{ij} + \sum_{k=1}^{m} \sum_{i=1}^{n} o_{kip} d_{ip} + \sum_{k=1}^{m} \sum_{j=1}^{n} o_{kpj} d_{pj} \tag{4.2}$$

$$f_2(\boldsymbol{x}) = \max_{1\leq k\leq m} \left(\sum_{i=1, i\neq p}^{n} \sum_{j=1, j\neq p}^{n} x_{kij} d_{ij} + \sum_{i=1}^{n} o_{kip} d_{ip} + \sum_{j=1}^{n} o_{kpj} d_{pj} \right) \tag{4.3}$$

$$\text{s.t.} \sum_{k=1}^{m} \sum_{j=1}^{n} x_{kij} = 1, \quad \forall i \in V \setminus \{p\} \tag{4.4}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{n} x_{kij} = 1, \quad \forall j \in V \setminus \{p\} \tag{4.5}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{n} o_{kip} = m, \quad \exists!\, p \in V \tag{4.6}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n} o_{kpj} = m, \quad \exists!\, p \in V \tag{4.7}$$

$$\sum_{k=1}^{m}\sum_{i=1,i\neq p}^{n}\sum_{j=1,j\neq p}^{n} x_{kij} = n \tag{4.8}$$

$$x_{kij}, o_{kip}, o_{kpj} \in \{0,1\} \tag{4.9}$$

The optimization problem aims to minimize two objectives: the combined traveling distance and the maximum traveling distance among the UAVs, represented by Equations (4.2) and (4.3), respectively. Constraints (4.4) and (4.5) ensure that each viewpoint is visited only once, except for the depot. Constraints (4.6) and (4.7) guarantee that all UAVs are dispatched from the charging station and return to it after visiting their assigned viewpoints. Constraint (4.8) ensures that all viewpoints are visited, while Constraint (4.9) enforces the integrality of the problem. The formulated problems present conflicting objectives. For instance, minimizing the combined traveling distance may lead to assigning all viewpoints to a single UAV, which creates an imbalance in workload among the UAVs and results in a larger maximum traveling distance. Although it is possible to reduce the MO problem to a weighted sum single-objective problem, this approach does not guarantee optimality and only represents a portion of the trade-off curve. Therefore, it is preferable to obtain the trade-off curve that represents the optimal solution set, allowing the decision maker to select the desired solution. This trade-off curve is represented by a set of non-dominated solutions known as the Pareto optimal front P.

4.3.2 NSGA-II

The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) was first proposed by Deb et al. [75]. This algorithm is a variant of the GA designed specifically to solve MO optimization problems. Its main framework is similar to that of traditional GA, employing various genetic operators to facilitate the evolution process. To determine the domination rank of solutions, NSGA-II utilizes a fast non-dominated sorting technique. Additionally, crowding distance sorting is implemented to enhance the diversity of the obtained Pareto optimal front. Elitism is also incorporated to accelerate the algorithm's convergence rate. The pseudo-code for NSGA-II is summarized in Algo. 4.1.

Algorithm 4.1 NSGA-II

Input: Population size P, number of generation G, crossover rate r_c, mutation rate r_m

Output: Found Pareto optimal front

1: Initialization
2: **for** $i \leftarrow 1$ to G **do**
3: Fitness evaluation
4: Perform crossover with the probability of r_c
5: Perform mutation with the probability of r_m
6: Fast non-domination and crowding distance sorting
7: **end for**
8: **return** found Pareto optimal front

The algorithm begins by generating a set of solutions randomly. A global search is then performed using the crossover operator, which aims to explore new areas within the solution space. Following this, a local search is conducted through the mutation operator, which exploits existing solutions to further enhance their fitness. To select appropriate solutions for the crossover operations, various selection operators can be employed, including roulette wheel selection, tournament selection, or rank weighting selection. Regardless of the selection method used, it is crucial to maintain adequate selection pressure towards promising solutions to ensure that the algorithm converges on the optimal solution. However, the crossover operator in NSGA-II has an exploitative nature, relying heavily on previous and initial solutions. If the initialization fails to generate promising solutions, subsequent iterations may be misled, hindering convergence to the optimal solution. Therefore, the initialization process can benefit from an informed search, utilizing heuristic information to improve solution quality.

4.3.3 Ant Colony Optimization

Ant Colony Optimization (ACO) was first proposed by Dorigo [76]. This bio-inspired algorithm mimics the behavior of ants in nature. When an ant from a colony discovers a food source, it lays down pheromone trails to guide other ants toward that source. As more ants traverse the same path, they deposit additional pheromones, further reinforcing the trail. Similarly, ACO utilizes two types of information: pheromone information and a heuristic function. The pheromone information is related to the objective function, while the heuristic function is inversely proportional to the distance between two points. The pseudo-code for the algorithm is summarized in Algo. 4.2.

Algorithm 4.2 Ant Colony Optimization

Input: Relative importance of pheromone trails α, relative importance of heuristic function β, pheromone evaporation rate ρ, pheromone update constant Q, initial pheromone matrix $\boldsymbol{P}$, heuristic matrix $\boldsymbol{D}$, number of generations G

Output: Best found solution

```
1: for i ← 1 to G do
2:     Artificial ant construct solution based on the state transition rule
3:     Global pheromone update
4: end for
5: return best found solution
```

At each iteration, the artificial ant will construct the solution based on the state transition rule, which is represented in Eq. 4.10,

$$s = \begin{cases} \arg\max_{j \in C} [\tau(i,j)]^{\alpha} \cdot [\eta(i,j)]^{\beta}, & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \tag{4.10}$$

where $\tau(i,j)$ is the pheromone trail between waypoint i and j, while $\eta(i,j)$ represents the heuristic function between the two waypoints. C is the list of all the unvisited waypoints, and q is the probability that the artificial ant will choose the next waypoint based on the state transition rule. If $q \geq q_0$, the next waypoint will be selected based on the probability:

$$p(i,j) = \begin{cases} \dfrac{([\tau(i,j)]^{\alpha} \cdot [\eta(i,j)]^{\beta})}{\sum_{l \in C}[\tau(i,l)]^{\alpha} \cdot [\eta(i,l)]^{\beta}}, & \text{if } s \in C \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

The probability will exhibit a selection bias toward waypoints with higher combined information. Once the solution is fully constructed, it will be evaluated, and the objective value will be used for the pheromone update. The global pheromone update aims to reward path segments that belong to better-performing inspection paths, reinforcing the pheromone information and contributing to the evolution process. The global pheromone update can be expressed as follows:

$$\tau(i,j) = (1-\rho) \cdot \tau(i,j) + \rho \cdot \Delta\tau(i,j) \tag{4.12}$$

The first term represents pheromone evaporation, which aims to erase information from past iterations, preventing the algorithm from getting trapped in local optima. The second term represents pheromone reinforcement. $\Delta\tau(i,j)$ is the amount of pheromone deposited which can be calculated as follows:

$$\Delta\tau(i,j) = \frac{Q}{F} \tag{4.13}$$

where F is the objective value of the optimization problem, and more pheromones will be deposited when the objective value decreases. The iteration continues until the maximum number of generations is reached. To further enhance search performance, a local pheromone update can be introduced. This update occurs during the solution construction phase. After the artificial ant selects the next waypoint to visit, the local pheromone update reduces the pheromone information on the recently selected path, discouraging subsequent ants from visiting the same route. This variant of ACO is known as the Ant Colony System (ACS), which exhibits improved exploration capabilities compared to standard ACO. Both ACO and its variants have demonstrated superior convergence rates due to their utilization of pheromone and heuristic information. Since initialization is aided by the heuristic function, ACO is less likely to be misled compared to GA. However, the algorithm is susceptible to local optima and stagnation, as the heuristic information may inhibit exploration of new areas in the solution space. To mitigate this behavior, a robust local search component can be employed. Additionally, ACO requires precise parameter tuning to achieve optimal performance, which complicates the algorithm's implementation.

4.4 Proposed method

As summarized, NSGA-II possesses a more powerful local search capability, but its performance and convergence rate are limited by the quality of the initial solution. Conversely, ACO demonstrates a fast convergence rate due to its use of pheromone information; however, its weak local search ability makes it vulnerable to local optima. It is evident that the strengths of both algorithms can complement each other. Therefore, a hybrid algorithm, NSGACO, is proposed to harness the benefits of both NSGA-II and ACO.

The flow chart of the proposed NSGACO is illustrated in **Fig.** 4.2. The algorithm adopts the main framework of NSGA-II while incorporating elements of ACO to enhance its performance. As previously mentioned, initialization is crucial to the evolution process of the algorithm. In NSGACO, initialization leverages both NSGA-II and ACO methodologies. Using heuristic information, the ACO component initializes one solution, while NSGA-II generates the remaining solutions through a random approach. The performance of the solutions is then evaluated and sorted according to their domination rank. Pheromone matrices are initialized based on the non-dominated solutions, and a subtour library is created in accordance with these matrices. Global and local searches are conducted using the crossover and mutation operators, respectively. If improvements are found in the newly generated solutions, a global pheromone update is executed, and the subtour library is updated accordingly. The iteration continues until the maximum number of generations is reached, at which point the best-found Pareto optimal front is returned. The subsequent sections will discuss different components of NSGACO in detail.

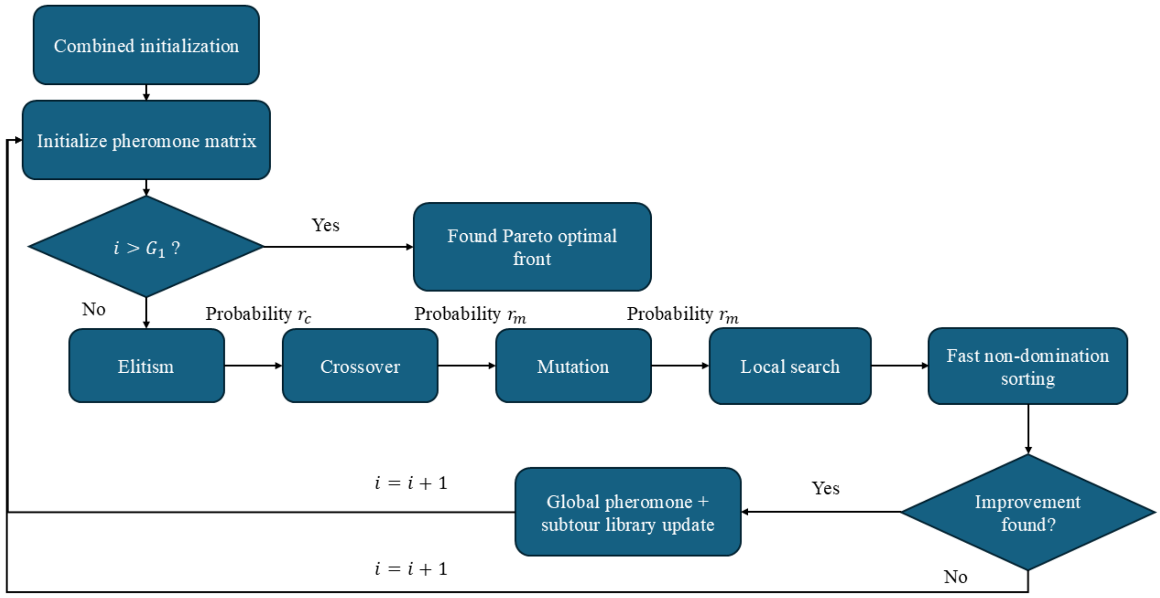


Fig. 4.2 Flow chart of NSGACO.

4.4.1 Chromosome representation

Before delving into the details of the algorithm, it is important to discuss chromosome representation. This aspect is crucial in GA and their variants, as an appropriately designed representation can significantly reduce computation time. The chromosome should be concise to minimize memory usage, and repetition must be avoided, ensuring that each combination is represented by only one chromosome. In NSGACO, a two-part chromosome representation is employed, as illustrated in **Fig.** 4.3.

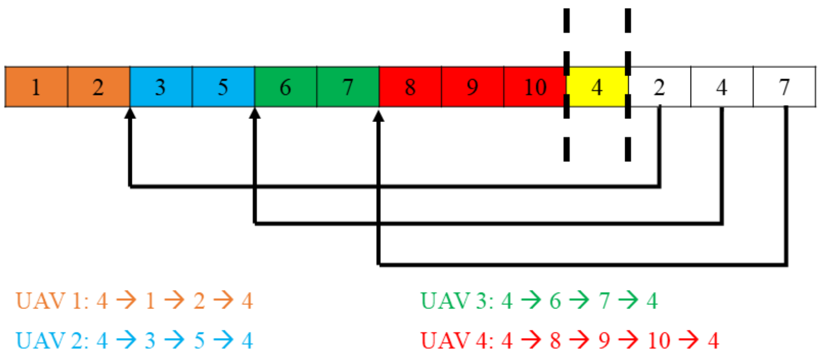


Fig. 4.3 Chromosome representation of an MTSP with 10 waypoints and 4 UAVs.

The chromosome consists of $(n + m - 1)$ gene. The first $(n - 1)$ genes represent the traversal sequence of each UAV, while the n^{th} gene represents the depot location. The rest of the chromosome represents the breakpoint, which is used to identify the corresponding UAV for each waypoint. This representation allows for concise encoding of all necessary information while minimizing redundant solutions.

4.4.2 Initialization

To ensure the convergence of the algorithm, it is crucial to obtain promising solutions during initialization. Therefore, ACO is employed in the initialization phase to enhance the performance of NSGACO. Initially, the heuristic function based on path segment distance is calculated. The ACO then generates one solution using this heuristic function, while NSGA-II randomly generates the remaining solutions. This ACO approach guarantees the acquisition of promising solutions, while NSGA-II explores the solution space freely, promoting a more diverse solution set. By ensuring that promising solutions are obtained, the convergence rate of the algorithm can be accelerated, preventing it from being misled.

4.4.3 Selection

Before the crossover operation, appropriate solutions must be selected as parent solutions to facilitate evolution. After obtaining the complete solution set, fast non-dominated sorting ranks the solutions based on their domination rank, and fitness functions are assigned to each solution. A selection operator is then employed to choose parent solutions from the population. The selection process should favor promising solutions while incorporating randomness to prevent convergence to local optima. In NSGACO, tournament selection is utilized. In this method, several solutions are first selected, and then solutions within this subset are chosen based on a probability proportional to their fitness values. This randomized approach in tournament selection ensures that diversified solutions can be selected while maintaining a bias toward promising solutions.

4.4.4 Crossover

One of the major issues with conventional crossover operators, such as Order Crossover (OX) and Partially Mapped Crossover (PMX), is the high probability of destroying schemata. The evolutionary process of GA relies heavily on the schemata of promising solutions, which represent beneficial combinations in the MTSP. To converge on the optimal solution, these schemata must be preserved in the child solutions during the crossover operation. A high probability of destruction prevents useful information from being carried forward to the next generation, undermining the evolution process. The primary reason for this high probability of destruction is that the operator lacks knowledge about the performance of different combinations, making it difficult to preserve the better-performing ones. By utilizing external information regarding the performance of corresponding combinations and constructing a reference table, the likelihood of destruction can be reduced. Consequently, an ACO-inspired crossover operator is employed in NSGACO, incorporating a subtour library during the operation to ensure that promising edges are preserved. Using pheromone matrices and the heuristic function, the decision information δ for different combinations is computed using Eq. 4.14.

$$\delta(i,j) = \tau_1(i,j)^{\alpha_1} \cdot \lambda\tau_2(i,j)^{\alpha_2} \cdot \eta(i,j)^{\beta} \tag{4.14}$$

where $\lambda \in \{1, m\}$ is the weighting used to balance the importance of the two pheromone matrices. If the decision information of a combination exceeds the average value, the corresponding combination is stored in the subtour library, which is represented by a $n \times n$ matrix. The average value serves as the cut-off point to ensure the library's diversity, allowing slightly worse-performing combinations to be included. This approach helps prevent the algorithm from becoming trapped in local optima.

In each iteration, the crossover operator is executed with a probability of r_c, and tournament selection is employed to choose two appropriate parent solutions based on their fitness. The crossover operation focuses solely on the traversal sequence of the chromosome. Therefore, the chromosomes of the parent solutions are first edited to obtain just the first section. Using the edited chromosomes, the operator selects a starting waypoint from either parent. This selected waypoint is then added to a tabu list to prevent waypoints from being revisited. After that, the subsequent waypoints are identified from the parent solutions. The operator checks if the identified connections belong to the subtour library. If one of the connections is found in the library, it will be selected as it represents a promising combination with higher-than-average decision information. If both connections belong to the library, the operator will choose the one with the higher decision information. In cases where the decision information is the same, a random connection will be selected from the candidates. If both candidates are in the tabu list, the operator will select a random unvisited waypoint. This process continues until the tabu list is full and all waypoints have been visited. The traveling sequence is then encoded to form a complete chromosome. Both the depot gene and the breakpoint sections are selected from one of the parent solutions to create the complete chromosome. The finalized chromosome is then added to the new population.

4.4.5 Mutation

Local search is then performed using mutation operators, which make small changes to the existing solutions. By only introducing minor adjustments, the risk of destroying schemata is minimized while exploring the neighborhood of the current solution to identify potential improvements. In NSGACO, three mutation operators are utilized: reverse mutation, slide mutation, and swap mutation. These operators are executed based on the same probability. **Figure** 4.4 summarizes their corresponding procedures.

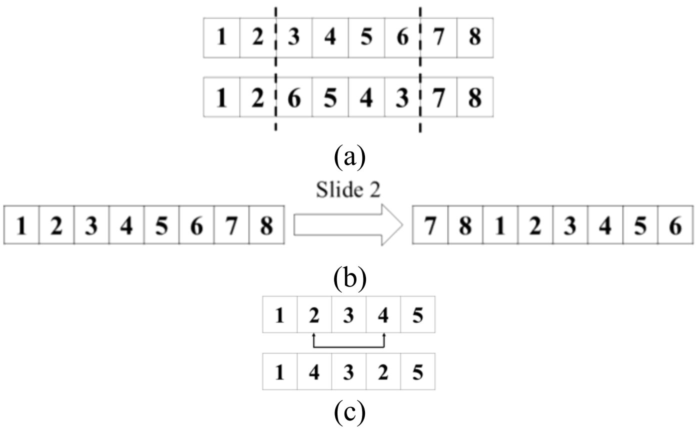


Fig. 4.4 Mutation operators used in NSGACO. (a) Reverse mutation. (b) Slide mutation. (c) Swap mutation.

Similar to the crossover operators, the mutation operators only modify the first section of the chromosome. The reverse mutation aims to eliminate path crossings within the traversal sequence by reversing a selected segment of the chromosome. This mutation first generates two random cutting points, illustrated by dashed lines in Figure 4.4(a). The section between the cutting points is extracted and reversed, while the other sections remain unchanged, effectively removing potential path crossings. The slide mutation modifies the allocation of waypoints by shifting the traversal sequence. A random variable is selected to determine the sliding size, and the traversal sequence is then shifted to the right according to this size. This method preserves the order of the sequence while altering waypoint allocation. The swap mutation also targets the removal of existing path crossings. Instead of changing the sequence of a segment, the swap mutation alters the positions of two genes within the chromosome. Two genes are randomly selected, and their positions are swapped. This mutation introduces small changes to the solution, enabling the algorithm to explore the neighborhood of the existing solution. With these three mutation operators, NSGACO ensures a thorough exploration of the neighborhoods of the solutions.

4.4.6 Local search

While the mutation operators can explore the neighborhood of existing solutions, improvement is not guaranteed. Therefore, additional local search operators are needed to increase the chances of finding improved solutions. In NSGACO, two additional local search operators are utilized. To avoid extensive computation, these local search operators are executed with a probability of r_m. The first local search operator is the 2-opt algorithm. This algorithm iteratively selects two pairs of path segments and swaps their corresponding positions. If a better solution is found, the changes are retained. The pseudo-code for the 2-opt algorithm is summarized in Algo. 4.3.

Algorithm 4.3 2-opt algorithm

Input: Initial chromosome

Output: Improved chromosome

1: Evaluate the performance of the solution

2: **for** every salesman **do**

3:     **repeat until** no improvement is made

4:     **for** every distinct and non-adjacent edge pair **do**

5:         Create a new solution by swapping the path segment pair and evaluate

6:         **if** improvement found **then**

7:             Replace old solution with new solution and update the best found tour length

Encode the improved solution into the chromosome

return final chromosome

The 2-opt operator only considers the performance of the inspection path for each UAV, limiting the potential for improvement since the waypoint allocation remains unchanged. Consequently, an additional local search procedure is necessary to enhance waypoint allocation. In NSGACO, a second local search operator, known as the exchange operator, is utilized to improve the distribution of waypoints among the UAVs. The pseudo-code for the exchange operator is summarized in Algo. 4.4.

Algorithm 4.4 Exchange operator

Input: Initial chromosome

Output: Improved chromosome

Decode the chromosome into m inspection paths

for every UAV pair **do**

  Calculate the combined path length

  repeat until no improvement is made

  for every waypoint pair in the UAV pair **do**

    for every possible insertion point **do**

      Exchange waypoints at the insertion points and evaluate

      if improvement found **then**

        Replace old solution with the improved solution and update best found result

Encode the inspection path into chromosome

return Improved chromosome

The exchange operator first identifies all possible UAV pairs and evaluates the combined inspection path length. For each waypoint pair, the operator creates a new inspection path by inserting the waypoints at different insertion points. The performance of this new inspection path is then assessed. If an improvement is found, the old path is replaced with the improved version. Similar to the 2-opt operator, this iteration continues until no further improvements can be made. The goal of the exchange operator is to transfer waypoints between UAVs, ensuring that waypoints are assigned to the appropriate

UAVs. By utilizing these two local search operators, both the performance of individual inspection paths and the waypoint assignments can be enhanced, further accelerating the convergence of the algorithm. However, since both local search modules are computationally expensive, they will only be executed based on a small probability.

4.4.7 Global pheromone update

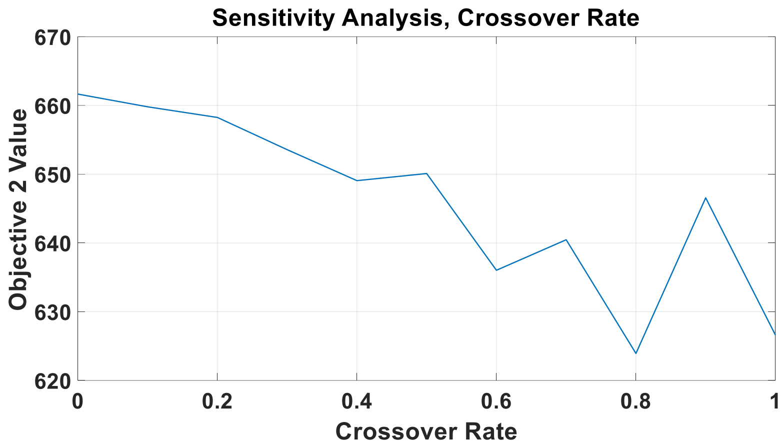
As improvements are identified during the search process, both the pheromone matrices and the subtour library should be updated to better represent the best-found solution. However, as noted in the literature review, the pheromone update process can be computationally intensive. Therefore, in NSGACO, the update process is only activated when actual improvements are found. After the crossover and mutation operations, fast non-dominated sorting is executed to arrange the solutions based on their domination rank. If the newly found solutions dominate the previous non-dominated solutions, improvements are confirmed, and the global pheromone update is initiated. The non-dominated solutions from the new generation are first extracted, and the global pheromone update is performed based on the information carried by these improved solutions. The pheromone update rule is summarized in Eq. (4.15).

$$\Delta\tau_f(i,j) = \frac{Q}{F_f}, \quad f \in \{1,2\} \tag{4.15}$$

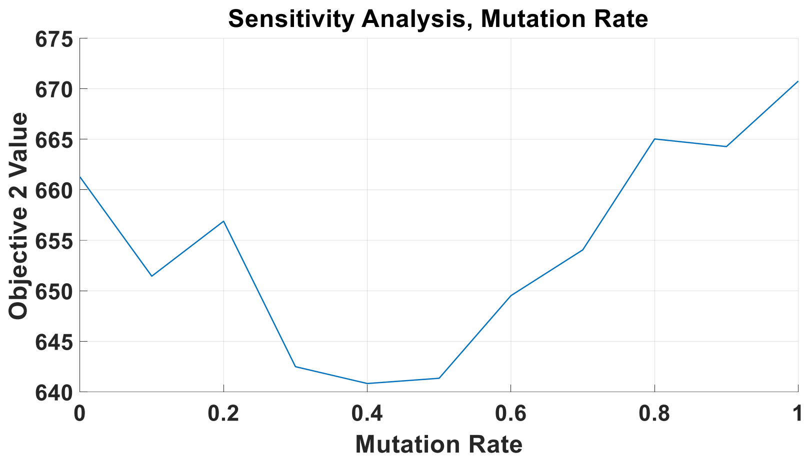
where the update value is inversely proportional to the objective values of the minimization problem. By establishing a condition for the pheromone update, this operation does not need to be executed in every iteration, especially as the algorithm approaches the end of its iterations. Thus, avoiding unnecessary computations.

4.4.8 Sensitivity Analysis

To determine the hyperparameters used for the designed algorithm, a sensitivity analysis is performed to identify the optimal hyperparameters for solving the MO problem. The primary hyperparameters affecting the performance of the NSGACO algorithm are the crossover rate and mutation rate. The crossover rate dictates how often existing information is utilized, while the mutation rate influences the frequency of exploration. Conventionally, a high crossover rate is preferred to encourage exploitation, whereas a low mutation rate helps prevent the destruction of promising solutions. The results of the sensitivity analysis support this theory, and they are summarized in **Fig.** 4.5.



(a)



(b)

Fig. 4.5 Results of the sensitivity analysis. (a) Crossover rate. (b) Mutation rate.

The sensitivity analyses are performed on the 3eil51 instance, using objective 2 to determine the optimality of the solution, as it is more representative of the MO problem. For the crossover rate, the solution quality decreases as the crossover rate increases. When the algorithm utilizes less exploitation, evolution occurs less frequently, leading to poor performance. Although solution quality is expected to improve with a higher crossover rate, diminishing returns are observed when $r_c > 0.8$. Excessive exploitation can trap the algorithm in local optima, hindering improvement. Therefore, a crossover rate of 0.8 is selected for the NSGACO.

A similar trade-off is observed in the mutation rate. Generally, the solution quality improves as the mutation rate decreases. A small mutation rate ensures that promising solutions are not easily destroyed while still providing adequate exploration of the solution space. Diminishing returns are also observed when $r_m < 0.4$, as a low exploration probability restricts the algorithm's searching ability, preventing thorough exploration of the solution space. Consequently, a mutation rate of 0.4 is selected for the NSGACO.

4.4.9 Migratory system

The proposed NSGACO focuses solely on searching for the Pareto optimal front of the MO problem with a given depot. Therefore, additional processes are necessary to identify the optimal depot location. In current research on the MTSP with an unknown depot, the depot location is often selected manually or determined through random search. However, manually selected depots may not be optimal, as they are not subjected to the evolutionary process. Conversely, the efficiency of random search is not guaranteed, and its effectiveness diminishes as problem complexity increases. To enhance search efficiency, a method that utilizes heuristic information is required. In this work, a heuristic approach based on pheromone intensity is proposed to guide the system's migration. For simplicity, the test problem names will be encoded as $(m + problem_{id})$. For example, "5eil51" represents the benchmark problem "eil51" with 5 UAVs.

In the single depot MTSP, each UAV should ideally be responsible for a distinct section of the inspection area, ensuring that the areas of interest do not overlap. To achieve this, the UAVs should depart from the depot and travel in different directions. If the depot location is suboptimal, the UAVs may tend to travel in the same direction, leading to path crossings and overlaps in their areas of interest. **Figure** 4.6 illustrates this scenario. Utilizing this behavior, heuristic information can be designed to monitor path crossings among the UAVs. In the proposed migratory system, pheromone intensity is employed to assess the proximity of path segments from different UAVs, tracking path crossings and evaluating the depot's optimality. This information facilitates an effective search for the optimal depot location.

Fig. 4.6 Depot comparison of 5eil51. (a) Ideal depot location. (b) Unideal depot location

Figure 4.7 illustrates the flowchart of the proposed migratory system. The process begins with the initialization of a randomly selected depot. Next, the NSGACO is executed to obtain the Pareto optimal front for the chosen depot. Following this, the pheromone intensity is calculated, and the migration direction is determined. Based on this direction, a new depot is selected, and the NSGACO is executed again. This iterative process continues until the stopping criterion is met. To ensure a comprehensive search, local searches are conducted at locations near the best-found depot. Finally, the Pareto optimal front, along with the optimal depot location, is returned.

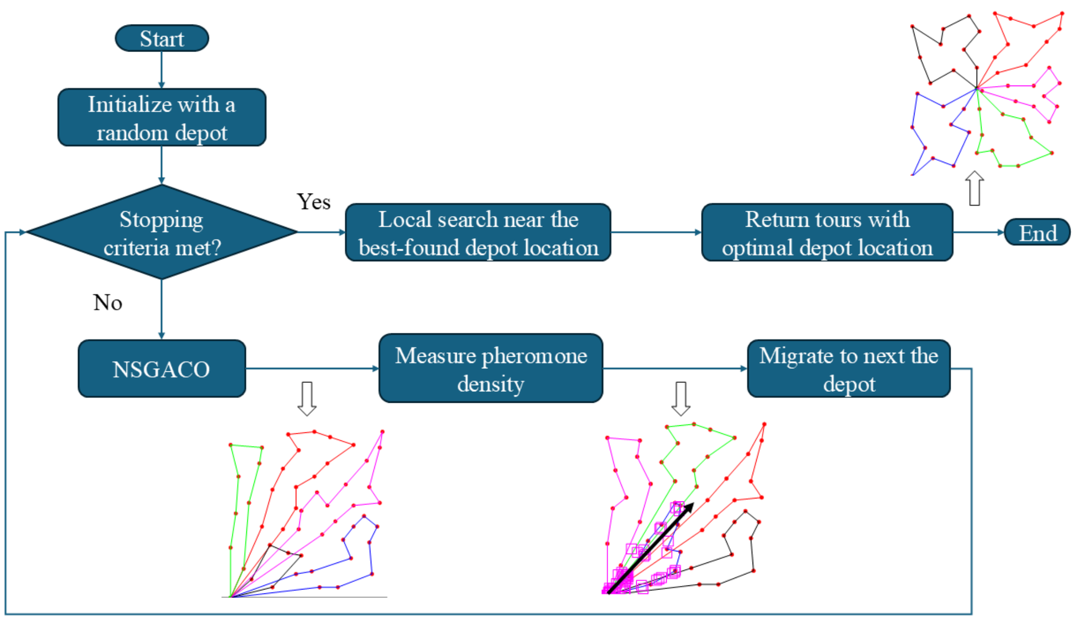


Fig. 4.7 Flow chart of the migratory system.

To calculate the pheromone intensity, all possible UAV pairs are first identified. Their corresponding inspection paths are then sampled, and the distances between the sampled points are calculated. If the distance between these points is less than the average value, the location is flagged as having abnormal pheromone intensity, indicating potential path crossings. These flagged locations are combined to form a vector that guides the migration. As illustrated in **Fig.** 4.8, if the depot is optimally located, the areas with abnormal pheromone intensity should only be near the depot and spread out in different directions. Consequently, the resulting vector will have a small magnitude, suggesting a weak tendency to migrate. Conversely, if the depot is poorly located, the areas with abnormal pheromone intensity will be distributed across the solution plane and aligned in the same direction. In this case, the resulting vector will have a large magnitude, indicating a strong tendency to migrate. After obtaining the resulting vector, a list of candidate depots along the migration direction is compiled. One of these candidates will be selected as the next depot, with a selection bias favoring closer candidates. This design aims to prevent oscillation; if the system continuously migrates to distant depots, it may oscillate around the ideal depot location and fail to converge. The bias encourages incremental changes, ensuring that the desired depot location is not overlooked. Once the stopping criterion is met, a local search is conducted to further explore improvements near the best-found solution. If no further improvements can be made, the migration process is terminated, and the Pareto optimal front corresponding to the best-found depot location is returned.

(a) (b)

Fig. 4.8 Illustration of areas with abnormal pheromone intensity on 5eil51. (a) Ideal depot. (b) Unideal depot.

4.4.10 Time Complexity

The proposed algorithm employs a nested loop design, with the NSGACO functioning as the inner loop and the migration process as the outer loop. For a selected depot, the corresponding MTSP is initially solved by the NSGACO. The pheromone intensity computed from the resulting Pareto optimal front is then used to determine the migration direction, leading to the selection of a new depot location. The MTSP with the newly selected depot is solved again, and this iteration continues until the stopping criteria are met. Consequently, the time complexity of the algorithm can be summarized as $O(G_1 \cdot G_2 \cdot P \cdot L)$, where $G_1$and G_2 represent the number of generations of the NSGACO and the migration process, respectively, while P is the population size and L is the length of the chromosome.

4.4.11 Practicability

To implement the proposed inspection path planning algorithm in a real-world scenario, the issue of obstacle avoidance must be addressed, as it is not considered in the current work. While the algorithm aims to generate an efficient order for the UAV to navigate through the viewpoints, it does not guarantee a collision-free path. Therefore, additional collision checks are necessary. Local path planning algorithms, such as RRT, can be utilized to establish collision-free connections between viewpoints, similar to prior work.

To execute the inspection mission in practice, the viewpoint allocation and optimal depot location should first be determined using the NSGACO with the migration mechanism. Next, RRT are utilized to establish collision-free connections between viewpoints. The charging station, representing the depot location, should then be set up. UAVs equipped with onboard cameras, flight controllers, and flight computers can then carry out the inspection task. The onboard camera will capture inspection-related images while also facilitating dynamic obstacle avoidance.

4.5 Simulation results and discussion

The performance evaluation is divided into two parts. First, the effectiveness of NSGACO in solving the MO-MTSP is assessed, and its performance is compared to other heuristic algorithms. Second, the migratory behavior of the system is analyzed.

4.5.1 MTSP solving performance comparisons

To evaluate the performance of the NSGACO, benchmark problems from the open-source library TSPLIB [77] are utilized. Additionally, to demonstrate improvements in convergence rate and solution quality, NSGACO is compared with other heuristic algorithms. Specifically, it is compared to two hybrid algorithms—GA-ACO and ACO-GA—and two standalone algorithms: GA and ACS. GA-ACO is a two-step algorithm where GA is used to initialize and search for the optimal solution during the first half of the iterations, while ACO is responsible for the latter half. Conversely, ACO-GA also follows a two-step approach, with ACO handling initialization and search during the first half and GA managing the latter half. Since the numerical studies focus solely on the performance of the algorithms in solving the MO-MTSP, all algorithms are evaluated with the same manually selected depot. The parameters of the algorithms are summarized in Table 4.1, while Table 4.2 compares the performance of the different algorithms on the benchmark problems.

Table 4.1 Parameters of the algorithms.

Parameter	Value
G	100
Population size	50
r_c	0.8
r_m	0.4
λ	m
α_1	1.1
α_2	1.1
β	4
ρ	0.9
Q	100

The performance of NSGACO is promising. Across the benchmark problems, NSGACO outperforms all other algorithms, with improvements becoming more pronounced as scenario complexity increases. Although NSGACO is occasionally outperformed by ACS, it is not entirely dominated by it. Significant improvements for objective 2 are observed across all scenarios. In contrast, NSGA-II exhibits the worst performance among all algorithms, as it is the only one that does not utilize any heuristic information. Consequently, it experiences a slow convergence rate and fails to reach the optimal solution. Both two-step algorithms, GA-ACO and ACO-GA, demonstrate similar performance, with their corresponding Pareto optimal fronts overlapping. However, ACS consistently outperforms both GA-ACO and ACO-GA, as heuristic information is utilized throughout the iterations, whereas it is only applied during half of the iterations in the two-step algorithms. This observation underscores the importance of employing heuristic information throughout the algorithm, justifying the inclusion of pheromone matrices and the subtour library in NSGACO.

The obtained solutions for the benchmark problem 5eil101 are illustrated in **Fig.** 4.9. Since the 2-opt algorithm is implemented in all heuristic algorithms, no local crossing paths are observed. In addition, the solution obtained by NSGACO, which utilizes the exchange operator, contains no Areas of Interest (AOI) overlap, while overlaps are present in the solutions from all other algorithms. Without the incorporation of heuristic information, NSGA-II fails to achieve a balance between the two objectives, resulting in a solution that primarily optimizes only one objective.

Table 4.2 Results of the algorithms.

Problem	m	NSGACO		NSGA-II		ACS		GA-ACO		ACO-GA	
		Obj1	Obj2	Obj1	Obj2	Obj1	Obj2	Obj1	Obj2	Obj1	Obj2
eil51	3	**162**	**466**	253	636	166	481	166	483	168	486
	4	**129**	**497**	206	641	132	510	135	518	133	513
	5	**113**	**527**	192	707	116	544	118	553	118	545
st70	3	267	**759**	470	1161	**265**	773	269	785	270	784
	4	225	**819**	395	1229	**222**	842	228	865	229	871
	5	**197**	**877**	353	1282	206	936	210	948	206	946
eil101	3	**241**	**698**	415	1088	244	714	245	721	246	725
	4	**194**	**721**	333	1069	195	749	200	760	200	762
	5	**166**	**759**	320	1165	166	789	178	834	170	804
kroA150	3	**10248**	**29703**	22134	57949	10251	29910	10544	30641	10688	31119
	4	**8171**	**31113**	18801	59925	8472	32027	8192	31549	8468	32402
	5	**7186**	**33481**	16830	62408	7472	35228	7635	35742	7675	35960

The diversity of the obtained solutions is then examined. One of the key objectives in solving multi-objective problems is to achieve a diversified Pareto optimal front, which represents a set of non-dominated solutions that decision-makers can choose from for specific purposes. An ideal Pareto optimal front should be as diverse as possible to effectively represent the trade-off curve. The 5 UAV variant of the benchmark problems is used to study solution diversity. **Figure** 4.10 illustrates the Pareto optimal front found by all algorithms across 10 runs.

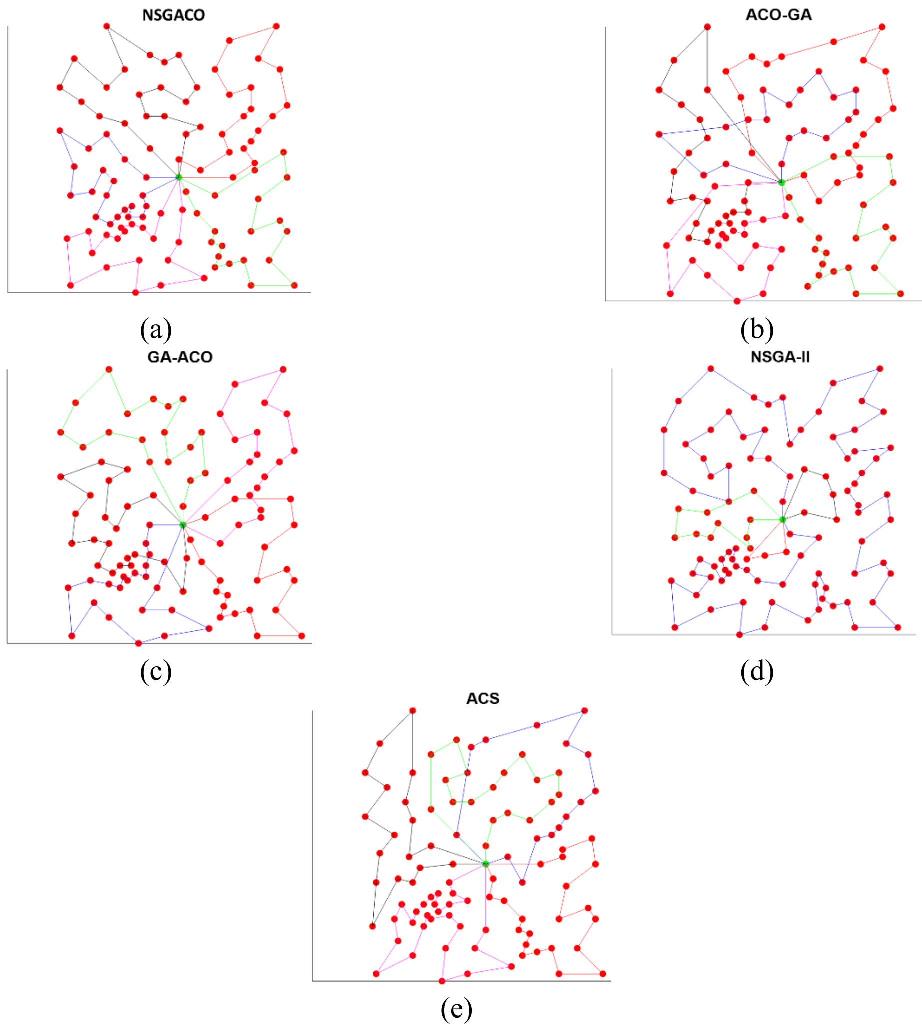


Fig. 4.9 Tours obtained for 5eil101. (a) NSGACO (b) ACO-GA (c) GA-ACO (d) NSGA-II (e) ACS.

NSGA-II is able to produce the most diverse Pareto optimal front among all algorithms. However, its solutions are dominated by all other algorithms, and its Pareto optimal front is significantly distanced from the others. When focusing only on the top four algorithms, NSGACO demonstrates the most diverse Pareto optimal front, as illustrated in **Fig.** 4.11. While the optimal fronts of other algorithms cluster in the same area, NSGACO generates an optimal front that is well-distributed across the solution plane. Due to the inherent nature of ACO, all hybrid algorithms that incorporate ACO are susceptible to local optima. However, the powerful mutation operators employed in NSGACO allow for small adjustments to existing solutions, introducing variation within the population and resulting in a more diversified Pareto optimal front.

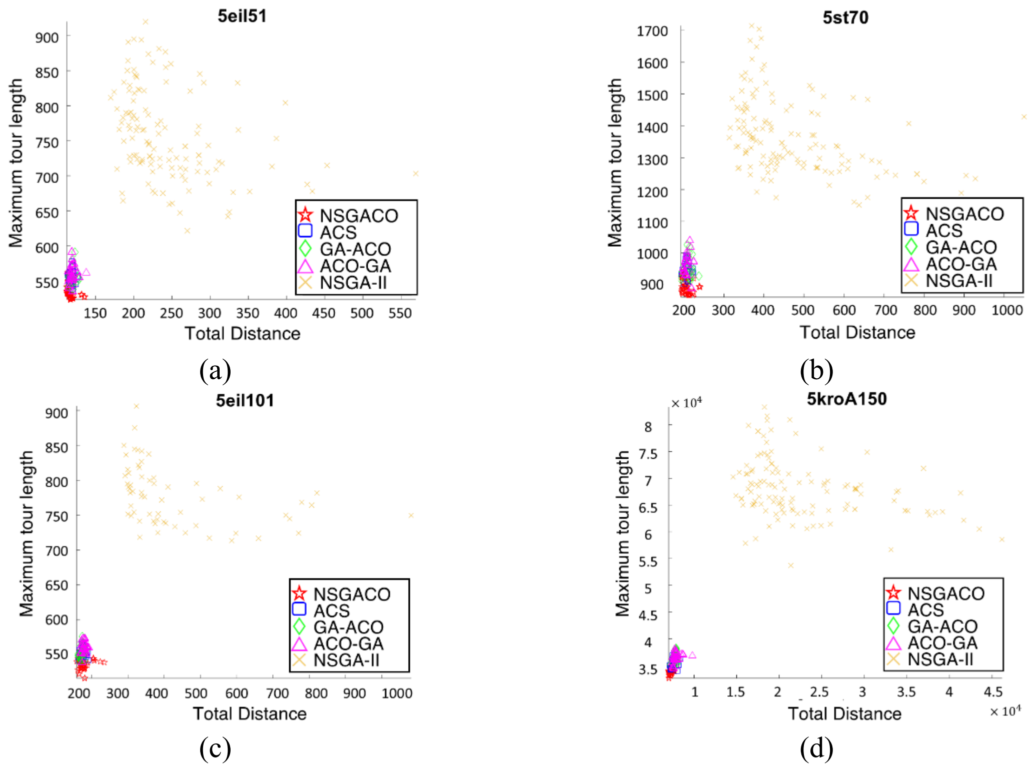


Fig. 4.10 Found Pareto optimal front of all 5 algorithms.

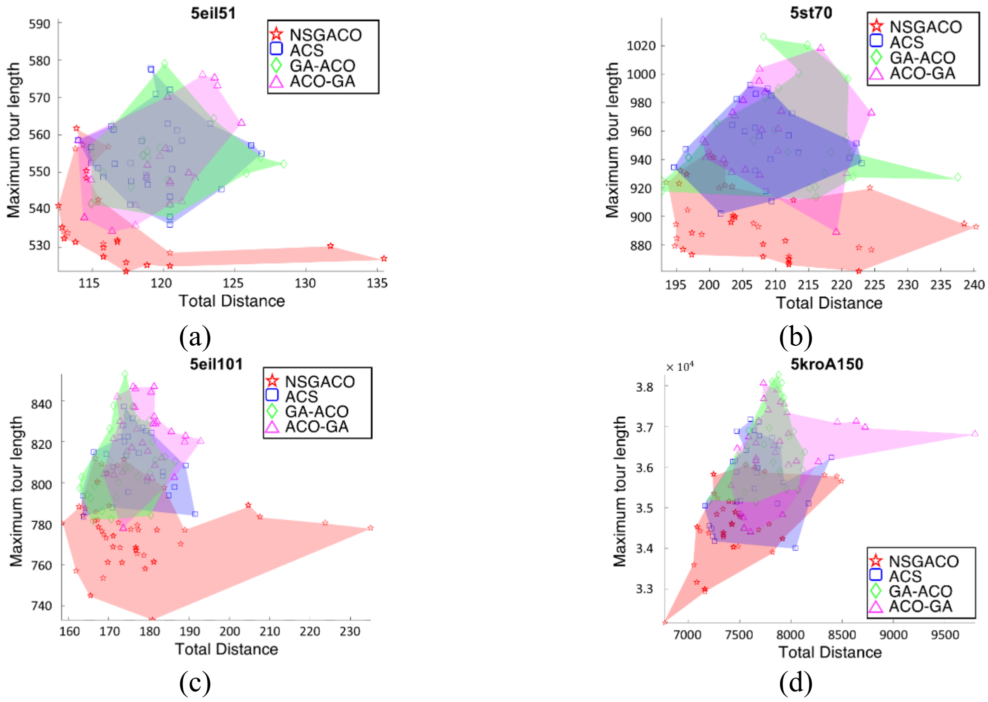


Fig. 4.11 Found Pareto optimal front of the top 4 algorithms.

The hypervolume indicator (Hv) is also utilized to analyze the performance of the algorithms. The Hv measures the quality of the solutions in terms of both convergence and diversity, representing the area between a reference point and the Pareto optimal front. The reference point is defined as a point worse than any feasible solution, positioned away from the Pareto optimal front. A larger Hv indicates a more optimal and diverse solution front. The Hv values are summarized in Table 3. Consistent with previous studies, NSGACO outperforms all other algorithms in terms of Hv, while NSGA-II shows the poorest performance. Although NSGA-II achieves the most diversified solution front, its solution quality is compromised, resulting in a low Hv value. Conversely, GA-ACO achieves a better Hv value than ACS, even though ACS produces a more optimal solution front. Since GA-ACO is initialized with GA, it can generate a more diverse solution set during the first half of the iterations. This diversity allows for a more comprehensive pheromone matrix, preventing pheromone accumulation on a small subset of path segments.

Table 4.3 Hypervolumes of the algorithms

Problem		NSGACO	NSGA-II	ACS	GA-ACO	ACO-GA
5eil51	Best	**9.20E-01**	5.63E-01	8.99E-01	9.00E-01	8.88E-01
	Mean	**8.91E-01**	4.24E-01	8.75E-01	8.77E-01	8.53E-01
	Worst	**8.62E-01**	3.02E-01	8.52E-01	8.23E-01	8.04E-01
5st70	Best	**8.53E-01**	3.89E-01	8.29E-01	8.37E-01	8.30E-01
	Mean	**8.30E-01**	2.45E-01	7.86E-01	7.89E-01	7.68E-01
	Worst	**7.96E-01**	1.33E-01	7.35E-01	7.55E-01	6.98E-01
5eil101	Best	**9.63E-01**	4.33E-01	9.47E-01	9.39E-01	9.17E-01
	Mean	**9.11E-01**	2.87E-01	9.08E-01	8.93E-01	9.03E-01
	Worst	8.65E-01	1.59E-01	8.66E-01	8.58E-01	**8.86E-01**
5kroA150	Best	**9.10E-01**	1.82E-01	8.80E-01	9.08E-01	8.89E-01
	Mean	**8.83E-01**	7.89E-02	8.48E-01	8.55E-01	8.23E-01
	Worst	**8.44E-01**	1.98E-02	8.17E-01	8.05E-01	7.69E-01

Next, the consistency of the algorithms is examined. An effective algorithm should demonstrate consistent performance across multiple instances. Accordingly, the boxplots for the Hv values of all algorithms across 10 runs are illustrated in **Fig.** 4.12. To indicate consistent performance, the boxplot should be as small as possible, reflecting minimal deviation in results. Due to the mutation and local search operators, some variation in performance is observed in NSGACO. While NSGACO does not obtain the best result in terms of performance consistency, its results remain competitive with those of other algorithms, and its superiority in terms of Hv values is again evident in the boxplot.

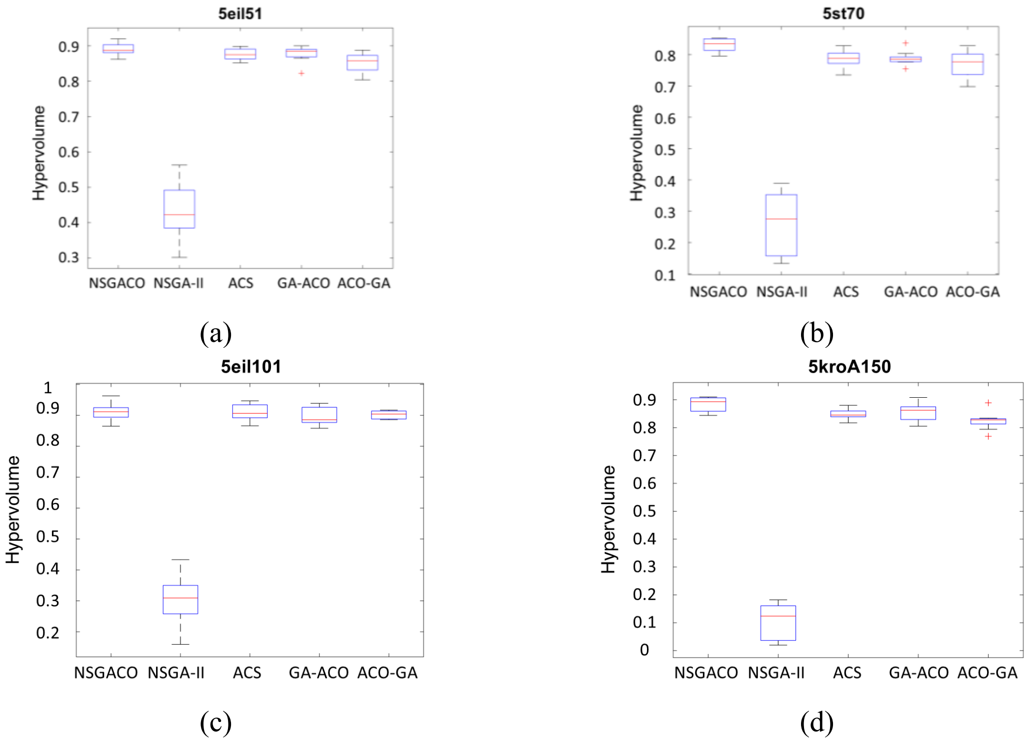


Fig. 4.12 Boxplot of the algorithms across 4 benchmark problems.

Lastly, the convergence rate of the algorithms is analyzed. The convergence curves of the algorithms across the benchmark problems are illustrated in **Fig.** 4.13. One of the primary objectives of NSGACO is to enhance the convergence rate of NSGA-II by incorporating pheromone matrices, and this improvement is evident across the various problems. Compared to NSGA-II, NSGACO shows significant enhancement, achieving near-optimal solutions within fewer generations. Once again, due to the absence of heuristic information, NSGA-II exhibits the worst convergence rate. The impact of heuristic information is also reflected in the convergence curve of GA-ACO. As the algorithm transitions from GA to ACO, its convergence rate accelerates immediately due to the utilization of heuristic information. Improvements provided by NSGACO become particularly apparent during the latter stages of the search process. While other algorithms exhibit stagnation and fail to find further improvements, NSGACO continues to explore for better solutions, highlighting the effectiveness of its mutation and local search operators. Additionally, ACS shows the most pronounced stagnation behavior, struggling to find improvements early in the iterations. The lack of local search operators prevents ACS from escaping local optima, resulting in poor performance. From the numerical studies, it can be concluded that NSGACO outperforms other algorithms in terms of solution quality, solution diversity, and search capability.

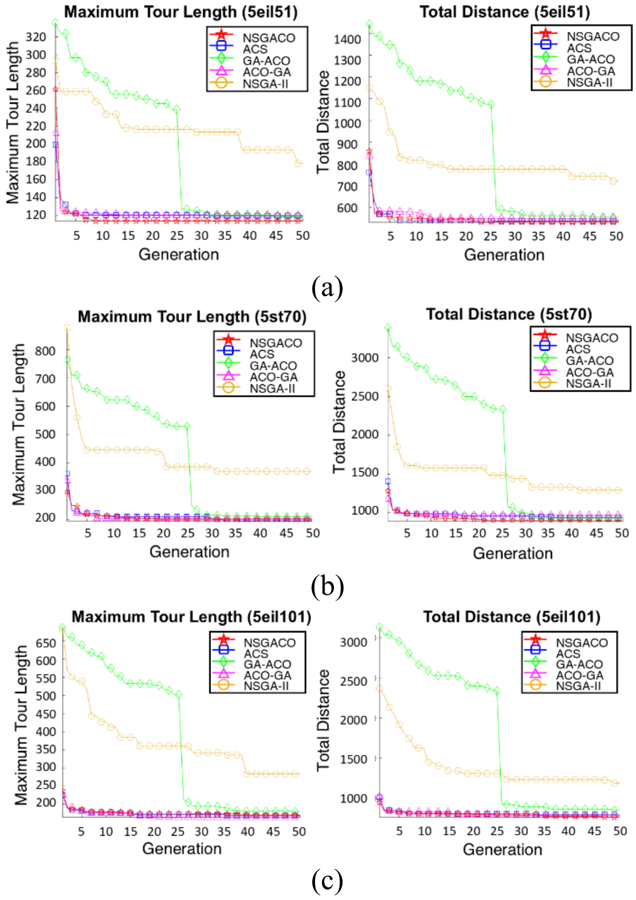


Fig. 4.13 Convergence rates of the algorithms in different benchmark problems.

Before studying the migratory behavior of the algorithm, the relationship between the number of UAVs and the objective values is studied. The numerical study focuses on the benchmark problem eil51. The simulation results with various numbers of UAVs are summarized in **Fig**. 4.14.

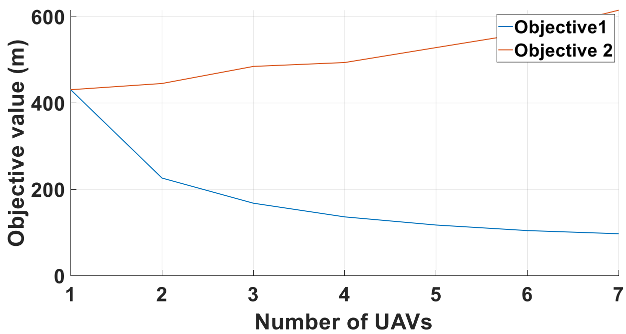


Fig. 4.14 The relationship between the objective values and the number of UAVs for eil51.

In the single UAV scenario, the conventional TSP is represented, where all viewpoints are assigned to one vehicle. This leads to extended path distances and potentially long mission times. As the number of UAVs increases, the maximum traveling distance among the UAVs gradually decreases. However, diminishing returns are observed when the number of UAVs exceeds two. Increased UAV numbers result in unavoidable path crossings, limiting the benefits of additional UAVs.

For Objective 1, there is a linear relationship with the number of UAVs, with incremental increases noted for each added UAV. In the studied benchmark problem, the optimal number of UAVs is two, as this configuration significantly reduces inspection time while only slightly increasing the total traveling distance. It is important to note that the optimal number of UAVs should be uniquely determined for each specific inspection task, taking into account the complexity of the mission.

4.5.2 Migratory behavior

Lastly, the migration behavior of the proposed system is examined. A simplified migration process is illustrated in **Fig.** 4.15. Initially, a random waypoint is selected as the depot location, which is suboptimal and leads to significant overlap between the areas of interest of different UAVs. The system then gradually migrates according to the direction indicated by the pheromone intensity. Ultimately, the system reaches an ideal depot location that results in no overlap between the areas of interest. To demonstrate the consistency of the migration performance, the boxplot of the number of iterations across 10 runs on various benchmark problems is presented in **Fig.** 4.16. Despite differences in the number of waypoints, the number of iterations remains consistent across the benchmark problems. The maximum number of iterations accounted for only 47%, 34%, and 24% of the total number of waypoints, respectively. With the selection bias during the migration process, oscillation is minimized, resulting in enhanced efficiency in problems with a greater number of waypoints. These results illustrate the effectiveness of the designed heuristic in guiding the migration process.

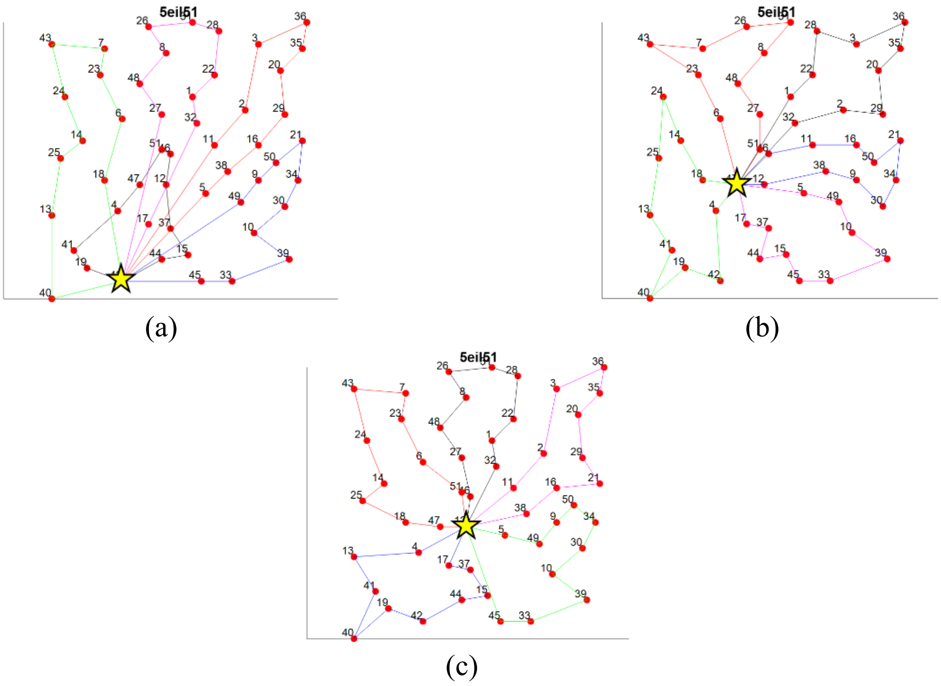


Fig. 4.15 Demonstration of the migration process. (a) Initial dispersion point (b) Migration (c) Final dispersion point.

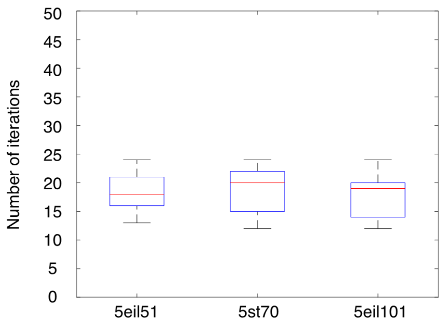


Fig. 4.16 Boxplot of the number of iterations of the migration process across the benchmark problems.

The consistency in terms of objective values is also examined. **Figure** 4.17 summarizes the objective values obtained across 10 runs. The objective values remain consistent across multiple runs, indicating that similar solutions can be reliably achieved. With the proposed heuristic, the migration module effectively directs the system to the ideal depot location, even when starting from different initial depots.

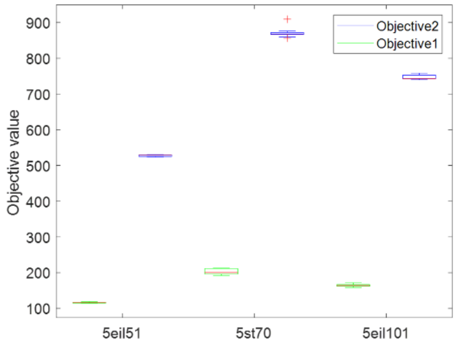


Fig. 4.17 Boxplot of the objective values across the benchmark problems.

4.6 Conclusion

In this section, a multi-UAV inspection system is proposed. An MO-MTSP with an unknown depot is formulated, and a heuristic hybrid algorithm, NSGACO, is introduced to solve the optimization problem. The proposed algorithm leverages features from both NSGA-II and ACO, and its modular design allows the advantages of both algorithms to be realized throughout the iterations. A migration mechanism is implemented to guide the system toward the ideal depot location. Since the depot location is a critical component of the system, incorporating it into the optimization problem enables further minimization of the objective values. Numerical simulations validate the performance of the proposed algorithm. In addressing the MO-MTSP, NSGACO outperforms other heuristic algorithms, and the effectiveness of the migration system is also confirmed.

5 Coverage Path Planning for Autonomous Aircraft Inspection using UAVs

In the previous chapters, the scalability and efficiency of inspection path planning were discussed, along with algorithms aimed at enhancing these aspects of autonomous inspection tasks. In this chapter, the focus shifts to the quality of autonomous inspection. A novel inspection path planning algorithm that utilizes parameterization is proposed. By generating a sweep path directly on the parametrized surface, the resulting inspection path can closely follow the contours of the inspection target, thereby enabling high-quality inspections.

5.1 List of notations

The notations used in this section are defined as follows:

d	Sweep direction
d'	Path direction
α	Path definition
β	Average triangle height
L	Average path segment length
n	Number of points per triangle
$\boldsymbol{p}$	2D sweep path
D	Offset distance
$\boldsymbol{A}$	Parameterized surface
$\boldsymbol{M}$	3D Model
$\boldsymbol{N}$	Surface normal
δ	Distance changed
$\boldsymbol{x_1}, \boldsymbol{x_2}$	Waypoint location
$\boldsymbol{V}$	Escaping vector
ϵ	Safety distance

μ	Ground clearance constraint
n_x	Number of waypoints
n_{rx}	Number of waypoints that require replanning

5.2 Motivation

Apart from infrastructure inspection, aircraft inspection is also a prominent application for UAVs. However, aircraft have complex features that pose challenges to the planning capabilities of existing algorithms. Additionally, safety is the top priority in the aviation industry, and the quality of the inspection task must take precedence over efficiency. Therefore, path planning algorithms should focus on enhancing this aspect.

Despite this need, there is limited research in the literature aimed at improving the quality of inspection tasks, highlighting a significant research gap. Furthermore, existing inspection path planning algorithms primarily rely on sampling-based techniques. While these techniques offer flexible path planning solutions, they may not be suitable for all inspection targets. The sampling-based approach typically requires a defined sampling space, which can be challenging for complex targets like aircraft. Additionally, the inherent nature of sampling can lead to inconsistent viewing distances, resulting in subpar inspection quality. To address these issues, this section proposes a parameterization-based path planning algorithm.

The proposed algorithm explores the relationship between 2D and 3D coverage path planning, utilizing parameterization to bridge different dimensions. Instead of treating the problem solely as a 3D issue, the approach simplifies the problem by converting it to a 2D framework. This not only enhances inspection quality but also unifies the coverage path planning problem across dimensions, enabling 2D planning algorithms to effectively address 3D problems. This approach offers a new perspective on solving 3D coverage path planning problem. The main contributions of this work are as follows:

1) This work presents a framework that extends 2D coverage planning to 3D coverage planning.

2) An optimization-based re-planning algorithm is proposed for collision avoidance, minimizing changes to the original planned path.
3) The generated waypoint-based solution can be directly executed on a UAV, facilitating the implementation of autonomous inspections.

5.3 Methodology

While the CPP is used for 3D coverage tasks, the 2D variant of CPP has been extensively studied. For 2D coverage tasks, the most intuitive method is the Boustrophedon path [78], which is a back-and-forth path that effectively covers the entire area of interest. To generate the sweep path, only a few simple parameters are needed, including the sweep direction, overlapping ratio, and turning radius. Although this method is straightforward to implement in 2D scenarios, it is rarely applied to 3D coverage tasks due to the increased complexity of parameter design in the additional dimension. While the longitudinal axis of the inspection target can be selected as the sweep direction, determining the corresponding path direction can be challenging, as illustrated in **Fig.** 5.1. Choosing a single direction for the path can result in an inspection path that fails to closely follow the contours of the inspection target, leading to poor inspection quality. Although optimization-based methods, such as the TSP, are feasible for generating the sweep path, their computational cost increases exponentially with the number of viewpoints, making it impractical to generate a high-definition path.

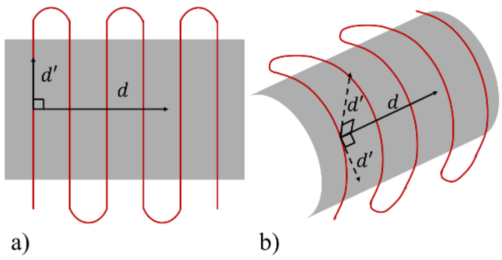


Fig. 5.1 Definition of sweep direction in different dimensions.

Another important aspect of 3D coverage path planning is the modeling technique, as it significantly impacts planning performance. While point clouds are easily accessible, they provide only coordinate information, necessitating additional computations to obtain other details, such as surface normals. In contrast, mesh models are self-contained, incorporating information about surface normals and connectivity. Moreover, additional processing techniques, such as semantic segmentation [79], can be readily implemented to divide the inspection target into application-specific components. This allows for flexible inspection paths to be designed specifically for each region, enhancing the overall effectiveness of the inspection process.

In this work, a parameterization-based inspection path planning algorithm is proposed that leverages the geometric information from a 3D mesh model to intuitively plan a sweep path. First, the 3D mesh model is transformed into a 2D UV map, allowing for straightforward sweep path planning without the complexities of a 3D environment. The planned 2D sweep path is then converted back into 3D using a barycentric transformation. Collisions are addressed through an optimization-based re-planning algorithm. Finally, a 2-opt algorithm is employed to eliminate path crossings and optimize the travel distance of the inspection path. The proposed planning approach is summarized in **Fig**. 5.2.

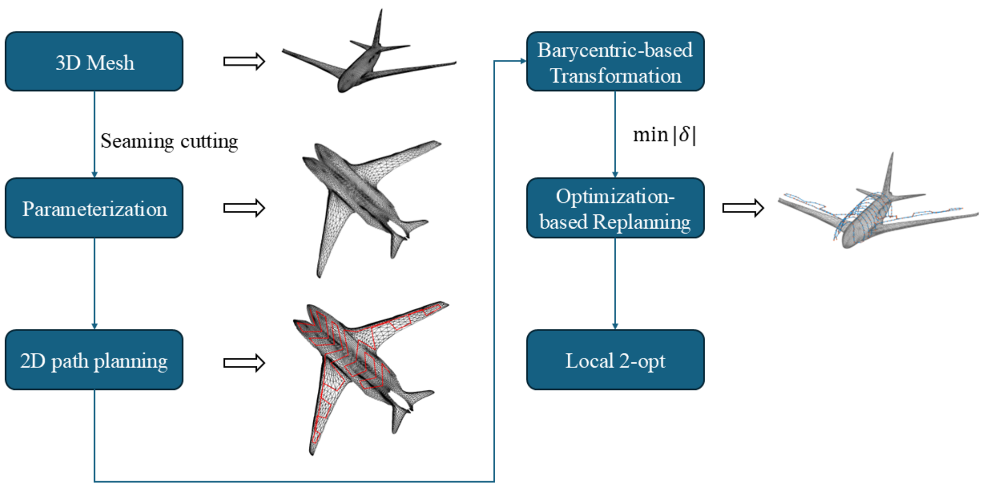


Fig. 5.2 Flow chart of the parameterization-based planning approach.

5.3.1 Parameterization

In graphic-related applications, parameterization refers to the process of computing a one-to-one mapping between two similar surfaces [80]. For mesh-related applications, the goal is to create a mapping to the parameter domain with minimal distortion. This process has numerous applications in computer graphics, including texture mapping, remeshing, and surface fitting. By transforming a complex 3D mesh model into a simplified 2D surface, various processing tasks can be performed more easily. One flexible method for achieving this transformation is seam selection, which involves choosing a set of cutting lines to dissect the 3D model. This allows the model to be converted into a single surface with minimal distortion. Additional information on mesh parameterization is summarized in [80-82]. In this section, the terms "2D surface" and "UV map" are used interchangeably, as both refer to the parameterized surface.

5.3.2 2D Sweep Path Planning

After obtaining the parameterized surface, a 2D sweep path can be easily planned. By reducing the dimensions of the area of interest, the sweep direction $\boldsymbol{d}$ and the corresponding path direction $\boldsymbol{d}'$ can be determined. By selecting an appropriate overlapping ratio for the specific inspection application, the sweep path on the parameterized surface can be generated. **Figure** 5.3 illustrates an example of a sweep path on the parameterized surface of the A380. Additionally, with the 2D map, further processing such as semantic segmentation can be conducted, allowing for task-specific planning.

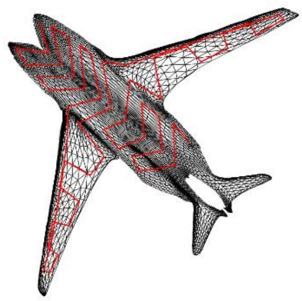


Fig. 5.3 Example sweep path on the UV map of an A380.

5.3.3 3D Sweep Path Generation

After obtaining the 2D sweep path, the next objective is to transform it back into 3D for execution by the UAV. Before discussing the transformation process, it is important to address path representation, as a low-definition path may lead to an infeasible 3D inspection path. If the 2D path is represented by only a few points, the transformed path may include segments that pass through the inspection target, as shown in **Fig.** 5.4. Therefore, the path's definition should be proportional to the size of the triangular patch of the inspection target, which can be defined as:

$$\alpha = \frac{\beta}{L} \times n \tag{5.1}$$

where β is the average triangle height, L is the average path segment length, and n is the number of points per triangle, and it is used to control the path's definition. An appropriate n should be selected to balance the path definition and the computational cost.

(a) (b)

Fig. 5.4 The effects of different path definition. (a) Low definition. (b) High definition.

The transformation aims to create a feasible inspection path while preserving the information of the original path. Specifically, it seeks to minimize distortion introduced during the transformation process. However, it is important to recognize that some distortion is unavoidable during parameterization and exists throughout the mapping. Consequently, transformation methods such as global interpolation can lead to accumulated distortion, resulting in poor performance. To address this, a local interpolation method is employed to minimize accumulated distortion. Specifically, a barycentric-based transformation is applied to the triangular mesh. In barycentric coordinates, a point's location is defined relative to the vertices of a triangle, allowing unique determination of any point within a triangular patch. Moreover, barycentric coordinates reference only the triangle's vertices, preventing distortion accumulation across the mapping. The barycentric-based transformation is summarized in Algo. 5.1.

Algorithm 5.1 Barycentric-based Transformation

Input: 2D sweep path $\boldsymbol{p}$, offset distance D, parameterized surface $\boldsymbol{A}$, 3D model $\boldsymbol{M}$, and surface normal $\boldsymbol{N}$

Output: 3D sweep path $\boldsymbol{P}$

for point $\leftarrow$ 1 to p **do**

$\quad$ $\mathrm{id} = \mathrm{Find}(\mathrm{point}, \boldsymbol{A})$

$\quad$ $x = \mathrm{UV2Bary}(\mathrm{point}, \boldsymbol{A}_{id})$

$\quad$ $x = \mathrm{Bary2XYZ}(\boldsymbol{M}_{id}, x)$

$\quad$ $X = \mathrm{Offset}(X, \boldsymbol{N}_{id}, D)$

$\quad$ $\boldsymbol{P} = \mathrm{Append}(\boldsymbol{P}, \boldsymbol{X})$

7: **end for**

8: **return *P***

For each point on the 2D sweep path, the transformation algorithm first identifies the corresponding triangular patch containing that point. The UV2Bary function then converts the point into barycentric coordinates by referencing the triangle's vertices, while the Bary2XYZ function transforms the barycentric coordinates into 3D coordinates based on the corresponding 3D vertex coordinates. The computed coordinates are then offset from the inspection target along the surface normal to ensure operational safety.

5.3.4 Point-wise Re-planning

The transformation is performed without collision checking, which may result in an inspection path containing self-intersections or infeasible segments. To ensure the feasibility of the inspection path, a collision check is conducted to identify potential hazards. For aircraft inspection, two primary hazards are considered: ground clearance and collision with the inspection target. The UAV must maintain a minimum altitude during operation to avoid collisions with ground objects or maintenance staff. Additionally, a safety distance should be established to ensure the UAV operates at a safe distance from the aircraft, preventing collisions. For each waypoint, its altitude and distance from the inspection target are calculated. If a waypoint fails to meet both requirements, it will be flagged for re-planning. To reduce unnecessary computations, the 3D space is discretized, allowing collision checks to be performed only with neighboring cells.

After identifying potential safety hazards, the re-planning algorithm is executed to adjust the waypoints. While this algorithm ensures operational safety, it can alter the information conveyed by the original planned 2D path. Relocating waypoints may change this information and potentially degrade the performance of the inspection path. To minimize the impact on inspection quality, only essential changes should be made. Therefore, to balance these objectives, an optimization-based re-planning algorithm is proposed.

As the re-planned waypoints deviate from the original, valuable information may be lost. Therefore, the optimization objective is to minimize the distance each re-planned waypoint is shifted. The formulated problem is presented as follows:

$$\min|\delta| \tag{5.2}$$

$$\delta = \boldsymbol{x_1} - \boldsymbol{x_2} \tag{5.3}$$

$$\boldsymbol{x_2} = \delta \times \boldsymbol{V} + \boldsymbol{x_1} \tag{5.4}$$

$$|\boldsymbol{x_2} - \boldsymbol{M}| > \epsilon \tag{5.5}$$

$$x_{23} > \mu \tag{5.6}$$

The objective (5.2) aims to minimize the distance changed μ for each re-planned waypoint, calculated by Eq. (5.3). Eq. (5.4) defines the coordinates of the re-planned waypoint as the current location plus the distance shifted along the escaping direction $\boldsymbol{V}$. Eq. (5.5) ensures that the safety distance ϵ is satisfied, while Eq. (5.6) addresses the ground clearance constraint. The escaping direction $\boldsymbol{V}$ is designed to guide the waypoint away from hazardous areas, specifically away from the inspection target. In a simplified scenario, the escaping direction can be represented by the surface normal of the nearest patch. However, in confined areas, the defined solution space may be infeasible, making the optimization problem unsolvable. Therefore, in the proposed method, $\boldsymbol{V}$ is represented by the combined surface normals from the neighboring discretized spaces, facilitating the re-planned waypoint's escape from confined areas. The proposed optimization problem is non-convex and is solved using a nonlinear optimizer.

While this work primarily focuses on improving the quality of the autonomous inspection task, small adjustments can still enhance inspection efficiency, particularly in single-agent scenarios. To eliminate self-intersections in the re-planned path, the 2-opt algorithm is employed to reduce the overall travel distance. However, to avoid excessive modifications that could alter the gathered information, the algorithm is executed only locally within a path segment to address local self-intersections.

5.3.5 Multi-UAV Aircraft Inspection

The proposed planning algorithm is integrated into the multi-UAV inspection framework outlined in Section 4, resulting in a multi-UAV aircraft inspection system. This system consists of several UAVs and a charging station from which the UAVs will operate. However, unlike in Section 4, candidate locations for the charging station are not selected from the waypoints. Instead, these candidate locations are uniformly sampled around the inspection target, as illustrated in **Fig.** 5.5. Initially, a random candidate is selected, and the system will then migrate toward the ideal charging station location that minimizes both the combined travel distance and the maximum distance among the UAVs.

After initializing the depot candidate, the corresponding MTSP is solved using the NSGACO. However, the use of high-density waypoints in the 3D sweep path can create an excessive computational load for the NSGACO. To make the number of waypoints manageable, the computed 3D sweep path is first down-sampled. The down-sampled waypoints, along with the depot candidate, are then input into the NSGACO to solve the MTSP. The algorithm will return the optimal depot location along with the waypoint assignments for each UAV.

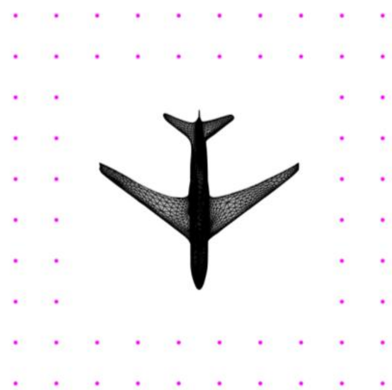


Fig. 5.5 Candidate charging station location for A380 inspection mission

5.3.6 Time complexity

Within the algorithm, the most computationally demanding processes are transformation and replanning. The time complexity of the transformation process is determined by the number of waypoints in the planned 2D path, represented as $O(n_x)$, where $n_x$is the number of waypoints. For the replanning module, the time complexity depends on the number of points requiring replanning, represented by n_{rx}. The optimization-based replanning module utilizes a nonlinear optimizer, with a time complexity of $O(n_{rx}^2)$. Therefore, the overall time complexity of the parameterization-based planning approach can be summarized as $O(n_x + n_{rx}^2)$.

5.3.7 Practicability

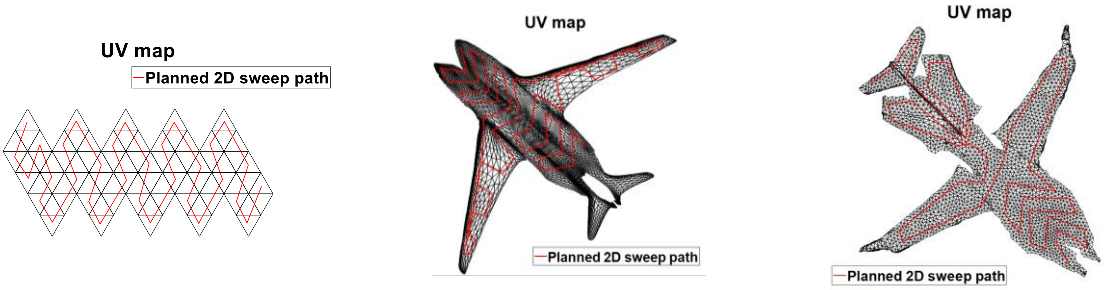
To implement the proposed planning approach in a real-world scenario, procedures similar to those used in previous work should be followed. Collision checks and local path planning algorithms, such as RRT, should be used collaboratively to ensure collision-free connections between waypoints. UAVs equipped with onboard cameras, flight controllers, and flight computers should be deployed to carry out the inspection mission. To fully utilize the information captured by the computed sweep path, video should be recorded instead of taking discrete photos. The recorded footage can then be sampled to obtain individual images for the reconstruction process.

5.4 Simulation Results and Discussions

Numerical and software-in-the-loop simulations are conducted to evaluate the performance of the proposed planning algorithm. Both single and multi-UAV scenarios are examined to assess the scalability of the approach.

5.4.1 Single UAV scenario

The single UAV scenario is discussed first, utilizing three inspection targets with different geometries for the numerical studies. The icosahedral sphere is a simple convex structure that should not present challenges to the planning algorithm, requiring only limited re-planning. The A380, with its complex concave structures, necessitates additional re-planning to relocate waypoints away from confined areas. Lastly, the Learjet 45 features two top-mounted engines, adding further challenges for the proposed planning algorithm. The UV map and the corresponding planned 2D path for the test cases are illustrated in **Fig.** 5.6. All test cases have a ground clearance constraint of 4 m, with designed inspection distances of 3 m, 10 m, and 3 m, respectively.



(a) (b) (c)

Fig. 5.6 The planned 2D path of the test cases. (a) Icosahedral sphere. (b) Airbus A380. (c) Learjet 45.

The viewing distance is used to estimate the quality of the inspection. A quality inspection should maintain a consistent viewing distance to ensure uniform coverage. The simulated results are summarized in Table 5.1.

Table 5.1 Summary of viewing distance of all test cases.

	Icosahedral sphere	A380	Learjet 45
Mean Inspection Distance (m)	2.97	9.87	2.77
Standard Deviation (m)	0.06	0.80	0.41

For the convex icosahedral sphere, the generated inspection path is illustrated in **Fig.** 5.7. A continuous inspection path is created for inspecting this 20-faced structure. The planned path adheres to the ground clearance constraint, with no self-intersections observed. Additionally, the inspection path maintains a consistent viewing distance with minimal deviation, as summarized in **Fig.** 5.8. Overall, promising results are demonstrated in the icosahedral sphere test case.

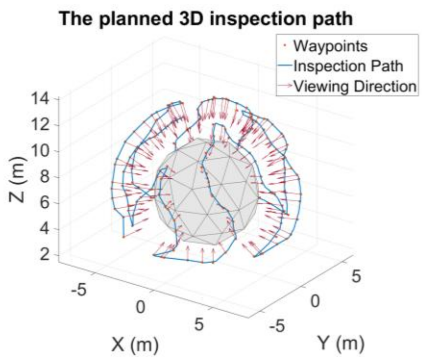


Fig. 5.7 The inspection path on the icosahedral sphere.

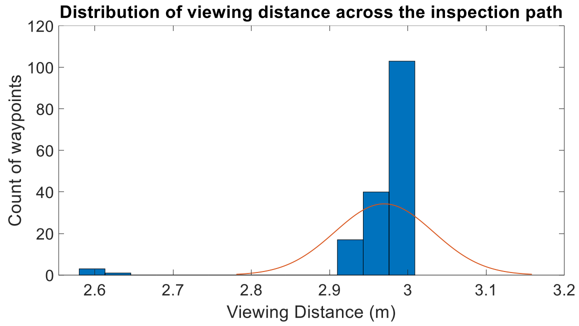


Fig. 5.8 The viewing distance distribution on the icosahedral sphere.

For the A380 test case, the inspection primarily focuses on the upper surface of the aircraft, so other structures, such as the engines and undercarriage, are removed to simplify the model. **Figure** 5.9 illustrates the generated 3D sweep path for the inspection task, while **Fig.** 5.10 summarizes the viewing distance distribution. Although the A380 presents additional challenges for the planning algorithm, it successfully generates a feasible inspection path. Furthermore, a consistent viewing distance is achieved, indicating a quality inspection and demonstrating the algorithm's capability in handling complex inspection tasks.

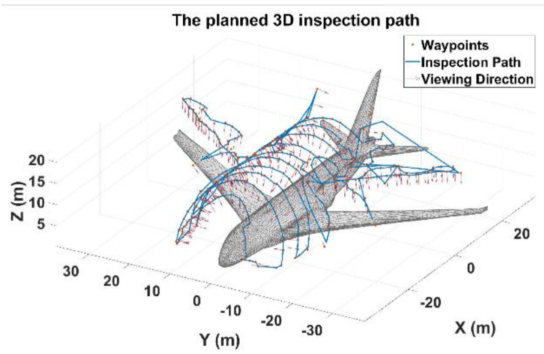


Fig. 5.9 The inspection path on the A380.

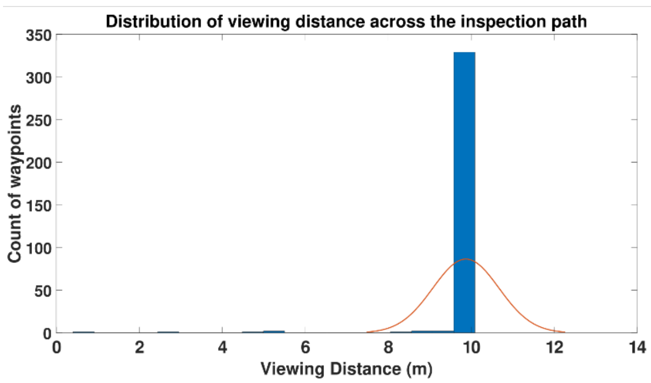


Fig. 5.10 The viewing distance distribution on the A380.

While the A380 presents additional challenges for the planning algorithm, its overall geometry remains relatively simple. Therefore, to further validate the performance of the proposed method, the Learjet 45 is used as the final test case. With its top-mounted engines, the Learjet 45 features both convex and concave elements that challenge the re-planning capabilities of the algorithm. The planned inspection path aims to cover most of the upper surface of the aircraft, including the fuselage, wings, and T-tail. The resulting inspection path is illustrated in **Fig.** 5.11. In this test case, the planning algorithm successfully generates a self-intersection-free inspection path. Despite the added complexities, it maintains a consistent viewing distance, as shown in **Fig.** 5.12. In summary, the numerical studies of the test cases validate the planning ability and robustness of the proposed approach.

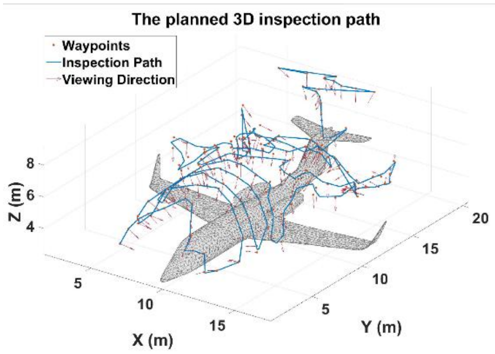


Fig. 5.11 The inspection path on Learjet 45.

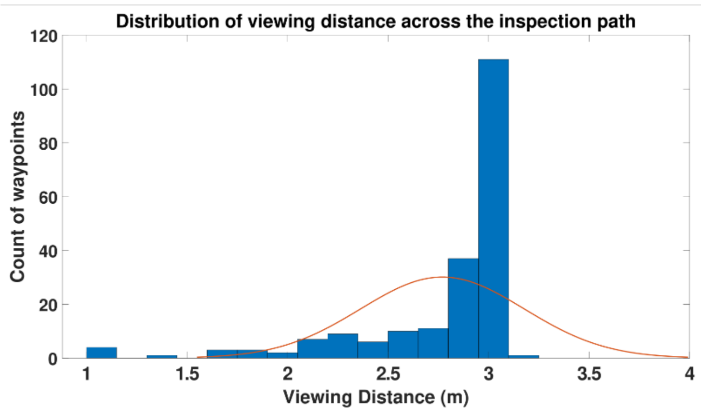
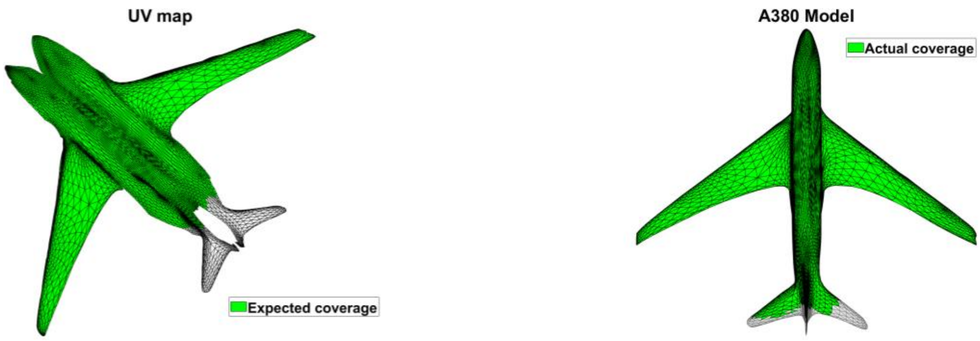


Fig. 5.12 The viewing distance distribution on Learjet 45.

The major modules in the proposed planning approach are the transformation and re-planning algorithms, which are responsible for preserving the information conveyed by the original 2D path. To evaluate their performance, the information loss during the process is quantified. Specifically, the coverage percentage and Hausdorff distance are used as the performance metric. The coverage percentage measures how well the expected coverage is being represented in the actual coverage, while the Hausdorff distance measures the deviation between the two. The actual coverage is calculated in 3D space, while the expected coverage is derived from the 2D sweep path, assuming the UAV flies directly above the parameterized surface with its camera facing downward. If a surface patch meets both the viewing distance and angle constraints, it is considered visible. This assumption disregards the effects of surface curvature in the 3D scenario, providing the maximum possible coverage to serve as a baseline for assessing information loss during the transformation process. Table 5.2 summarizes the performance across the aircraft test cases, and visual comparisons are illustrated in **Fig.** 5.13 and 5.14.

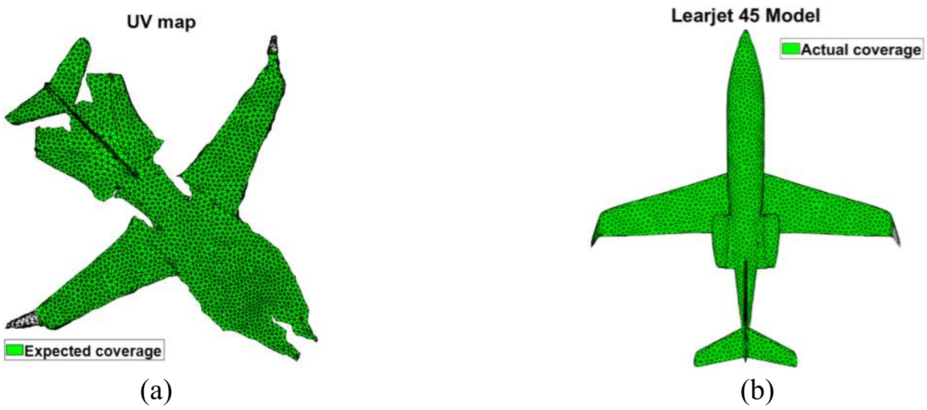
Table 5.2 Coverage percentage of the A380 and Learjet 45 test cases.

	A380	Learjet 45
Coverage Percentage	94%	88%
Hausdorff Distance	43	23



(a) (b)

Fig. 5.13 The computed coverage of the A380. (a) Expected coverage. (b) Actual coverage.



(a) (b)

Fig. 5.14 The computed coverage of the Learjet 45. (a) Expected coverage. (b) Actual coverage.

The test cases yield promising results, with coverage percentages exceeding 85% across all scenarios. The actual coverage represents less of the expected coverage on the Learjet 45 due to its complex geometry; however, the coverage percentage remains within a reasonable range. Regarding the deviation between the two coverages, the Hausdorff distance remains small in both cases when compared to the complexity of the inspection target, which consists of thousands of surface patches. The results are further validated through visualizations of the coverages, demonstrating that the actual coverages closely resemble the expected coverages in both instances. These findings indicate that the proposed transformation and re-planning scheme effectively converts the 2D path into 3D while minimizing information loss during the process.

To evaluate the performance of the proposed algorithm in realistic scenarios, software-in-the-loop simulations are conducted. To further demonstrate improvements in inspection quality, the proposed approach is compared with a sampling-based technique. In the sampling-based approach, a safe sampling space is first defined by dilating the inspection target by a specified operational distance. Next, viewpoints are randomly sampled within this space, and their visibility information is estimated accordingly. The sampling process continues until the coverage requirements are met. To eliminate redundant viewpoints, an optimization problem, such as the SCP, is solved to minimize the number of viewpoints used. Finally, an efficient route guiding the UAV through the viewpoints is obtained by solving the TSP. Similar framework had been used in [16-19 23, 24]. The simulations focus on the aircraft test cases, with photogrammetry selected as the inspection method. The inspection path generated by the sampling technique is illustrated in **Fig.** 5.15.

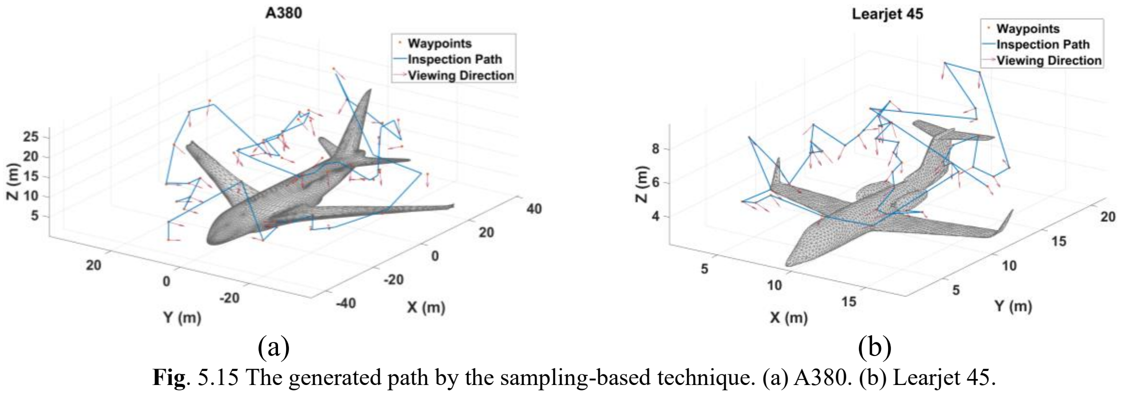
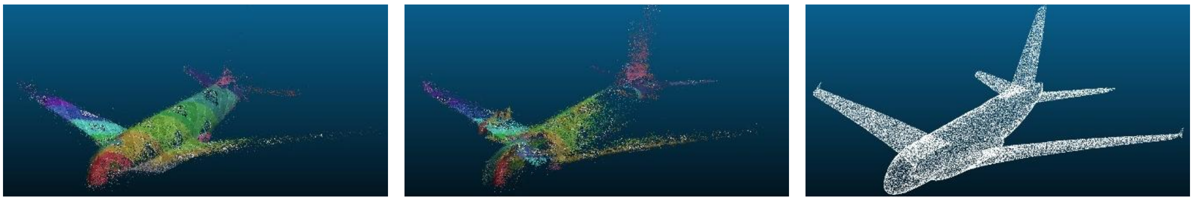


Fig. 5.15 The generated path by the sampling-based technique. (a) A380. (b) Learjet 45.

Sampling-based techniques are widely used in autonomous inspection, providing a well-established framework that serves as a benchmark for evaluating the performance of other coverage path planning methods. Despite its sampling nature, this method also falls under the category of model-based approaches, as it utilizes geometric information from the inspection target to define the sampling space. Consequently, the sampling-based approach serves as an authoritative benchmark for assessing the performance of the parameterization-based planning approach.

The reconstruction results of the algorithms are illustrated in **Fig.** 5.16 and 5.17. To enhance the reconstruction process, a colored grid texture is applied to the model. Comparing the generated point clouds reveal that the result from the proposed approach is significantly denser than that of the sampling method. Additionally, the reconstructed model from the proposed approach exhibits much less noise and fewer false reprojections, demonstrating that a higher quality result can be achieved using this algorithm.



(a) (b) (c)

Fig. 5.16 Pointclouds of the A380. (a) Proposed method. (b) Sampling method. (c) Reference

(a) (b) (c)

Fig. 5.17 Pointclouds of the Learjet 45. (a) Proposed method. (b) Sampling method. (c) Reference

In addition to qualitative analysis, quantitative analysis is conducted to evaluate the quality of the reconstruction results. Specifically, the cloud-to-cloud (C2C) distance is used to measure the deviation of the reconstruction from the reference point cloud, with a smaller C2C distance being preferred. The results are summarized in Table 5.3. Across the test cases, the proposed method achieves a smaller average C2C distance compared to the sampling approach, leading to a higher quality model.

Furthermore, the standard deviation of the C2C distance is also improved relative to the sampling method, indicating that the proposed approach not only produces a more accurate model but also demonstrates significantly more consistent performance.

Table 5.3 C2C distance of the aircraft test cases.

	Proposed method		Sampling method	
	A380	Learjet 45	A380	Learjet 45
Mean distance (m)	**0.2**	**0.0057**	0.49	0.07
Standard deviation (m)	**0.23**	**0.06**	0.51	0.09

To investigate the reasons behind these improvements, the viewing distances of the two algorithms are compared, with the results summarized in **Fig.** 5.18. Across the test cases, the proposed planning algorithm achieves a much more consistent viewing distance along the inspection path, while the viewing distance of the sampling method varies significantly. Photogrammetry relies on feature matching across images to reconstruct the 3D model. If images are taken at inconsistent distances, the reconstruction process may struggle to determine the exact dimensions and geometry of the target object, leading to potential scale ambiguity and false reprojection. Therefore, by carefully controlling the viewing distance along the inspection path, the proposed approach ensures a consistent viewing distance, providing optimal image quality for the reconstruction process and ultimately improving the quality of the reconstructed model.

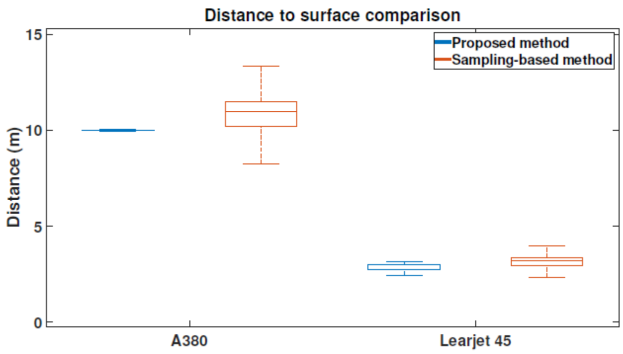


Fig. 5.18 Viewing distance comparison of the two methods.

To demonstrate the practicality of the proposed algorithm, a real-world experiment was conducted, with bridge inspection selected as the target application. The UAV was tasked with inspecting the surfaces of a bridge pillar. It was equipped with an onboard camera capable of capturing high-definition images of the pillar's surface, as well as a LiDAR system to obtain an updated 3D model of the pillar. Apart from dynamic obstacle avoidance, the onboard camera is also used from real-time crack detection. Figure 5.19 illustrates the planned path for the inspection task, while **Fig**. 5.20 shows the experimental scene

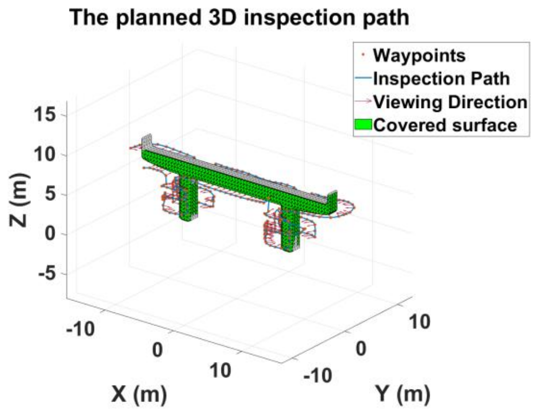


Fig. 5.19 Planned inspection path for the bridge pillar inspection.

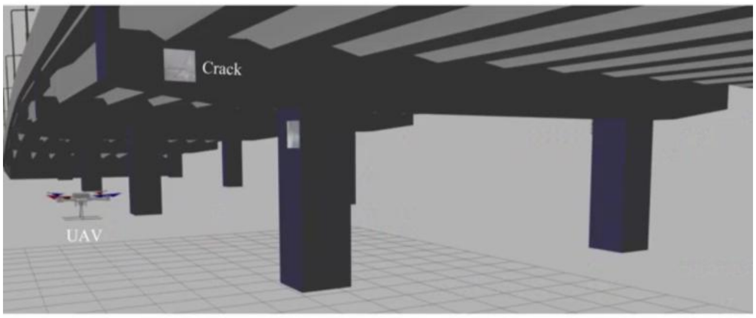


Fig. 5.20 The experiment scene.

The UAV navigates through the computed path efficiently. The onboard camera successfully detects cracks on the pillar surfaces, as shown in **Fig**. 5.21. The scene reconstruction results are summarized in **Fig**. 5.22. Additionally, the onboard LiDAR captures an accurate point cloud of the bridge pillar, which can be used for future inspection missions. This real-world experiment demonstrates the performance and practicality of the proposed planning approach.

Fig. 5.21 Crack detection performance during the experiment.

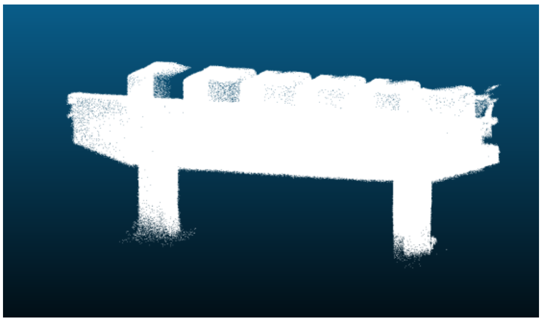


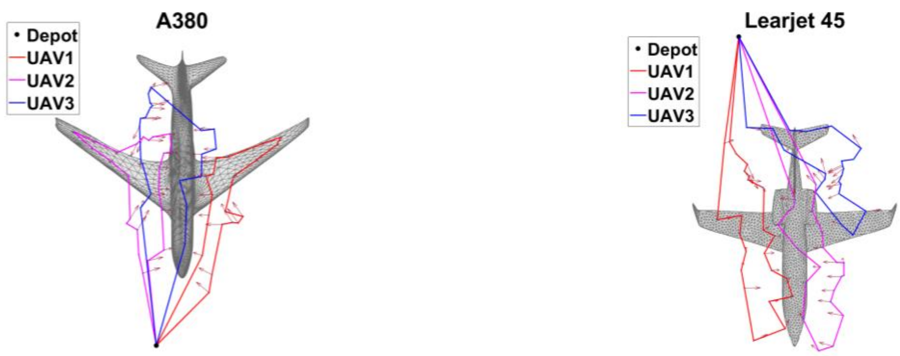
Fig. 5.22 The reconstructed scene.

5.4.2 Multi-UAV scenario

To demonstrate the scalability of the proposed planning algorithm, its performance in a multi-UAV scenario is studied. Similarly, the proposed algorithm is compared with a sampling-based technique. Since parameterization-based path planning was initially developed for single UAV inspection tasks, modifications are necessary to adapt the method for multi-UAV scenarios. To streamline the computation of the NSGACO, the number of waypoints representing the inspection path is first reduced. After obtaining the 3D inspection path, it is simplified through sampling. The reduced waypoints are then input into the NSGACO to compute the multi-UAV inspection path. While the sweep path planned in 2D cannot be preserved, the waypoints should adequately represent the original path, maintaining the objective of improving inspection quality. In the sampling-based approach, viewpoints are randomly sampled, and the previously proposed NSGACO is used to assign these viewpoints to each UAV. To ensure fairness between the planning approaches, the same depot location is utilized.

Figure 5.23 illustrates the computed inspection path of the proposed method and the corresponding charging station location for each test case. By minimizing the combined traveling distance and aligning the depot with this objective, the areas of interest among the UAVs are mostly separated, with each UAV responsible for a specific section of the aircraft. Generally, the areas of interest can be divided into the left wing, right wing, and fuselage. Without semantic segmentation of the inspection target, effective work division can be achieved using the proposed optimization problem. Additionally, minimizing overlap between areas of interest significantly reduces the possibility of collisions among the UAVs, facilitating the collision avoidance process during execution.

Before the inspection mission commences, the UAVs will be stationed at the charging station, separated by one meter. Once the mission is initiated, the UAVs will depart from the charging station and navigate towards their assigned viewpoints. After visiting all designated viewpoints, the UAVs will return to the charging station. Table 5.4 summarizes the simulation results of the aircraft test cases for the proposed method.



(a) (b)

Fig. 5.23 Computed path of the two inspection missions.

Table 5.4 Performances of the algorithms in multi-UAV scenario.

Parameter	A380	Learjet 45
UAV1 Traveling Distance (m)	180	59
UAV2 Traveling Distance (m)	192	60
UAV3 Traveling Distance (m)	192	49
Total Traveling Distance (m)	566	169
Inspection Time	6 mins 34 s	3 mins 09 s

Table 5.5 Performance comparison on single and multi-UAV scenarios.

	A380		Learjet 45	
Parameter	Single UAV	3 UAVs	Single UAV	3 UAVs
Inspection time	17 mins 17 s	6 mins 34 s	4 mins 50 s	3 mins 09 s
Average Traveling Distance (m)	/	188	/	56
Maximum Traveling Distance (m)	858	192	180	60

By minimizing the overlap between the AOIs, the traveling distance for each UAV is also reduced. Additionally, by minimizing the maximum traveling distance among the UAVs, their travel distances become more balanced, which reduces the variation in mission times and ultimately decreases the overall inspection time. The simulation results further reinforce the findings from Section 4, demonstrating the benefits of solving the MO problem.

The performance of single UAV versus multi-UAV scenarios is summarized in Table 5.5. Utilizing multiple UAVs allows for the distribution of workload, reducing the traveling distance for each UAV and significantly decreasing the overall inspection time. However, diminishing returns are observed in scenarios with shorter distances, such as the Learjet 45 test case. As more UAVs are deployed, the number of redundant paths in the solution increases, inevitably raising the overall traveling distance. Therefore, the number of UAVs should be adjusted based on the scale of the inspection task.

The improvement in inspection quality is studied using software-in-the-loop simulations. Photogrammetry is again selected as the inspection method, and both qualitative and quantitative analyses are performed. **Figures** 5.24 and 5.25 illustrate the reconstruction results of the proposed method and the sampling-based method, while Table 5.6 summarizes the results of the quantitative analysis.

(a) (b) (c)

Fig. 5.24 Pointclouds of the A380 in multi-UAV scenario. (a) Proposed method. (b) Sampling method. (c) Reference.


(a) (b) (c)
Fig. 5.25 Pointclouds of the Learjet 45 in multi-UAV scenario. (a) Proposed method. (b) Sampling method. (c) Reference.

Qualitative analysis reveals that the quality of the reconstructed model using the proposed method outperforms that of the sampling-based method. First, the proposed method yields a reconstructed model with less noise, as demonstrated by the sharp outlines of the point clouds. Additionally, false reprojections are absent in the proposed method, while they appear in the results of the sampling method, leading to larger reconstruction errors. Further quantitative analysis confirms these observations. The average error of the proposed method in both test cases is significantly smaller than that of the sampling method. Moreover, the standard deviation of the error is also much lower in the proposed method, indicating that the planning algorithm can achieve more consistent performance even in multi-UAV scenarios. Although the original 2D sweep path is not preserved, the sampled viewpoints adequately represent the original path, allowing the quality information carried by the original path to be retained in the multi-UAV inspection path. The simulation results demonstrate the scalability of the proposed planning approach, as well as the effectiveness of the MO-MTSP and NSGACO in realistic scenarios.

Table 5.6 Performances of the algorithms in multi-UAV scenario.

	Proposed method		Sampling method	
	A380	Learjet 45	A380	Learjet 45
Mean distance (m)	**0.18**	**0.04**	0.45	0.05
Standard deviation (m)	**0.19**	**0.05**	0.58	0.08

5.5 Conclusion

In this section, a parameterization-based inspection path planning algorithm is proposed. The approach begins by transforming the 3D model of the inspection target into a corresponding 2D surface. A sweep path is then planned on this 2D surface. A barycentric-based transformation and an optimization-based re-planning algorithm are introduced to convert the 2D path back into 3D while minimizing distortion during the process. To enhance the efficiency of the planned inspection, the algorithm is further extended for multi-UAV applications by integrating it with the multi-UAV system proposed in Section 4. Numerical simulations verify the feasibility of the computed inspection paths, and the performance of the transformation and re-planning algorithm is validated by analyzing the coverage percentage. In software-in-the-loop simulations, the proposed algorithm outperforms the sampling-based approach in terms of inspection quality, and the scalability of the planning algorithm is demonstrated through multi-UAV simulations. Real world experiment is also conducted to demonstrate the practicability of the proposed algorithm.

6. Conclusion and future work

6.1 Conclusion

In this thesis, a series of autonomous inspection approaches using UAVs have been proposed. The different approaches aim to improve different aspects of the autonomous task such that autonomous inspection using UAVs can be more comprehensive and efficient, accelerating its implementation in the emerging low-altitude economy. The scalability of the inspection task is first discussed, and a mixed viewpoint generation approach is proposed to improve this aspect of the task. By using the mixed viewpoint generation technique, suitable inspection paths for different inspection purposes can be generated using a single path planning approach, eliminating the need for additional planning approaches. The proposed optimization problem also generates a more kinodynamically feasible inspection path, reducing the traversal difficulty during the execution.

To further improve the efficiency of the autonomous inspection task, a multi-UAV approach is proposed. The proposed scenario also recognizes the importance of the location of the charging station of the multi-UAV system. Hence, it has also been included in the proposed optimization problem. A heuristic approach, the NSGACO is proposed to solve the MO problem, and the algorithm has been proven to outperform other heuristic algorithms. A migration algorithm has also been proposed to search for the optimal location of the charging station, and its effectiveness has been proven through numerical simulations.

Lastly, the most important aspect of autonomous inspection tasks, the inspection quality, is discussed, and a parameterization-based path planning approach has been proposed to improve this aspect. The proposed approach first parameterized the inspection target into a corresponding 2D surface and sweep path can then be planned intuitively. A barycentric-based transformation and optimization-based re-planning approach are used to transform the 2D path back into 3D while minimizing the information loss during the process. The feasibility of the inspection path is validated through numerical and software-in-the-loop simulations. The simulations result also demonstrate the superiority of the proposed approach in terms of inspection quality and a more accurate reconstructed model can be obtained compared to the sampling-based approach. The proposed approach also provides a new perspective for researchers to solve the 3D CPP, establishing a connection between the 2D and 3D variant of the problem. Furthermore, to demonstrate the scalability of the parameterization-based path planning approach, the planning approach is combined with the multi-UAV system proposed. In the multi-UAV scenario, the improvement in inspection quality is preserved while the efficiency of the task is greatly improved.

6.2 Future Work

To improve the comprehensiveness of the autonomous inspection task, several other research directions can be investigated.

6.2.1 Semantic segmentation

First, as mentioned during Section 5, additional preprocessing approaches such as semantic segmentation can be used to develop task-oriented planning such that the inspection approach is more suited for specific inspection purposes. Currently, the planning approach only aims to cover the targeted area without any task-specific work division. By utilizing semantic segmentation, the targeted area can be separated logically and task-oriented work division among the UAVs can be achieved. The collected information can then be more organized, further streamlining the implementation of UAVs in autonomous inspection tasks.

6.2.2 Adaptive planning

Currently, the planning approaches assume that the inspection tasks are performed in an ideal environment where visual occlusion is not considered. However, in a realistic scenario, the inspection mission would encounter several deployment challenges. First, sensor noise and visual occlusion caused by external objects could affect the actual coverage of the inspection task, leading to further deviation from the expected coverage. To ensure that the collected data meets the coverage requirements, adaptive planning can be implemented. During the inspection task, the ground computer can monitor the current coverage and dynamically generate viewpoints to cover the occluded areas. By utilizing dynamic feedback, a more comprehensive inspection can be achieved. Second, dynamic obstacles may exist within the environment, presenting additional safety hazards to the inspection task. To address this, dynamic planning can be integrated into the proposed algorithms, enabling effective collision avoidance. A depth camera can be utilized for obstacle identification, ensuring accurate dynamic collision avoidance during the inspection process.

6.2.3 2D-3D CPP unionization

As mentioned in Section 5, the proposed parameterization-based planning approach establishes a connection between the 2D and 3D planning problems. Further investigation is warranted to study the feasibility of transformation techniques in solving 3D CPP using 2D CPP algorithms. Traditionally, 2D and 3D CPP are treated as distinct research fields, with different algorithms employed in each area. Knowledge is rarely shared between these related topics due to the lack of a connection between the two fields. However, if the transformation approach proves feasible and various 2D CPP algorithms can be adapted to 3D without significant distortion, a new research area may emerge, providing a novel perspective for researchers to address these problems. The unionization of 2D and 3D CPP could standardize path planning approaches across robotics domains, enabling future CPP solutions to tackle both 2D and 3D challenges simultaneously.

References

[1] H. Shakhatreh et al., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," in IEEE Access, vol. 7, pp. 48572-48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[2] C. Stef, “How Building Age Impacts Rent in NYC,” Multi-Housing News, Aug. 28, 2017. [Online]. Available: https://www.multihousingnews.com/how-building-age-impacts-rent-in-nyc/ . [Accessed: May 2, 2025].

[3] Y. Scholars, “Rebuilding Hong Kong’s Old Building Stock: Trends & Impact,” ArcGIS StoryMaps, Mar. 06, 2024. [Online]. Available: https://storymaps.arcgis.com/stories/4894d232d9f242129087371fd0e23643 . [Accessed: May 2, 2025].

[4] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” Robotics and Autonomous Systems, vol. 61, no. 12, pp. 1258–1276, Dec. 2013, doi: 10.1016/j.robot.2013.09.004.

[5] T. Rakha and A. Gorodetsky, “Review of Unmanned Aerial System (UAS) Applications in the Built Environment: Towards Automated Building Inspection Procedures Using Drones,” Automation in construction, vol. 93, pp. 252–264, 2018.

[6] I. Bartoli, A. Kontsos, A. Pradhan, and A. Ellenberg, “Masonry Crack Detection Application of an Unmanned Aerial Vehicle,” in Computing in Civil and Building Engineering, 2014, pp. 1788–1795.

[7] Zulgafli, M.N. and Tahar, Khairul Nizam, “Three dimensional curve hall reconstruction using semi-automatic UAV,” ARPN Journal of Engineering and Applied Sciences 12, no. 10, pp. 3228-3232, 2017.

[8] A. Colantonio and G. McIntosh, “The Differences Between Large Buildings and Residential Infrared Thermographic Inspections Is Like Night and Day,” 2015. https://www.semanticscholar.org/paper/The-Differences-Between-Large-Buildings-and-Is-Like-Colantonio-McIntosh/eb9f6fdf23e1c7715af475ce8752cd55c9cb9ef7#citing-papers [Accessed Dec. 12, 2025].

[9] D. González-Aguilera, S. Lagüela, P. Rodríguez-Gonzálvez, and D. Hernández-López, “Image-

based thermographic modeling for assessing energy efficiency of buildings façades," Energy and Buildings, vol. 65, pp. 29–36, Oct. 2013, doi: https://doi.org/10.1016/j.enbuild.2013.05.040.

[10] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," SN Applied Sciences, vol. 1, no. 8, pp. 1–24, 2019.

[11] K. Alexis, C. Papachristos, R. Siegwart and A. Tzes, "Uniform Coverage Structural Inspection Path–planning for Micro Aerial Vehicles," 2015 IEEE International Symposium on Intelligent Control (ISIC), 2015, pp. 59-64.

[12] W. R. Scott, "Model-based view planning," Machine Vision and Applications, vol. 20, no. 1, pp. 47–69, 2007.

[13] B. Englot and F. S. Hover, "Sampling-based Sweep Planning to Exploit Local Planarity in the Inspection of Complex 3D Structures," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4456-4463.

[14] B. Englot and F. S. Hover, "Three-dimensional coverage planning for an underwater inspection robot," The International Journal of Robotics Research, vol. 32, no. 9–10, pp. 1048–1073, Aug. 2013, doi: https://doi.org/10.1177/0278364913490046.

[15] G. Tarbox and S. Gottschlich, "Planning for Complete Sensor Coverage in Inspection," Computer vision and image understanding, vol. 61, no. 1, pp. 84–111, 1995.

[16] W. Jing, J. Polden, W. Lin and K. Shimada, "Sampling-based View Planning for 3D Visual Coverage Task with Unmanned Aerial Vehicle," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 1808-1815.

[17] W. Jing and K. Shimada, "Model-based view planning for building inspection and surveillance using voxel dilation, Medial Objects, and Random-Key Genetic Algorithm," Journal of Computational Design and Engineering, vol. 5, no. 3, pp. 337–347, Dec. 2017, doi: 10.1016/j.jcde.2017.11.013.

[18] W. Jing, D. Deng, Z. Xiao, Y. Liu, and K. Shimada, "Coverage Path Planning using Path Primitive Sampling and Primitive Coverage Graph for Visual Inspection," Nov. 2019, doi:

10.1109/iros40897.2019.8967969.

[19] W. Jing, D. Deng, Y. Wu and K. Shimada, "Multi-UAV Coverage Path Planning for the Inspection of Large and Complex Structures," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 1480-1486, doi: 10.1109/IROS45743.2020.9341089.

[20] S. Song and S. Jo, "Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 6217-6224, doi: 10.1109/ICRA.2017.7989737.

[21] E. Palazzolo and C. Stachniss, "Effective Exploration for MAVs Based on the Expected Information Gain," Drones, vol. 2, no. 1, p. 9, Mar. 2018, doi: 10.3390/drones2010009.

[22] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias and G. Cai, "GPU accelerated coverage path planning optimized for accuracy in robotic inspection applications," 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), Abu Dhabi, United Arab Emirates, 2016, pp. 1-4, doi: 10.1109/MWSCAS.2016.7869968.

[23] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri and L. Seneviratne, "Coverage Path Planning with Adaptive Viewpoint Sampling to Construct 3D Models of Complex Structures for the Purpose of Inspection," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 7047-7054, doi: 10.1109/IROS.2018.8593719.

[24] R. Almadhoun, T. Taha, J. Dias, L. Seneviratne, and Yahya Zweiri, “Coverage Path Planning for Complex Structures Inspection Using Unmanned Aerial Vehicle (UAV),” Lecture notes in computer science, pp. 243–266, Jan. 2019, doi: 10.1007/978-3-030-27541-9_21.

[25] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," Control Eng. Pract., vol. 74, pp. 118–131, May 2018, doi: 10.1016/j.conengprac.2018.03.002.

[26] I. Z. Biundini, M. F. Pinto, A. G. Melo, A. L. M. Marcato, L. M. Honório, and M. J. R. Aguiar, "A framework for coverage path planning optimization based on point cloud for structural inspection," Sensors, vol. 21, no. 2, p. 570, Jan. 2021, doi: 10.3390/s21020570.

[27] O. Cheikhrouhou and I. Khoufi, “A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy,” Computer Science Review, vol. 40, p. 100369, May 2021, doi: 10.1016/j.cosrev.2021.100369.

[28] R. M. F. Alves and C. R. Lopes, "Using genetic algorithms to minimize the distance and balance the routes for the multiple Traveling Salesman Problem," 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 2015, pp. 3171-3178, doi: 10.1109/CEC.2015.7257285.

[29] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, pp. 3–34, 2011, doi: 10.1007/978-0-85729-652-8_1.

[30] Y. Shuai, S. Yunfeng, and Z. Kai, “An effective method for solving multiple travelling salesman problem based on NSGA-II,” Systems Science & Control Engineering, vol. 7, no. 2, pp. 108–116, Oct. 2019, doi: 10.1080/21642583.2019.1674220.

[31] Z. Lu, K. Zhang, J. He, and Y. Niu, “Applying K-means Clustering and Genetic Algorithm for Solving MTSP,” Communications in Computer and Information Science, pp. 278–284, 2016, doi: 10.1007/978-981-10-3614-9_34.

[32] Z. Wang, X. Fang, H. Li and H. Jin, "An Improved Partheno-Genetic Algorithm With Reproduction Mechanism for the Multiple Traveling Salesperson Problem," in IEEE Access, vol. 8, pp. 102607-102615, 2020, doi: 10.1109/ACCESS.2020.2998539.

[33] H. Zhou, M. Song, and W. Pedrycz, “A comparative study of improved GA and PSO in solving multiple traveling salesmen problem,” Applied Soft Computing, vol. 64, pp. 564–580, Mar. 2018, doi: 10.1016/j.asoc.2017.12.031.

[34] M. Dorigo and L. M. Gambardella, “Ant colonies for the travelling salesman problem,” Biosystems, vol. 43, no. 2, pp. 73–81, Jul. 1997, doi: 10.1016/s0303-2647(97)01708-5.

[35] R. Zhan, P. Shao, K. Tang, Y. Zhang, and L. Wang, “MTSP-ACA for Multi-UAV Path Planning Applied in SAR Operations in Mountainous Terrain,” Lecture notes in electrical engineering, pp. 3260–3268, Jan. 2022, doi: 10.1007/978-981-16-9492-9_320.

[36] S. Wang et al., "Cooperative Task Allocation for Multi-Robot Systems Based on Multi-

Objective Ant Colony System," in IEEE Access, vol. 10, pp. 56375-56387, 2022, doi: 10.1109/ACCESS.2022.3165198.

[37] H. Zhao and C. Zhang, "An ant colony optimization algorithm with evolutionary experience-guided pheromone updating strategies for multi-objective optimization," Expert Systems with Applications, vol. 201, p. 117151, Sep. 2022, doi: 10.1016/j.eswa.2022.117151.

[38] X. Chen, P. Zhang, G. Du and F. Li, "Ant Colony Optimization Based Memetic Algorithm to Solve Bi-Objective Multiple Traveling Salesmen Problem for Multi-Robot Systems," in IEEE Access, vol. 6, pp. 21745-21757, 2018, doi: 10.1109/ACCESS.2018.2828499.

[39] T. Lai, H. Li, H. Sun, Z. Wang, and C. Chen, "A Novel Multi-objective UAV Trajectory Planning Method Based on Modified BSO Algorithm," Lecture notes in electrical engineering, pp. 1244–1252, Jan. 2022, doi: 10.1007/978-981-16-9492-9_124.

[40] X. Peng, F. Guan, Z. Wang, and S. Gao, "Simulation and Optimization of Multi-UAV Route Planning Based on Hybrid Particle Swarm Optimization," Advances in intelligent systems and computing, pp. 223–232, Jan. 2021, doi: 10.1007/978-981-33-4575-1_22.

[41] W. -W. Qian, X. Zhao and K. Ji, "Region Division in Logistics Distribution With a Two-Stage Optimization Algorithm," in IEEE Access, vol. 8, pp. 212876-212887, 2020, doi: 10.1109/ACCESS.2020.3040004.

[42] I. Vandermeulen, R. Groß, and A. Kolling, "Balanced Task Allocation by Partitioning the Multiple Traveling Salesperson Problem," in Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Canada, 2019, pp. 1479-1487, doi: 10.5555/3306127.3331861.

[43] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," Operations Research, vol. 21, no. 2, pp. 498–516, Apr. 1973, doi: 10.1287/opre.21.2.498.

[44] C. Jiang, Z. Wan, and Z. Peng, "A new efficient hybrid algorithm for large scale multiple traveling salesman problems," Expert Systems with Applications, vol. 139, p. 112867, Jan. 2020, doi: 10.1016/j.eswa.2019.112867.

[45] Y. R. Cui, "Study of Parameters Configuration in Ant Colony Algorithm," Applied Mechanics

and Materials, vol. 678, pp. 51–54, Oct. 2014, doi: 10.4028/www.scientific.net/amm.678.51.

[46] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem," Applied Soft Computing, vol. 30, pp. 484–490, May 2015, doi: 10.1016/j.asoc.2015.01.068.

[47] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," Applied Soft Computing, vol. 107, p. 107439, Aug. 2021, doi: 10.1016/j.asoc.2021.107439.

[48] H. M. Botee and E. Bonabeau, "Evolving Ant Colony Optimization," Advances in Complex Systems, vol. 01, no. 02n03, pp. 149–159, Jun. 1998, doi: 10.1142/s0219525998000119.

[49] S. de, S. Pires, and B. de, "Genetic and Ant Colony Algorithms to Solve the Multi-TSP," Lecture Notes in Computer Science, pp. 324–332, Jan. 2021, doi: 10.1007/978-3-030-91608-4_32.

[50] Ana Maria Nogareda, Javier Del Ser, Eneko Osaba, and D. Camacho, "On the design of hybrid bio-inspired meta-heuristics for complex multiattribute vehicle routing problems," Expert Systems, vol. 37, no. 6, Jan. 2020, doi: 10.1111/exsy.12528.

[51] R. B. Erkartal, Ömer Çetin, and A. Yılmaz, "Data Collection from Wireless Sensor Networks: OpenMP Application on the Solution of Traveling Salesman Problem with Parallel Genetic Algorithm and Ant Colony Algorithm", JAST, vol. 15, no. 2, pp. 108–124, Jul. 2022.

[52] Y. Liu et al., "A scheduling route planning algorithm based on the dynamic genetic algorithm with ant colony binary iterative optimization for unmanned aerial vehicle spraying in multiple tea fields," vol. 13, Sep. 2022, doi: 10.3389/fpls.2022.998962.

[53] G. Dong, W. W. Guo, and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems," Expert Systems with Applications, vol. 39, no. 5, pp. 5006–5011, Apr. 2012, doi: 10.1016/j.eswa.2011.10.012.

[54] S. Gao, Z. Zhang, and C. Cao, "A Novel Ant Colony Genetic Hybrid Algorithm," Journal of Software, vol. 5, no. 11, Nov. 2010, doi: 10.4304/jsw.5.11.1179-1186.

[55] D. Gong and X. Ruan, "A hybrid approach of GA and ACO for TSP," Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788), Hangzhou, China, 2004, pp. 2068-

2072 Vol.3, doi: 10.1109/WCICA.2004.1341948.

[56] D. M. Chitty, "An Ant Colony Optimisation Inspired Crossover Operator for Permutation Type Problems," 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 2021, pp. 57-64, doi: 10.1109/CEC45853.2021.9504893.

[57] B. Zhang, T. Jing, X. Lin, Y. Cui, Y. Zhu, and Z. Zhu, “Deep Reinforcement Learning-based Collaborative Multi-UAV Coverage Path Planning,” Journal of physics. Conference series, vol. 2833, no. 1, p. 12017, 2024, doi: 10.1088/1742-6596/2833/1/012017

[58] W. Hu et al., "Multi-UAV Coverage Path Planning: A Distributed Online Cooperation Method," in IEEE Transactions on Vehicular Technology, vol. 72, no. 9, pp. 11727-11740, Sept. 2023, doi: 10.1109/TVT.2023.3266817.

[59] J. Chen, Z. Wang, Z. Li, J. Shen and P. Chen, "Multi-UAV Coverage Path Planning Based on Q-Learning," in IEEE Sensors Journal, doi: 10.1109/JSEN.2025.3580995.

[60] C. Papachristos, K. Alexis, L. R. G. Carrillo and A. Tzes, "Distributed infrastructure inspection path planning for aerial robotics subject to time constraints," 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 2016, pp. 406-412, doi: 10.1109/ICUAS.2016.7502523.

[61] C. Gao, X. Wang, X. Chen, and B. M. Chen, “A hierarchical multi-UAV cooperative framework for infrastructure inspection and reconstruction,” Control Theory and Technology, vol. 22, no. 3, pp. 394–405, Mar. 2024, doi: 10.1007/s11768-024-00202-0.

[62] D. N. Bui, T. N. Duong and M. D. Phung, "Ant Colony Optimization for Cooperative Inspection Path Planning Using Multiple Unmanned Aerial Vehicles," 2024 IEEE/SICE International Symposium on System Integration (SII), Ha Long, Vietnam, 2024, pp. 675-680, doi: 10.1109/SII58957.2024.10417512.

[63] Y. Sun and O. Ma, “Automating Aircraft Scanning for Inspection or 3D Model Creation with a UAV and Optimal Path Planning,” Drones, vol. 6, no. 4, p. 87, Mar. 2022, doi: 10.3390/drones6040087.

[64] Y. Wang, S. Wan, R. Qiu, and X. Li, “A Path Planning Algorithm for UAV 3D Surface

Inspection Based on Normal Vector Filtering and Integrated Viewpoint Evaluation," Journal of Field Robotics, Feb. 2025, doi: 10.1002/rob.22530.

[65] Z. Huang, "View planning for visual detection coverage tasks of large airplane upper surface using UAVs," Biomimetic Intelligence and Robotics, p. 100228, Feb. 2025, doi: 10.1016/j.birob.2025.100228.

[66] S. Zhang, C. Liu, and N. Haala, "Guided by model quality: UAV path planning for complete and precise 3D reconstruction of complex buildings," International Journal of Applied Earth Observation and Geoinformation, vol. 127, p. 103667, Jan. 2024, doi: 10.1016/j.jag.2024.103667.

[67] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. New York: Cambridge University Press, 2017.

[68] H. V. Abeywickrama, B. A. Jayawickrama, Y. He and E. Dutkiewicz, "Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance," in IEEE Access, vol. 6, pp. 58383-58394, 2018.

[69] H. V. Abeywickrama, B. A. Jayawickrama, Y. He and E. Dutkiewicz, "Empirical Power Consumption Model for UAVs," 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1-5, 2018.

[70] R. Alyassi, M. Khonji, A. Karapetyan, S. C. -K. Chau, K. Elbassioni and C. -M. Tseng, "Autonomous Recharging and Flight Mission Planning for Battery-Operated Autonomous Drones," in IEEE Transactions on Automation Science and Engineering, 2022.

[71] A. Fischer, F. Fischer, G. Jäger, J. Keilwagen, P. Molitor, and I. Grosse, "Exact algorithms and heuristics for the Quadratic Traveling Salesman Problem with an application in bioinformatics," Discrete Applied Mathematics, vol. 166, pp. 97–114, Nov. 2013, doi: 10.1016/j.dam.2013.09.011.

[72] J. H. Holland, ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS : an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge: The Mit Press, 1992.

[73] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators," Artificial

Intelligence Review, vol. 13, no. 2, pp. 129–170, 1999, doi: 10.1023/a:1006529012972.

[74] A. Aggarwal, D. Coppersmith, S. Khanna, R. Motwani, and B. Schieber, “The angular-metric traveling salesman problem,” pp. 221–229, Jan. 1997, doi: 10.5555/314161.314259.

[75] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[76] K. Deb, “Optimization, learning and natural algorithms”, Ph.D. Thesis, Politechnico di Milano, 1992.

[77] G. Reinelt, “TSPLIB—A Traveling Salesman Problem Library,” ORSA Journal on Computing, vol. 3, no. 4, pp. 376–384, Nov. 1991, doi: 10.1287/ijoc.3.4.376.

[78] H. Choset, “Coverage of Known Spaces: The Boustrophedon Cellular Decomposition,” Autonomous Robots, vol. 9, no. 3, pp. 247–253, 2000, doi: 10.1023/a:1008958800904.

[79] Z. Liu, S. Cheng, Y. Su, G. Duan and J. Tan, "Semantic Segmentation Based Spraying Trajectory Planning for Complex Product," in IEEE Transactions on Industrial Informatics, vol. 20, no. 7, pp. 9416-9426, July 2024, doi: 10.1109/TII.2024.3383865.

[80] K. Hormann, K. Polthier, and A. Sheffer, “Mesh parameterization,” ACM SIGGRAPH ASIA 2008 courses, pp. 1–87, Dec. 2008, doi: 10.1145/1508044.1508091.

[81] O. Sorkine, D. Cohen-Or, R. Goldenthal and D. Lischinski, "Bounded-distortion piecewise mesh parameterization," IEEE Visualization, 2002. VIS 2002., Boston, MA, USA, 2002, pp. 355-362, doi: 10.1109/VISUAL.2002.1183795.

[82] X. Gu, S. J. Gortler, and H. Hoppe, “Geometry images,” ACM Transactions on Graphics, vol. 21, no. 3, pp. 355–361, Jul. 2002, doi: 10.1145/566654.566589.