ECOLE
POLYTECHNIQUE
DE BRUXELLES

Automated Path Planning for Fixed-Wing UAVs Coverage Missions in Obstacle-Rich Environments

Thesis submitted for the award of the degree of

Master of Science in Computer Science and Engineering

Author: MAVROS Mickail

Supervisor:

GARONE Emanuele

MELEGA Marco

Academic year:

2023-2024

Abstract

This Master's Thesis was submitted by Mickaïl Mavros in the academic year 2023-2024 for the award of the degree of Master of Science in Computer Science and Engineering. The thesis consists of the "Automated Path Planning for Fixed-Wing UAVs Coverage Missions in Obstacle-Rich Environments".

This works aim to develop a comprehensive understanding of automated path planning techniques for fixed-wing Unmanned Aerial Vehicles (UAVs) engaged in coverage missions within complex, obstacle-laden landscapes. The primary goal is to contribute innovative solutions to enhance the efficiency and effectiveness of UAV path planning in challenging environments.

Building upon an algorithm initially developed during an internship at Sabca for multi-rotor and helicopter UAVs, this research extends the methodology to fixed-wing UAVs, aiming to create a versatile solution applicable to a broader range of Unmanned Aerial Systems (UAS). The thesis integrates several advanced techniques, including point placement, area subdivision into convex sub-cells, and efficient tour length computation for global Dubins path analysis. It employs a range of algorithms—heuristic methods, TSP relaxation, Boustrophedon path planning, and iterative techniques—to assess their performance in various environments characterised by different zone types, turning radii, and obstacle densities.

The findings reveal that the TSP relaxation method is optimal for environments with complex shapes and obstacles, providing a good balance between solution quality and computational efficiency. In contrast, the iterative method excels in scenarios with increased turning radii, showing the best adaptability to stricter constraints.

Overall, this thesis tries establishes a solid foundation for advanced fixed-wing UAV path planning, offering valuable insights and methodologies that enhance autonomous aerial operations and address real-world challenges.

Keywords : "Fixed-Wings UAV", "coverage path planning", "Dubins path", "Heuristics algorithms"

Github repository : `https://github.com/Mickmav/Master_Thesis_UAV`

Contents

List of Figures

List of Tables

1 Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have sparked profound revolutions across a multitude of industries, from precision agriculture and infrastructure monitoring to critical search and rescue operations. Their unparalleled adaptability and precision in traversing diverse terrains and accessing remote locations have made them indispensable tools for data acquisition, enabling insights previously unattainable through conventional means.

As the demand for efficient coverage of expansive areas characterised by varying terrain intensifies, the utilisation of UAVs becomes imperative. Fixed-wing UAVs, in particular, are increasingly favoured for their autonomy and agility, excelling in tasks such as extensive mapping and surveillance missions requiring prolonged endurance and rapid area coverage. However, their limited ability to rotate poses a challenge in navigating tight spaces and avoiding obstacles.

This master's thesis sets out to address a critical challenge in UAV operations: automated path planning for coverage missions in obstacle-rich environments. Focused on enhancing efficiency and precision, the research employs advanced trajectory planning techniques tailored specifically for fixed-wing UAVs. Recognising the need to overcome the limitations in rotation inherent to fixed-wing UAVs, the study draws inspiration from the Dubins Vehicle model.

The Dubins Vehicle model, originally developed for studying the shortest paths of vehicles with fixed velocities and constrained turning radii, provides a theoretical framework for optimising trajectories while accounting for the UAV's restricted manoeuvrability. By leveraging the Dubins Vehicle model to tackle the Travelling Salesman Problem (TSP), the study aims to develop an automated process for optimising trajectory planning.

In addition to the TSP-based approach, the research also evaluates two other path planning methods: the Boustrophedon method and an iterative approach. The Boustrophedon method, known for its effectiveness in sweeping large areas, is tested for its ability to systematically cover obstacle-rich environments. Meanwhile, the iterative method is explored as an adaptable approach to refine and optimise coverage paths depending on the UAV's constraints.

By integrating the TSP algorithm with the Dubins Vehicle framework, as well as evaluating alternative methods, this research endeavours to attain an efficient solution for navigating complex terrains while maximising coverage efficiency. Through comprehensive analysis and experimentation, the study seeks to enhance UAV mission capabilities, enabling seamless navigation and data collection in challenging environments.

The integration of TSP, Dubins Vehicle algorithms, and the additional path planning methods

provides a comprehensive approach to addressing the challenges of autonomous aerial operations. By exploring and comparing these techniques, this research aims to improve the efficiency and effectiveness of UAV missions. The results could contribute to better data acquisition and spatial coverage in various industries that depend on UAV technology.

The report begins by outlining the approach and underlying assumptions of the research, setting the stage for the analysis that follows. It then reviews the current state-of-the-art methods in UAV path planning, providing a foundation for understanding the advancements proposed in this study. Following this, the report delves into the specific methodology and algorithms used, offering a detailed examination of the techniques employed. Finally, the effectiveness of these algorithms is evaluated through testing in various environments, assessing their performance across different scenarios to determine their practical applicability.

Figure 1: Fixed wings UAV, image taken form Sabca's website : `https://www.sabca.be/markets/industry/business_unit/unmanned-autonomous-systems`

1.1 Approach

In this subsection, we outline the methodology employed in this project for path planning in the context of Unmanned Aerial Vehicle (UAV) operations, with a particular focus on resolving the Travelling Salesman Problem (TSP) using the Dubins Vehicle model.

1. **Geo-spatial Analysis**

   The first step involves analysing the input area, representing it as a polygon. If necessary, geodetic coordinates are transformed to ensure uniform spatial reference. Similarly, the geometries of any obstacles within the area are analysed and converted into a suitable format for subsequent processing.

2. **Algorithms** Different algorithms can be used to cover the generated area.

   - **Dubins TSP resolution:**

     For these method an efficient point placement is required to generate a TSP like instance inside of the designated area. Depending of the methodology a matrix capturing the distances between all the designated points can be computed.
     Depending on the method used to solve the problem a cost evaluation function can be used to take into account the constraint of the problem.

   - **Adaptive approach:**

     For this approach, the an automated and generic approach is used. Those method take into account the constraint of the UAV to build a solution. The solution can be build either by generating specific path that are efficient and adaptable to different areas, or simulating the movement the UAV in the area to cover it efficiently.

3. **Final path and Reporting**

   The selected optimisation algorithm is executed to find the best possible path within the given parameters. The resulting optimal path is then visualised and documented, and the solution is stored in the specified directory for further analysis and reference.

This methodology provides a systematic approach to path planning for UAVs within complex geo-spatial environments, ensuring efficient and obstacle-aware route determination. By integrating the Dubins Vehicle model with TSP resolution techniques, the research aims to optimise trajectory planning for fixed-wing UAVs, enabling them to navigate obstacle-rich environments with precision and efficiency. The Dubins TSP resolution serves as the primary focus of this thesis, with discussions centred around enhancements and applications within the context of UAV path planning.

1.2 Assumption and formalism

1.2.1 Dubins vehicle

The Dubins Travelling Salesman Problem (DTSP) is a variant of the classic Travelling Salesman Problem (TSP) tailored specifically for Dubins vehicles, which are characterised by fixed-wing UAVs or other vehicles constrained to move along paths with minimum turning radius.

In DTSP, the objective is to determine the shortest feasible tour that visits a set of target points while adhering to the motion constraints imposed by Dubins paths. These constraints typically include limitations on turning radius, minimum curvature, and maximum velocity, reflecting the vehicle's kinematic capabilities.

Unlike the standard TSP where Euclidean distances between points are considered, DTSP requires the calculation of Dubins paths, which are the shortest paths between two points for a vehicle with fixed turning radius and maximum velocity. These paths can take different forms, including straight segments, circular arcs, and transition curves, depending on the geometry of the problem and the constraints of the vehicle.

Solving DTSP involves finding a sequence of way points and corresponding Dubins paths that minimise the total distance travelled while satisfying the Dubins vehicle constraints. The challenge lies in efficiently exploring the solution space to identify the optimal or near-optimal tour that meets both the geometric and kinematic requirements of the problem.

Overall, DTSP presents a complex optimisation problem that requires specialised algorithms, such as genetic algorithms, heuristic methods, or exact algorithms tailored for Dubins vehicles, to effectively find solutions that balance tour length and feasibility within the constraints of the vehicle's motion dynamics.

In this report, we focus on exploring efficient heuristic and meta-heuristic approaches to solve the Dubins Travelling Salesman Problem (DTSP), acknowledging its NP-hard nature3. We do not delve into exact methods due to their computational complexity and impracticality for large-scale instances, as they often require exponential time to find optimal solutions. Instead, our emphasis lies on developing and evaluating heuristic algorithms capable of providing near-optimal solutions within reasonable time frames, thus addressing real-world applications where timely decision-making is crucial. By prioritising computational efficiency while maintaining solution quality, our approach aims to offer practical solutions for DTSP instances encountered in various domains, such as robotics, logistics, and aerial surveillance.

1.2.2 Different assumption

In this project, several assumptions are made to simplify the problem-solving process and streamline the development of algorithms for the Dubins Travelling Salesman Problem (DTSP). These assumptions include:

1. **Sufficient Space for Two-Way Paths:** It is assumed that the spaces between target points are large enough to accommodate two-way paths for the Dubins vehicle, allowing for seamless navigation between way-points without encountering spatial constraints.

2. **Simplified Polygon Shapes:** The shapes of both the area of interest and obstacles within it are simplified to basic polygonal forms. This reduction in complexity facilitates the implementation of path planning algorithms, as it allows for easier computation of intersections and distances between points and obstacles.

3. **Absence of Isolated Areas:** It is assumed that the area of interest is connected, with no isolated regions or inaccessible areas. This ensures that the Dubins vehicle can navigate through the entire space without encountering obstacles that completely block its path.

4. **Two-Dimensional Coordinates:** The problem is simplified to two-dimensional coordinates, neglecting any additional complexities that may arise in higher-dimensional spaces. This simplification allows for easier visualisation and computation of Dubins paths while still addressing the core challenges of the DTSP.

5. **Relatively Small Distance Between Points/Areas:** It is assumed that the distances between target points or areas of interest are relatively small compared to the capabilities of the Dubins vehicle. This assumption ensures that the problem remains tractable and that the Dubins vehicle can feasibly navigate between points without encountering excessively long or impractical paths, while allowing an optimal coverage of the area.

These assumptions help to define the scope of the problem and guide the development of algorithms tailored for DTSP solutions. While they may not capture all real-world complexities, they provide a practical framework for addressing Dubins vehicle path planning in various applications.

2 State of the art

This section provides an in-depth analysis of the state-of-the-art algorithms utilised for coverage missions using fixed wings UAVs. While the list presented here is not exhaustive, it encompasses a diverse range of algorithms that have been implemented and will thus be rigorously compared against the previously described methodologies outlined in this thesis.

2.1 Dubins path resolution

This subsection delves into the exploration of state-of-the-art algorithms specifically tailored for coverage missions employing fixed-wing UAVs.

2.1.1 Genetic algorithms resolution[1]

Genetic algorithms (GAs) have been widely applied to solve Dubins TSP (DTSP) problems due to their ability to efficiently explore large solution spaces and find near-optimal solutions through iterative optimisation processes inspired by natural selection and genetics.
In the context of DTSP, genetic algorithms typically involve encoding solutions as strings of genes representing sequences of way points or paths between targets. These strings are subjected to genetic operators such as mutation, crossover, and selection, mimicking biological evolution. Through repeated generations of solutions, the algorithm evolves towards solutions that exhibit desirable characteristics, such as minimal tour length while adhering to Dubins vehicle constraints.

2.1.2 Euclidean TSP based [2]

The main idea is to first start with a standard euclidean TSP resolution. When a sufficient solution is reached the different zone that doesn't respect Dubins vehicle formalism are actualised with the a possible rotation cycle to obtain a solution for the DTSP. That solution is then further optimised (keeping the original structure of the solution).
The proposed algorithm improves tour length with small turning radii by allowing various headings, preventing large loops. However, the gap narrows compared to the Alternating Algorithm as radius and target numbers increase due to limited manoeuvrability, resulting in wide turns and longer connections. To enhance such solutions, minimising turns is crucial, constrained by the vehicle's limited manoeuvrability.

2.1.3 Extended Tour algorithm[2]

Commencing with an empty solution space, the construction of the Dubins Tour unfolds methodically, as a sequence of Dubins paths are incrementally added to link the target points. Each step entails meticulous selection of the most optimal Dubins path, factoring in considerations such as turning radius and obstacle avoidance. This iterative process continues until all target points are seamlessly interconnected, marking the completion of the tour. To add points, either a greedy extension or k-extension method can be employed. The algorithms show significant tour length improvements over the Nearest Neighbour Heuristic, especially with more targets. The Greedy-2-extend is considered as sufficient with really low improvement when increasing the k-value but with a significantly longer running times making it impractical for larger test cases.

2.1.4 Insertion algorithms[2]

The starting tour (generated with greedy insertion) is usually some tour consisting of three targets. A new target is inserted into the tour at a position that causes the minimum increase in the length of the tour. The major difference between the insertion algorithms is the order in which the targets are inserted. The nearest and farthest insertion algorithms, for instance,insert the target whose minimal distance to a target of the tour is minimal or maximal.The cheapest insertion algorithm chooses a target whose insertion causes the lowest increase in the length of the tour.
It turns out that the performance is strongly depending on the turning radius of the vehicle. If the radius is small the efficiency of the algorithms is improved.

2.1.5 Heading Discretisation [3]

This method consist in using an algorithm for the Dubins Travelling Salesman Problem (DTSP) inspired by a method utilised for the curvature-constrained shortest path problem. Initially, a finite set of possible headings is chosen for each point, resulting in the construction of a graph with clusters corresponding to way-points. Each cluster contains nodes representing the choice of headings. Dubins distances between configurations of nodes in distinct clusters are then computed. Subsequently, the problem is framed as the generalised asymmetric travelling salesman problem, which is reduced to a standard asymmetric travelling salesman problem (ATSP) over designated nodes. This ATSP can be solved using established software such as Helsgaun's implementation of the Lin-Kernighan heuristic or other approximation algorithms.

2.1.6 Graph based approach[4]

A method is proposed to solve the Dubins Travelling Salesman Problem with Neighbourhoods (DTSPN) using a sampling-based road-map technique and a flexible Noon and Bean transformation for the Generalised Travelling Salesman Problem (GTSP). This approach yields tours comparable to previous methods but performs better in regions with frequent intersections. The algorithm is applied to guide a UAV in collecting data from a sparsely deployed sensor network. By converting the DTSPN into a GTSP through sampling and transforming it into an Asymmetric Travelling Salesman Problem (ATSP) with the Noon and Bean technique, the method effectively solves instances with several hundred nodes. However, for larger instances, using a direct GTSP solver like the memetic algorithm may be more appropriate.

2.1.7 Neighbourhood [5]

The proposed technique examines the optimal solution properties of the Dubins Traveling Salesman Problem with Neighbourhoods (DTSPN), presenting an external analysis. It enables the rapid approximation of solutions to the Dubins planning problem, termed the Dubins Touring Regions Problem (DTRP), for a specified sequence of region visits. The primary concept aligns with existing methodologies, determining the visit sequence prior to Dubins path determination. However, rather than explicitly discretising potential visit points within each goal area, an iterative local optimisation technique is employed to identify visit points and vehicle headings iteratively. This method swiftly generates initial solutions for problems with hundreds of targets within hundreds of milliseconds using standard computing resources, while maintaining solution quality comparable to more computationally intensive evolutionary approaches.

2.2 Other methods

This subsection delves into an exploration of various methods employed in coverage missions utilising fixed-wing UAVs.

2.2.1 Approach with Theta and Clothoids [6]

This paper introduces a path-planning technique tailored for fixed-wing unmanned aerial vehicles (UAVs), leveraging the Theta* algorithm. The method incorporates novel elements, such as the integration of Euler spirals, or clothoids, as connection arcs between nodes to address trajectory discontinuities. The design of clothoids is tailored to the aircraft's performance model, establishing a connection between curvature constraints and vehicle characteristics. Additionally, to alleviate computational complexity, the approach implements adaptive exploration distances and a vision cone, reducing the search space. This methodology ensures optimized flight paths for fixed-wing UAVs operating in static environments, offering enhanced trajectory smoothness.

2.2.2 Decomposition-based [7]

This article presents a novel method for planning fixed-wing aerial survey paths to efficiently cover large, complex agricultural fields. By decomposing intricate fields into multiple convex polygons, traditional back-and-forth boustrophedon paths ensure coverage. A top-down recursive greedy approach minimises survey flight time, computed rapidly enough for field application. Wind's impact on flight time is systematically incorporated into calculations, with potential for further enhancement to account for windy conditions. Real surveys confirm the method's effectiveness in providing complete image coverage under various conditions. Additionally, wind fields around survey areas are measured, affirming assumptions about wind's behaviour in UAV aerial surveys.

3 Methodology

The methodology for computing Dubins paths and strategically placing points for computation is fundamental to the effectiveness of our approach in coverage missions employing fixed-wing UAVs. Firstly, we strategically position points within the designated area, ensuring comprehensive coverage while considering the terrain and any obstacles present. These points serve as pivotal nodes for path planning and optimisation. Subsequently, we employ algorithms to compute the length of the Dubins paths between these strategically placed points, considering factors such as turning radius and obstacle avoidance. The chosen algorithms are meticulously selected to ensure efficiency and accuracy in trajectory planning, vital for successful UAV operations in dynamic environments.

3.1 Point placement

Achieving efficient coverage of a designated area, particularly when devoid of specific points of interest, necessitates strategic initialisation of points for the UAVs. Various methods are available for this purpose:

- **Generic Placement:** This involves systematic grid placement, ensuring uniform distribution of points across the area of interest. This method offers simplicity and ease of implementation, making it a viable option for initial point generation.
- **Poisson Disk Sampling:** Utilising this technique, uniformly distributed points are generated within a polygon, ensuring adequate spatial coverage while minimising point clustering. Poisson Disk Sampling provides flexibility in point distribution, catering to the intricacies of the terrain and enhancing coverage efficiency.

Each method offers distinct advantages and considerations, influencing their suitability based on the specific requirements and characteristics of the coverage mission.

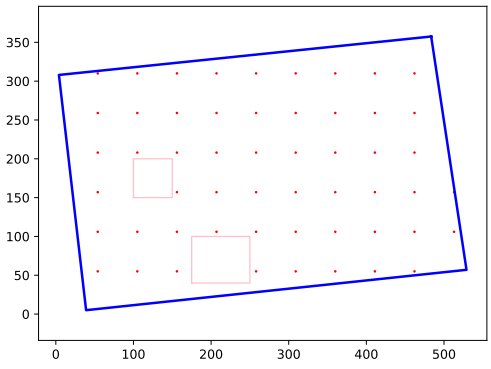

Figure 2: Generic point placement in a polygon

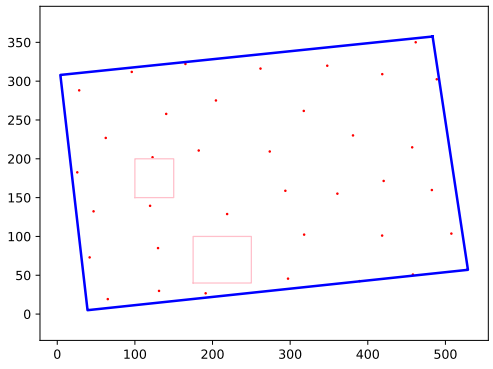

Figure 3: Random point placement in a polygon

Generic placement excels in simplicity and computational efficiency, making it particularly effective in straightforward cases with wide areas to cover. Its systematic grid-based approach ensures rapid point generation, facilitating quick deployment of UAVs for coverage missions.

On the other hand, while Poisson Disk Sampling offers enhanced spatial distribution and minimises point clustering, it entails higher computational overhead, especially in intricate spaces. This method scales poorly for wide areas due to its increased computational demands, making it more suitable for scenarios where precise point distribution is paramount, but computational efficiency may be sacrificed for accuracy. Another downside of Poisson Disk Sampling lies in its inherent randomness, which can result in different point distributions for the same area across multiple runs. While this randomness may be advantageous in certain scenarios, it introduces variability and unpredictability, potentially impacting the consistency of coverage outcomes.

Given the trade-offs between computational efficiency and point distribution accuracy, we have opted to utilise the generic placement method exclusively in our approach. This decision not only streamlines computation but also ensures consistency and repeatability in point placement across various coverage missions. By leveraging the systematic grid-based approach of generic placement, we aim to simplify the computational process while maintaining adequate coverage effectiveness for a wide range of scenarios.

3.2 Subdivision for Enhanced Computation Efficiency

To address the escalating computational demands and inefficiencies associated with coverage missions, particularly in expansive areas, subdivision emerges as a promising strategy. By subdividing the area of interest into smaller, more manageable sub-areas, computational complexities are alleviated, enabling more focused and efficient processing.

3.2.1 Benefits of Subdivision

1. **Reduced Computational Load:** Subdividing the area distributes the computational workload, preventing overwhelming processing demands often encountered in large-scale operations.

2. **Improved Scalability:** Handling smaller sub-areas enhances computational scalability, maintaining consistent resource requirements as the area size increases, thus mitigating scalability issues.

3. **Enhanced Resource Utilisation:** Subdivision allows for better utilisation of computational resources, with processing tasks distributed across multiple processors or computing nodes, accelerating computation and improving efficiency.

4. **Optimised Performance:** Processing each sub-area independently facilitates the application of optimisation techniques tailored to its characteristics, resulting in optimised performance and outcomes.

5. **Allow treatment of convex areas only:** One significant advantage of subdivision is that it facilitates the exclusive treatment of convex areas. By subdividing the original area into smaller convex sub-areas, computational processes are greatly simplified. This approach streamlines algorithms for path planning, optimisation, and coverage, as convex shapes eliminate the complexities associated with concave regions, such as handling obstacles and irregular boundaries. Consequently, the program can operate more efficiently and accurately, leveraging well-established methods tailored for convex areas, thereby enhancing overall computational performance and reliability.

3.2.2 Drawback of subdivision

The major drawback of subdividing areas is that the solution needs to be recomposed with careful coordination and integration of results from each sub-area. This re-composition process can introduce several challenges:

1. **Synchronisation Overhead:** Combining results from various sub-areas often requires synchronisation, which can introduce delays and increase overall computation time. The need to ensure that all sub-areas have been processed before re-composing the solution can create bottlenecks.

2. **Boundary Issues:** When sub-areas are processed independently, handling the boundaries between these sub-areas can be problematic. Discrepancies or inconsistencies at the boundaries can lead to inaccuracies in the final solution, requiring additional computation to resolve these issues.

3. **Complexity in Integration:** Re-composing the final solution from multiple sub-area results can be complex and require sophisticated algorithms to ensure seamless integration. This added complexity can negate some of the efficiency gains achieved through subdivision.

3.2.3 Considerations

When employing subdivision, it is crucial to address the potential drawbacks to fully leverage its benefits. Strategies such as adaptive subdivision techniques, robust boundary refinement algorithms, efficient communication protocols, load balancing mechanisms, and advanced integration techniques can mitigate the challenges associated with re-composing the final solution. By carefully planning and implementing these strategies, the subdivision approach can be optimised for enhanced computational efficiency in large-scale coverage missions, particularly benefiting boustrophedon path planning and simulation.

3.2.4 Boustrophedon cell decomposition

Boustrophedon cell decomposition is a method used in robotics and path planning for dividing a given space into manageable, non-overlapping cells or regions. This technique involves sweeping a line through the space, akin to how a farmer plows a field, where the direction alternates at each pass. This "zigzag" pattern helps in systematically covering the entire area, making it useful for applications such as robotic vacuum cleaning, agricultural automation, and coverage path planning. The cells created by this method can then be individually navigated, ensuring that the robot can efficiently cover the entire area without missing any spots. The decomposition is well-suited for coverage missions due to its structured and systematic approach. By ensuring that each cell is convex and obstacles are precisely handled, the method guarantees complete coverage of the area.

- **Splitting Areas**

The boustrophedon approach involves traversing the area in a back-and-forth manner, creating parallel strips. When an obstacle is encountered, the path bifurcates, splitting the area into smaller, manageable cells. Each obstacle triggers the creation of new cells, ensuring that the space around the obstacles is adequately decomposed and covered. This process continues until the entire area is subdivided into distinct cells that can be easily navigated and processed.

- **Creation of Convex Cells**

  A notable advantage of the boustrophedon cell decomposition method is its ability to generate only convex cells. Convex cells simplify the computational process as they eliminate the complexities associated with non-convex shapes, such as handling irregular boundaries and multiple local minima in optimisation tasks. The subdivision into convex cells enhances the efficiency and accuracy of path planning algorithms, making it an ideal choice for coverage missions.

- **Merging Areas**

  To merge different areas, the proposed method involves first computing the path within each convex hull. Then, a heuristic algorithm is used to merge these paths, minimising both the total distance and the distance travelled within obstacles or outside the designated area.

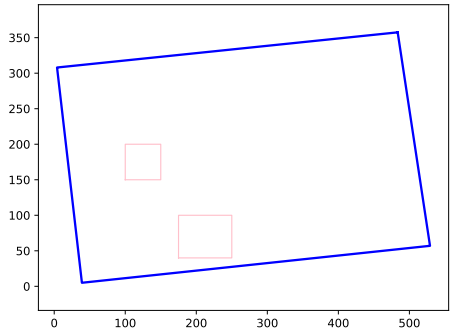

Figure 4: Polygon with the obstacles

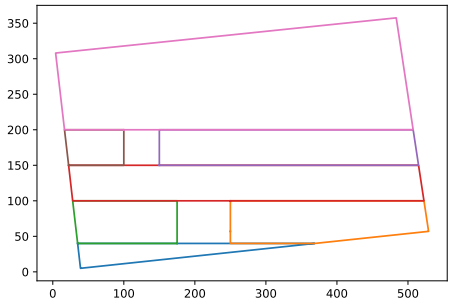

Figure 5: Decomposition of the polygons into different convex cells

Overall, the boustrophedon cell decomposition method exemplifies the practical application of subdivision in enhancing computational efficiency and performance in coverage missions. Its ability to handle obstacles dynamically and create manageable convex cells underscores its value in various operational contexts.

3.3 Tour Length Computation

An important step in computing the optimal path to cover an entire region with fixed-wing UAVs is to have an efficient method for calculating the tour length. This involves determining the total distance the UAV needs to travel to cover all designated points within the area. Efficient tour length computation is essential for optimising flight paths, minimising travel time, and ensuring comprehensive coverage. In the following section, we will explore various approaches and algorithms for accurately and efficiently computing tour lengths in coverage missions.

3.3.1 Dubins Path Library 8

For this project, the Dubins path library was used to calculate optimal trajectories for UAVs. Dubins paths are designed to find the shortest path between two points in a plane while adhering to constraints on turning radius and heading angle. This is particularly useful for vehicles that must move forward and have a fixed turning radius, such as fixed-wing UAVs.

The Dubins path library implements the Dubins' problem, which is formulated to find the shortest path connecting two points with specified orientations, considering a vehicle's constraint on turning radius. The Dubins path consists of segments with straight lines and circular arcs.

The main components of Dubins paths are:

1. **Straight-Line Segments:** These are segments where the vehicle moves in a straight line.
2. **Circular Arcs:** These segments represent the vehicle's turns, constrained by a fixed radius. The vehicle can make left or right turns.

The Dubins path library uses a combination of these segments to compute the shortest feasible path. The path is described by three possible segments: L, R, and S for left turn, right turn, and straight line, respectively. The library optimally sequences these segments to form a path that connects the start and end points while satisfying the turning constraints.

$$d_{\text{path}} = L + R + S$$

The path calculation considers the following constraints:

- *Minimum Turning Radius (ρ):* The smallest radius of curvature the vehicle can achieve.
- *Initial and Final Heading:* The orientation of the vehicle at the start and end points.

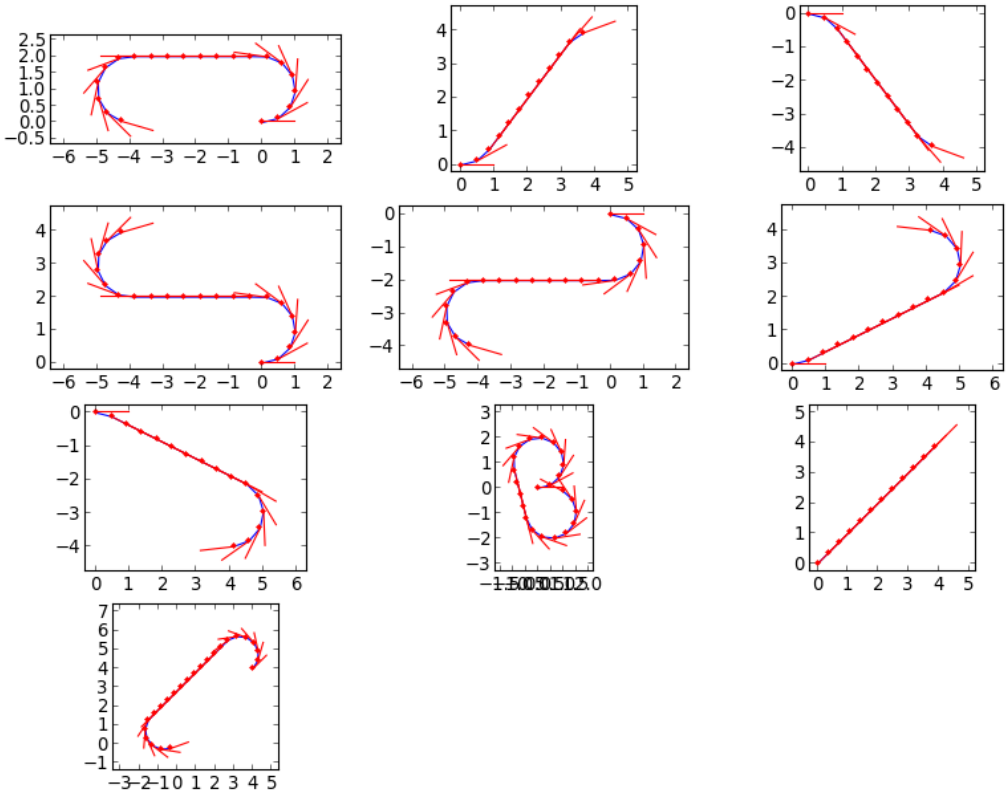

Figure 6: Dubins vehicle computation example from Andrew Walker 8

The equations governing the Dubins path calculation are based on the geometric properties of circles and lines. For instance, the length of a Dubins path segment is determined by the distance between points and the curvature of the trajectory.

The Dubins path problem is therefore solved by finding the minimum-length path among all feasible paths consisting of these segments. The distance between two points with orientations θ_1 and θ_2, and a turning radius ρ, can be computed using the following equations for the various path types.

3.3.2 Heading Computation

The computation of headings between points is crucial for understanding the direction changes required throughout the tour. Two methods were employed to compute these headings:

1. **Direct Heading Calculation:** This approach computes the heading by determining the angle between two consecutive points. The heading angle θ is given by:

$$\theta = \arctan 2(dy, dx)$$

where dx and dy are the differences in the x and y coordinates between the points.

2. **Bisector Vector Computation:** This method calculates headings by first determining vectors between consecutive points and then computing the angle of the bisector of these vectors. The heading angle is derived from the normalised bisector vector:

$$\mathbf{v}_1 \leftarrow \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}\mathbf{v}_2 \leftarrow \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}$$

$$\mathbf{v}_{\text{sum}} = \mathbf{v}_1 + \mathbf{v}_2$$

$$\theta = \arctan 2(v_{\text{sum},y}, v_{\text{sum},x})$$

where v_{bisector} is the vector obtained by summing and normalising the vectors from the previous and next points.

These methods ensure accurate heading computations, which are integral to optimising the UAV's path and ensuring efficient coverage.

3.3.3 Advantages of Bisector Vector Calculation

When computing headings for path planning, the second method, *Bisector Vector Calculation*, offers significant advantages over the direct heading calculation method. One notable benefit is its enhancement in the Dubins path computation, which is crucial for managing the turning radius constraint of UAVs.

The bisector vector method calculates headings by considering the vectors between consecutive points and computing the angle of their bisector. This approach provides several key advantages:

1. **Improved Handling of Turning Radius Constraints:** The bisector vector calculation method reduces the impact of the turning radius constraint by providing smoother transitions between consecutive segments. Instead of merely computing the heading based on the direct line segment between two points, it incorporates the geometric relationship between adjacent vectors. This results in a heading that better accommodates the vehicle's turning radius, leading to more feasible and efficient Dubins paths.

2. **Smoother Trajectories:** By calculating the bisector of vectors from neighbouring points, the method helps in generating smoother paths. This smoothness is essential for vehicles with turning constraints, as it minimises sharp direction changes and ensures that the computed path adheres to the vehicle's turning radius requirements. This smooth transition reduces the likelihood of sharp turns that could exceed the vehicle's turning capabilities.

3. **Enhanced Path Optimisation:** The bisector vector calculation allows for a more nuanced adjustment of headings, which can lead to better optimisation of the overall path. It enables more accurate alignment with the required turning radius, potentially reducing the total path length and enhancing efficiency. This optimisation is particularly beneficial for complex coverage patterns where direct heading calculations might result in sub-optimal paths.

4. **Reduced Path Deviations:** By considering the angles between vectors and their bisectors, this method mitigates abrupt deviations that could occur with direct heading calculations. This leads to paths that are more consistent with the intended trajectory, ensuring that the UAV can follow the computed path more closely and efficiently.

In summary, the bisector vector calculation method enhances the Dubins path computation by diminishing the adverse effects of the turning radius constraint. It provides a more refined and smoother heading adjustment, leading to more feasible and optimised paths for UAVs, particularly in complex coverage scenarios.

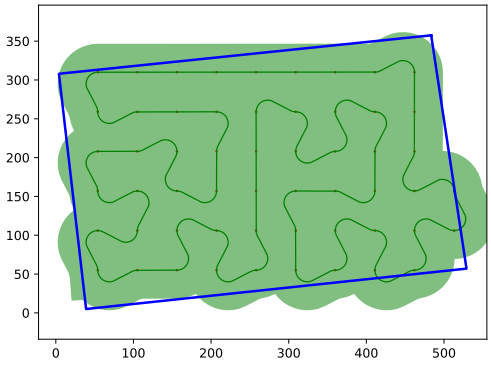

Figure 7: Dubins path generation with direct headings, path length = 3286.8

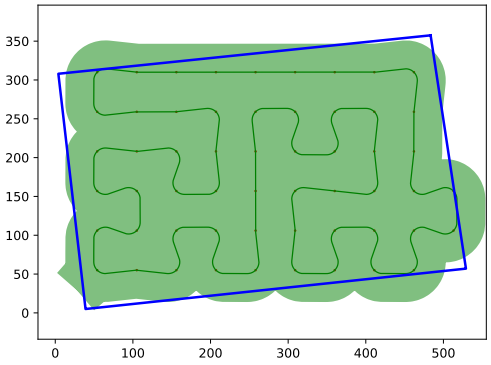

Figure 8: Dubins path with bisector vector headings computation, path length = 2962.5

4 Algorithms

In this section we will describe in depth the different algorithms that will be implemented to solve our different instance

4.1 Heuristic resolution

Heuristic methods are invaluable in optimisation, offering efficient solutions to complex problems where exact methods fall short. These techniques use domain-specific knowledge and approximations to explore large search spaces and find near-optimal solutions quickly. This section delves into three prominent heuristic approaches: Classical Genetic Algorithms, Simple Local Search, and Simulated Annealing. Each method has unique strengths, making them suitable for various optimisation challenges, including path planning for fixed-wing UAV missions, scheduling, and resource allocation. A key feature of these methods is their adaptability to different cost evaluations, such as the Dubins cost evaluation, which is crucial for fixed-wing UAV path planning. By incorporating Dubins cost evaluation, these heuristic methods can be tailored to meet specific optimisation requirements, enhancing performance and achieving desirable outcomes in diverse applications.

4.1.1 Classical Genetic Algorithms

In this subsection, we delve into the realm of Classical Genetic Algorithms (CGAs), renowned for their adeptness in navigating complex search spaces and uncovering optimal solutions through the principles of natural selection and genetic recombination. Serving as a cornerstone in evolutionary computation, CGAs provide a versatile framework adaptable to a myriad of optimisation problems, including path planning for fixed-wing UAV missions.

- **Algorithmic Framework:**

    1. Initialisation: The process commences with the initialisation of a candidate solution population, typically represented as chromosomes in a genetic encoding scheme. These initial solutions lay the groundwork for subsequent evolutionary processes.
    2. Selection: Through a process reminiscent of natural selection, individuals from the population are chosen for reproduction based on their fitness, evaluated by their performance in solving the optimisation problem. Mechanisms like tournament selection or roulette wheel selection ensure that fitter individuals have a higher likelihood of becoming parents for the next generation.

3. Crossover and Mutation: Selected individuals undergo genetic operations such as crossover and mutation to generate offspring with characteristics inherited from their parents. Crossover merges genetic information from two parent solutions, while mutation introduces random changes to diversify the population and prevent premature convergence.
4. Evaluation: The fitness of offspring is assessed based on their problem-solving ability. This evaluation guides the evolutionary process, favouring individuals with higher fitness for survival and reproduction.
5. Termination: The evolutionary process continues through multiple generations until meeting a termination criterion, such as reaching a maximum number of generations or achieving a satisfactory level of solution quality. At this juncture, the best-performing individual, or a set of elite individuals, is chosen as the final solution.

- **Practical Considerations:**

  Classical Genetic Algorithms offer a versatile and robust approach to path planning for fixed-wing UAV missions. Their adeptness in navigating complex search spaces and identifying near-optimal solutions makes them well-suited for addressing challenges like terrain variability, obstacle avoidance, and mission-specific constraints. Through an iterative process of evolution and selection, CGAs empower the derivation of efficient trajectory plans tailored to the unique requirements of fixed-wing UAV operation.

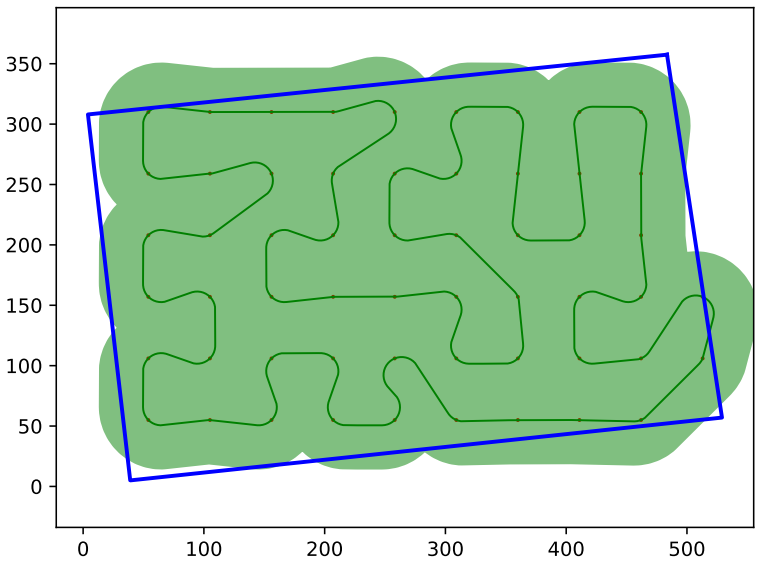

Figure 9: Genetic Algorithms resolution

4.1.2 Simple Local Search

Simple Local Search (SLS) is a heuristic method that iterative improves a single candidate solution by exploring its local neighbourhood. SLS is widely used due to its simplicity and effectiveness in finding satisfactory solutions within reasonable time-frames. In this context, the neighbourhood exploration is specifically defined by a 2-exchange mechanism, where two elements of the solution are swapped to generate a new candidate solution.

- **Algorithmic Framework:**

  1. **Initialisation:** Begin with an initial solution, selected randomly or based on a heuristic.
  2. **Neighbourhood Exploration:** Evaluate the neighbouring solutions of the current solution using a 2-exchange method, which involves swapping two elements within the solution to create a new one.
  3. **Selection:** Choose the best neighbouring solution according to a predefined criterion (e.g., minimum cost or maximum fitness).
  4. Iteration: Replace the current solution with the selected neighbouring solution.
  5. **Termination:** Repeat the process until a termination condition is met, such as a maximum number of iterations or a solution quality threshold.

- **Practical Considerations:**
  Simple Local Search is especially useful for problems where an approximate solution is sufficient, and rapid convergence is desired. It is applicable in various domains, including scheduling, routing, and resource allocation. The use of 2-exchange ensures a straightforward yet effective mechanism for generating new candidate solutions.

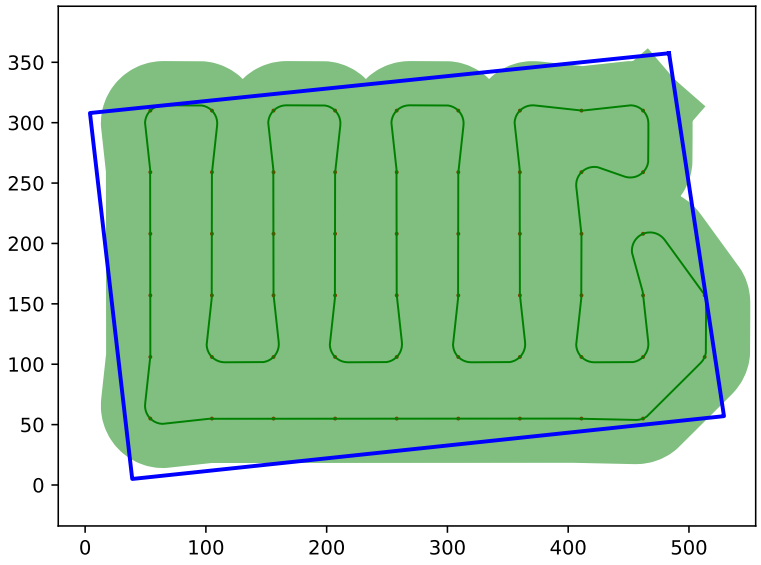

Figure 10: Simple Local Search resolution

4.1.3 Simulated Annealing

Simulated Annealing (SA) is a probabilistic heuristic method inspired by the annealing process in metallurgy. It is designed to escape local optima by allowing occasional uphill moves, thereby exploring a broader search space. Similar to Simple Local Search, the neighbourhood exploration in Simulated Annealing utilises a 2-exchange mechanism to generate new candidate solutions.

- **Algorithmic Framework:**

  1. **Initialisation:** Start with an initial solution and an initial temperature.
  2. **Neighbourhood Exploration:** Generate a neighbouring solution using a 2-exchange method, where two elements within the solution are swapped to create a new one.
  3. **Acceptance Criterion:** Decide whether to move to the neighbouring solution based on the difference in objective function values and the current temperature. Even worse solutions can be accepted with a certain probability.
  4. **Temperature Update:** Gradually decrease the temperature according to a cooling schedule.
  5. **Iteration:** Repeat the process until the system cools sufficiently, or a termination condition is met.

- **Practical Consideration:**

  Simulated Annealing is effective for large and complex optimisation problems, where avoiding

local optima is crucial. It is widely used in fields such as material design, scheduling, and VLSI design. Its flexibility and robustness make it suitable for applications requiring global optimisation. The 2-exchange mechanism facilitates a diverse exploration of the solution space, enhancing the method's ability to escape local optima.

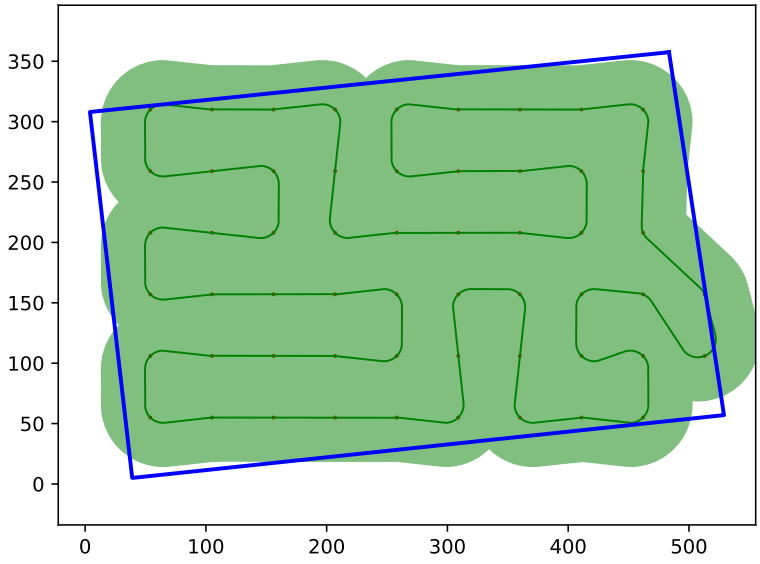

Figure 11: Simulated annealing resolution

By incorporating Classical Genetic Algorithms, Simple Local Search, and Simulated Annealing, Heuristic Resolution provides a comprehensive suite of methods to tackle diverse optimisation challenges.

4.2 TSP Relaxation Method: Modifying TSP Solutions for Fixed-Wing UAVs

The TSP Relaxation Method proposes a strategy to tailor TSP solutions for the unique requirements of fixed-wing UAV operations. By starting with a TSP solution generated using an efficient TSP solver, this method adapts the solution to align with the constraints and characteristics inherent to fixed-wing UAV flight paths. This approach can be viewed as a relaxation of the DTSP instance, aimed at obtaining a sub-optimal yet practical solution, akin to employing heuristics.

4.2.1 Algorithmic Process

The TSP Relaxation Method proposes a strategy to tailor TSP solutions for the unique requirements of fixed-wing UAV operations. Starting with a TSP solution generated using an efficient TSP solver,

this method adapts the solution to align with the constraints and characteristics inherent to fixed-wing UAV flight paths. This approach can be viewed as a relaxation of the DTSP instance, aimed at obtaining a sub-optimal yet practical solution, akin to employing heuristics.

4.2.2 Algorithmic Process

1. **TSP Initialisation:** The process begins with the generation of a TSP solution using a robust TSP solver. This generic solution serves as a starting point but does not yet consider fixed-wing UAV constraints.

2. **Adaptation to UAV Constraints:** The obtained TSP solution is used as an initial solution for a Simple Local Search, which incorporates the turning radius constraint and evaluates paths using the Dubins cost evaluation. This step ensures the trajectory is feasible for fixed-wing UAVs.

3. **Heuristic Refinement:** Through iterative refinements, the modified TSP solution is transformed into a trajectory plan optimised for fixed-wing UAV missions. The process balances computational efficiency with solution quality to produce a practical, sub-optimal solution.

4. **Evaluation and Iteration:** The adapted solution undergoes rigorous evaluation against predefined performance metrics, facilitating iterative improvements. This process continues until a satisfactory solution, balancing optimality and practical feasibility, is achieved.

4.2.3 Practical Implications

The TSP Relaxation Method offers a pragmatic approach tailored to the challenges encountered in fixed-wing UAV missions. By prioritising practicality over strict optimally, this method ensures that resulting trajectory plans remain viable and effective within real-world operational constraints. This approach sacrifices perfection in favour of practical feasibility, enabling the derivation of trajectory plans optimised for fixed-wing UAV operations. Its main advantage remain its improved efficiency by using existing TSP solvers to generate an initial solution quickly, which is then refined.

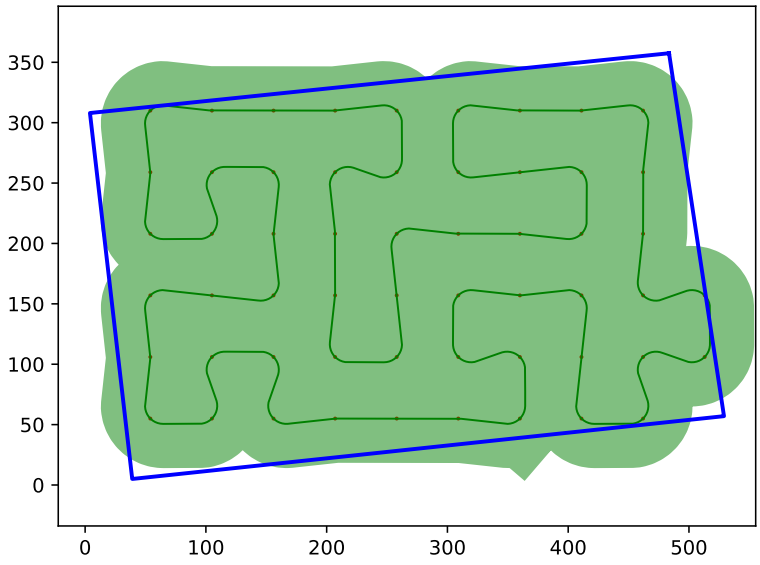

Figure 12: TSP Relaxation Method

4.3 Boustrophedon Path Planning for Fixed-Wing UAVs

The Boustrophedon method is an effective path planning strategy designed for navigating environments with complex geometries, such as those encountered in fixed-wing UAV operations. This approach is named after the ancient method of writing that alternates direction every line, which is mirrored in the way the UAV would traverse the space in a back-and-forth manner.

4.3.1 Methodology

The Boustrophedon path planning methodology involves the following key steps:

1. **Grid-based Decomposition**: The area to be covered is first divided into a grid of cells or strips. Each cell is typically of a width related to the UAV's vision radius to ensure complete coverage. This decomposition allows the path planning process to handle complex shapes and obstacles effectively.

2. **Sweeping and Path Generation**: The UAV then follows a sweeping motion across the cells. Starting from one edge of the grid, it moves in a straight line until it reaches the edge, then shifts to the next parallel line in the opposite direction, effectively covering the entire area. The direction alternates with each line, ensuring full coverage of the area.

3. **Handling Obstacles and Constraints**: Obstacles and constraints such as the UAV's turning

radius are incorporated into the path generation process. This involves adjusting the path to ensure it avoids obstacles and adheres to the UAV's operational constraints, such as minimum turning radius. The path is often adjusted to fit within these constraints while still achieving full coverage.

4. **Path Optimisation**: After generating the initial path, further optimisation can be applied to refine the path. This includes minimising the total distance travelled and ensuring the path adheres to specific constraints like turning radius. Techniques such as Dubins path evaluation can be used to optimise the trajectory to meet these constraints.

5. **Path Reunification**: One of the main challenges in the Boustrophedon method is to reunite the separate paths generated for each grid cell into a continuous trajectory. This involves combining the paths from each cell while ensuring smooth transitions and avoiding redundant traversals. Techniques to handle this include optimising connections between cell paths and applying the different constraints to obtain a suitable solution.

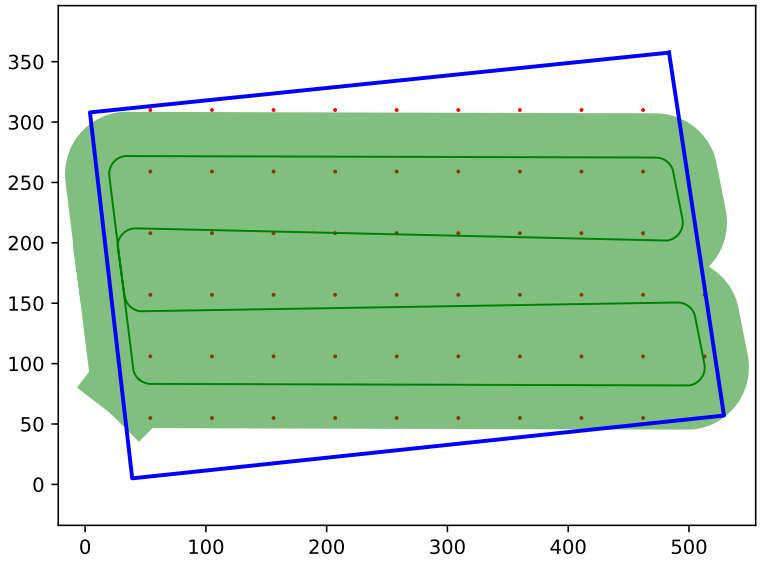

Figure 13: Boustrophedon resolution

4.3.2 Potential Advantages

- **Coverage Efficiency**: Boustrophedon path planning ensures complete coverage of an area through its systematic back-and-forth motion, making it ideal for applications like surveillance or mapping.

- **Obstacle Handling**: The grid-based approach allows effective handling of obstacles and complex terrain by adjusting paths within each cell and avoiding obstacles.
- **Adaptability**: This method is adaptable to different operational constraints. The grid size and sweeping patterns can be adjusted based on the UAV's vision radius and turning radius, making it flexible for various mission requirements.
- **Simple Implementation**: The Boustrophedon method is relatively straightforward to implement compared to other more complex path planning algorithms due to its grid-based approach and systematic coverage strategy.

4.3.3 Practical Considerations

When applying the Boustrophedon method to fixed-wing UAV operations, it is crucial to consider the specific characteristics of the UAV, such as its turning radius and the need for smooth path transitions. Incorporating these factors into the path planning process, such as through Dubins path evaluation, ensures that the UAV can navigate the environment efficiently while adhering to operational constraints.

In summary, the Boustrophedon method provides a robust and adaptable approach for path planning in complex environments. Its systematic coverage and flexibility make it a valuable tool for optimising fixed-wing UAV missions, though careful attention is needed for effectively combining and smoothing the generated paths.

4.4 Iterative Path Planning

The Iterative Path Planning method is designed to address challenges associated with high turning radii compared to the vision radius in fixed-wing UAV operations. This approach focuses on generating feasible paths even when the turning radius is significantly larger than the vision radius, which is common in practical UAV scenarios.

4.4.1 Methodology

The Adaptive Path Planning methodology involves the following steps:

1. **Point Generation**: The process begins by generating potential next points for the UAV. This is achieved by considering a range of angles and distances within a defined scaling factor relative to the turning radius. The UAV explores different heading directions and computes

possible next points while ensuring these points are valid by checking constraints such as polygon boundaries and the vision radius.

2. **Path Validation**: For each generated point, the method evaluates whether a Dubins path segment can be successfully created between the current and next points. This validation ensures that the generated paths adhere to the UAV's turning radius and fit within the operational constraints.

3. **Path Expansion and Iteration**: The valid paths are incorporated into the vehicle's path, and the process iterates to find the next feasible point. This iterative process continues until a satisfactory path is found or a stopping criterion is met, such as reaching a maximum number of iterations or covering a significant portion of the area.

4. **Final Path Optimisation**: Once a preliminary path is generated, further optimisation is performed to refine the trajectory. This involves minimising the total travel distance and ensuring smooth transitions between path segments. Techniques such as Dubins path optimisation are applied to adjust the final path and adhere to the constraints of the UAV.

5. **Path Reunion**: One of the primary challenges in this approach is to reunite the separate path segments into a continuous trajectory. This involves combining the paths from different iterations while ensuring that the transitions between segments are smooth and that the UAV does not retrace or overlap unnecessary paths.

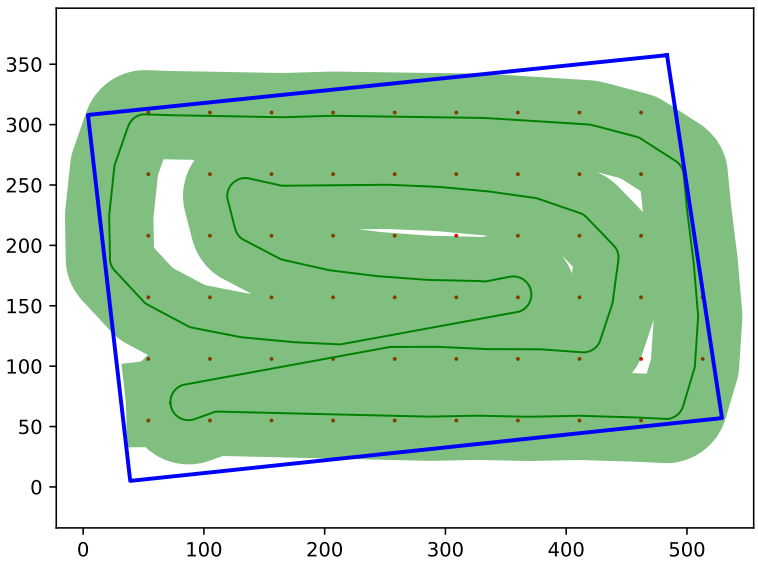

Figure 14: Iterative Path Planning

4.4.2 Potential Advantages

- **Feasibility with Large Turning Radius**: This method is particularly effective in scenarios where the UAV has a high turning radius relative to its vision radius. It enables the generation of feasible paths by accommodating the large turning constraints while still considering the vision capabilities of the UAV.
- **Adaptability**: The approach is adaptable to various operational constraints and environments. By iteratively generating and validating potential paths, it can handle complex terrains and obstacles efficiently.
- **Effective Path Optimisation**: The final optimisation phase ensures that the trajectory is not only feasible but also efficient. This is achieved through careful adjustment of path segments to meet the UAV's operational constraints and minimise travel distance.
- **Continuous Path Generation**: The iterative nature of the method allows for the continuous generation of path segments, which helps in maintaining a feasible and practical trajectory throughout the mission.

4.4.3 Practical Considerations

When applying the Adaptive Path Planning method to fixed-wing UAV operations, it is essential to consider the specific turning radius and vision radius of the UAV. Ensuring that the generated paths are practical and feasible within these constraints is crucial for successful mission execution. Additionally, careful attention must be given to the integration of path segments to maintain a smooth and continuous trajectory.

In summary, the Adaptive Path Planning method provides a robust approach for navigating environments with high turning radii, enabling the generation of practical and efficient paths for fixed-wing UAV missions. By addressing the challenges of large turning constraints and optimising the final trajectory, this method enhances the effectiveness of UAV operations in complex scenarios.

5 Experiments

In this section, we outline the different test scenarios used to evaluate the performance of the algorithms. We also describe the methodology employed to ensure a fair and comprehensive comparison.

5.1 Different experiment

To comprehensively evaluate the performance and adaptability of the proposed algorithms, different test were conducted in various environmental settings and under different conditions. These tests aimed to simulate real-world scenarios and assess how well the algorithms can handle different complexities and constraints.

5.1.1 Different Zones

To evaluate how well the algorithms adapt to environments of varying complexity, we tested them in three different zones:

- **Zone 1:** A simple rectangular area.
- **Zone 2:** An irregularly shaped area.

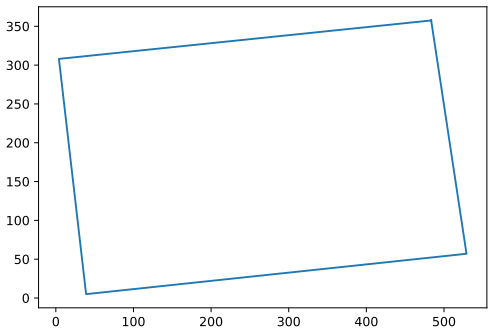

Figure 15: Rectangular area

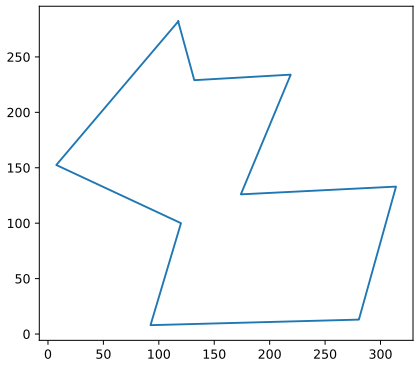

Figure 16: Irregularly shaped area

The purpose of testing in different zones is to assess the adaptability and robustness of each algorithm in environments ranging from simple to highly complex. This allows us to determine if an algorithm performs consistently across various types of environments or if its efficiency degrades with increased complexity.

5.1.2 Turning Radius

The algorithms were tested using three different turning radii to evaluate their performance under varying manoeuvrability constraints. These turning radii were assessed based on the UAV's vision radius, the primary parameter in coverage missions. The value of the turning radius introduces a new constraint, allowing the scaling of parameters to suit the different environments.:

- **Small Turning Radius:** Suitable for highly manoeuvrable systems.
  Turning radius : $1/2*$ vision radius.
- **Medium Turning Radius:** Represents average manoeuvrability.
  Turning radius : vision radius.
- **Large Turning Radius:** For systems with limited turning capabilities.
  Turning radius : $1.5*$ vision radius.

By varying the turning radius, we aim to understand the impact of manoeuvrability constraints on the performance of each algorithm. This is crucial for applications where the vehicle or robot has limited turning capabilities, and the algorithm needs to account for these constraints to ensure efficient path planning.

5.1.3 Obstacles

To evaluate how well the algorithms handle obstacles, we tested them under three different obstacle conditions:

- **No Obstacles:** Open area without any obstructions.
- **Sparse Obstacles:** Few obstacles present in the polygon.

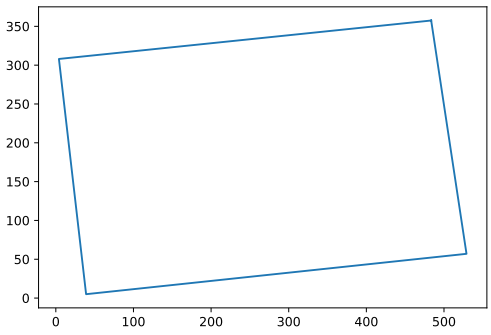

Figure 17: Rectangular area

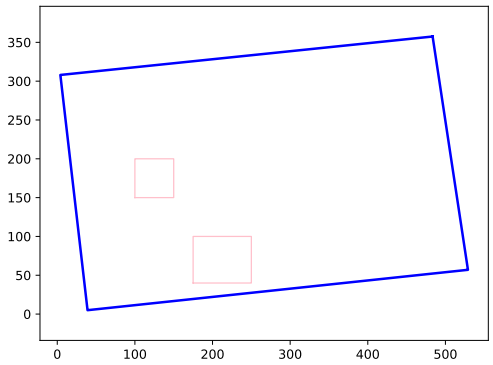

Figure 18: Rectangular area with obstacles

Testing with different densities of obstacles helps us determine how effectively each algorithm can navigate and find optimal paths in the presence of obstacles. This is particularly important for real-world applications where obstacles are common and need to be accounted for in the path planning process.

5.2 Methodology

Each algorithm was subjected to the above scenarios to evaluate its performance based on the following metrics:

- **Execution Time:** The time taken to complete the task.
- **Path Efficiency:** The optimally of the path taken.

The algorithms were run multiple times for each scenario to ensure reliability and accuracy of the results. The tests were designed to simulate realistic conditions and measure the algorithms' performance comprehensively.

6 Results

This section presents the results of the performance evaluation, detailing how each algorithm performed under the various test scenarios.
All Images represent a typical path in the specified environment.

6.1 Standard experiments

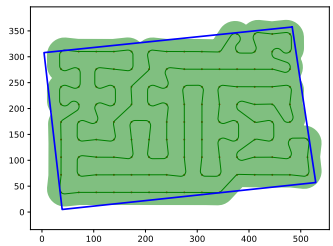

Figure 19: SA resolution in polygon 0

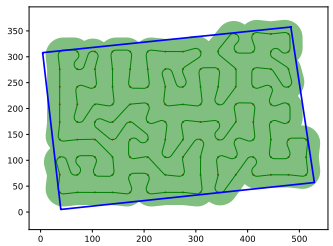

Figure 20: GA resolution in polygon 0

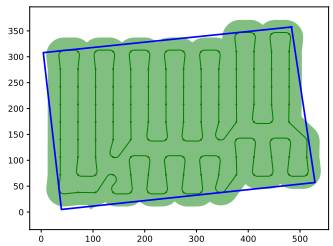

Figure 21: SLS resolution in polygon 0

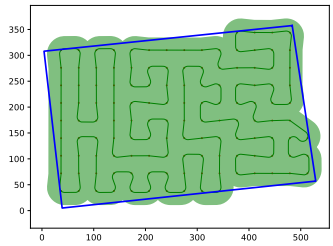

Figure 22: TSP relaxation resolution in polygon 0

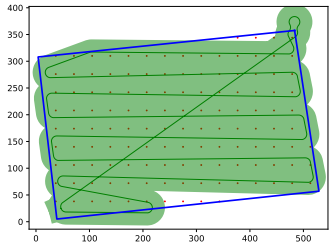

Figure 23: Boustrophedon resolution in polygon 0

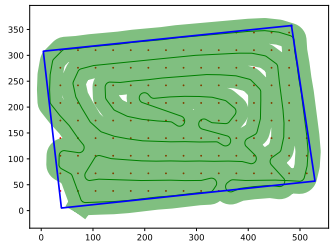

Figure 24: Iterative resolution in polygon 0

6.2 Performance in Different Zones

Algorithms	Polygon 0 17		Polygon 1 16	
	Path length	Computation time	Path length	Computation time
Dubins SA	4670.7	48.2	2655.1	52.3
Dubins GA	4942.9	862.5	2657.3	1275.7
Dubins SLS	4559.6	35.4	2637.9	47.9
TSP Relax	4541.4	6.6	2521.2	9.1
Boustrophedon	4574.5	0.1	2315.5	0.11
Iterative	4478.7	614.1	2384.9	171.8

Table 1: Performance of Different Algorithms in two different areas, no obstacles, turning radius = vision radius/2

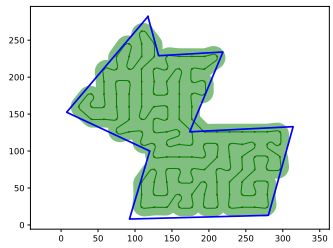

Figure 25: SA resolution in polygon 1

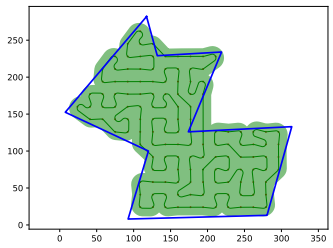

Figure 26: GA resolution in polygon 1

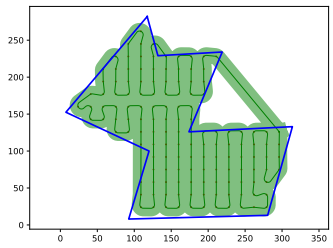

Figure 27: SLS resolution in polygon 1

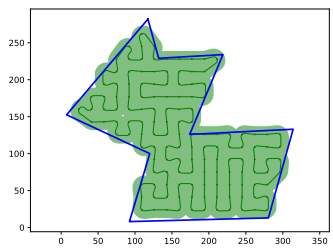

Figure 28: TSP relaxation resolution in polygon 1

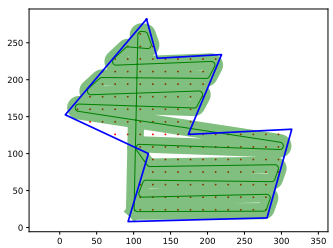

Figure 29: Boustrophedon resolution in polygon 1

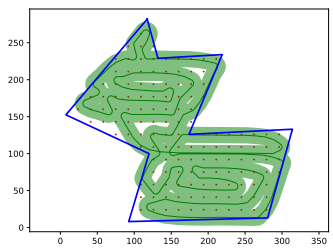

Figure 30: Iterative resolution in polygon 1

6.3 Performance with Varying Turning Radii

Algorithms	Turning Radius = VR/2		Turning Radius = VR	
	Path length	Computation time	Path length	Computation time
Dubins SA	4670.7	48.2	6248.8	51.7
Dubins GA	4942.9	862.5	6455.3	1289.1
Dubins SLS	4559.6	35.4	5559.6	55.3
TSP Relax	4541.4	6.6	5916.9	91.8
Boustrophedon	4574.5	0.1	4633.4	0.13
Iterative	4478.7	614.1	3842.1	283.7

Table 2: Performance of Different Algorithms in two different areas, no obstacles, turning radius = vision radius/2 and turning radius = vision radius

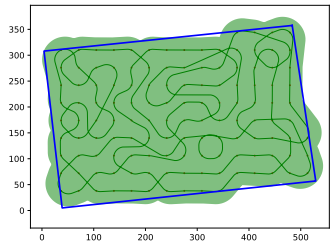

Figure 31: SA resolution in polygon 0, TR = VR

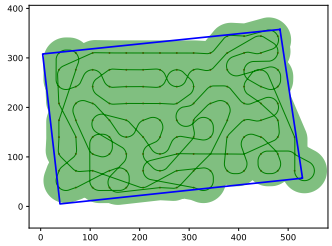

Figure 32: GA resolution in polygon 0, TR = VR

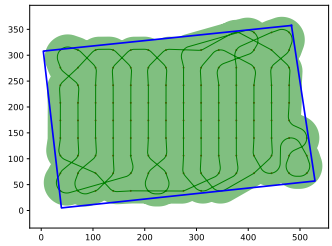

Figure 33: SLS resolution in polygon 0, TR = VR

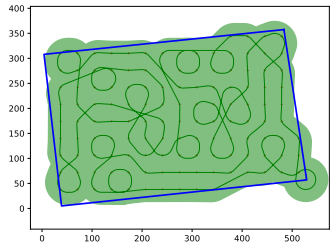

Figure 34: TSP relaxation resolution in polygon 0, TR = VR

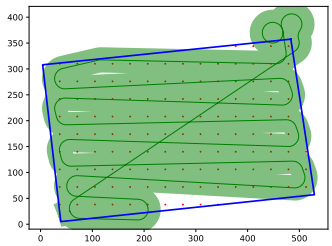

Figure 35: Boustrophedon resolution in polygon 0, TR = VR

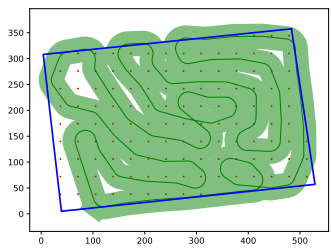

Figure 36: Iterative resolution in polygon 0, TR = VR

Algorithms	**Turning Radius = VR∗1.5**	
	Path length	**Computation time**
Dubins SA	8223.8	51.7
Dubins GA	8135.1	1734.4
Dubins SLS	6784.4	62.5
TSP Relax	7772.7	93.5
Boustrophedon	5574.2	0.23
Iterative	3465.2	124.1

Table 3: Performance of Different Algorithms in two different areas, no obstacles, turning radius = vision radius ∗ 1.5

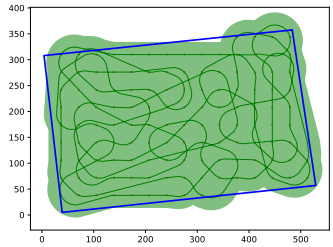

Figure 37: SA resolution in polygon 0, TR = 1.5 ∗ VR

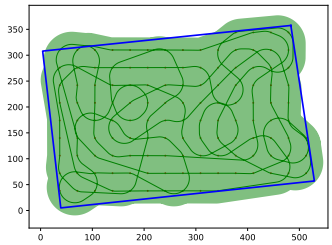

Figure 38: GA resolution in polygon 0, TR = 1.5 ∗ VR

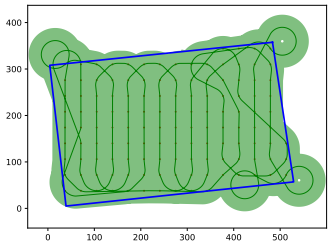

Figure 39: SLS resolution in polygon 0, TR = 1.5 ∗ VR

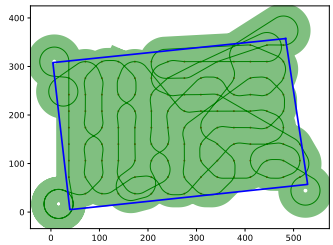

Figure 40: TSP relaxation resolution in polygon 0,TR = 1.5 ∗ VR

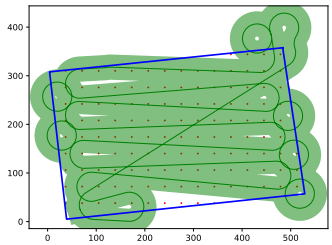

Figure 41: Boustrophedon resolution in polygon 0, TR = 1.5 ∗ VR

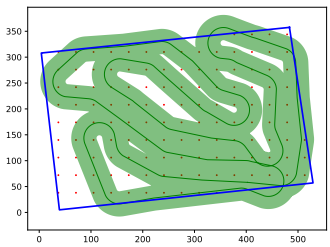

Figure 42: Iterative resolution in polygon 0, TR = 1.5 ∗ VR

6.4 Performance with and without Obstacles

Algorithms	No obstacles		Sparse obstacles	
	Path length	Computation time	Path length	Computation time
Dubins SA	4670.7	48.2	4535.8	45.5
Dubins GA	4942.9	862.5	4593.3	903.6
Dubins SLS	4559.6	35.4	4399.8	33.8
TSP Relax	4541.4	6.6	4403.8	6.5
Boustrophedon	4574.5	0.1	4759.5	0.13
Iterative	4478.7	614.1	4471.3	140.1

Table 4: Performance of Different Algorithms in polygon 0, turning radius = vision radius/2

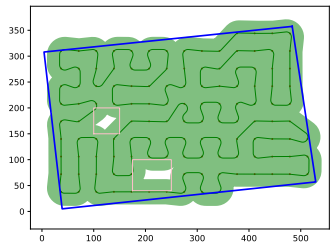

Figure 43: SA resolution in polygon 0, obstacles

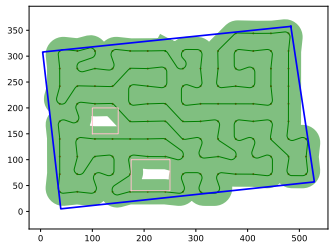

Figure 44: GA resolution in polygon 0, obstacles

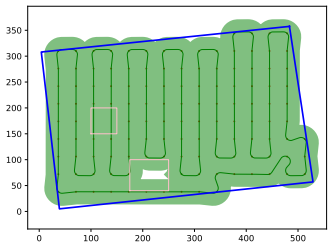

Figure 45: SLS resolution in polygon 0, obstacles

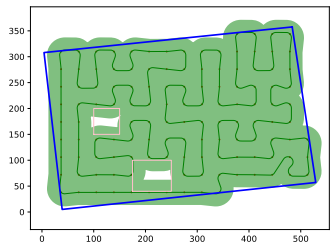

Figure 46: TSP relaxation resolution in polygon 0, obstacles

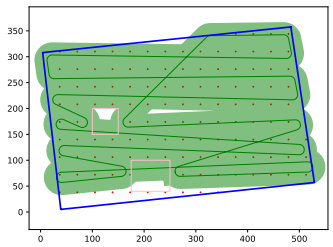

Figure 47: Boustrophedon resolution in polygon 0, obstacles

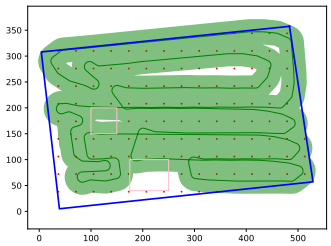

Figure 48: Iterative resolution in polygon 0, obstacles

7 Discussion

In this section, we will analyse and discuss the results obtained from the experiments to evaluate the strengths and weaknesses of each implemented algorithm.

7.1 Performance in different areas

Based on the experimental results, the genetic algorithm consistently underperforms compared to the other two heuristic alternatives. It generates longer path lengths, sometimes up to 7% longer, and requires significantly more computation time—up to 20 times more in some cases. While these heuristic methods perform relatively well in convex areas, they tend to deviate outside the polygon in irregularly shaped areas. This issue arises because the implementation does not check at each step whether a path between two points is entirely within the area to be covered. Incorporating such checks would further increase the computational cost, scaling with the number of edges and potential obstacles. Additionally, the curvature of Dubins paths is not easily predictable due to their dependency on headings, which prevents these paths from being anticipated beforehand. In contrast, the TSP relaxation method allows for better-quality solutions with minimal computational time, providing a similar type of resolution. This method effectively balances solution quality and computational efficiency.
Regarding the boustrophedon and iterative solutions, they generally yield better-quality solutions but do not explore the area as efficiently as other methods. The boustrophedon method, in particular, is significantly faster in constructing the solution. However, a notable downside is that when rearranging the paths together, it can sometimes extend outside the designated area. Additionally, both methods demonstrate limitations in complex, irregularly shaped areas where path planning requires more intricate adjustments.

In summary, the genetic algorithm lags behind in performance and efficiency, while the TSP relaxation method stands out for its balance between solution quality and computational time. The boustrophedon and iterative methods offer faster construction times and better solution quality but are less efficient in exploring complex areas and may sometimes extend beyond the target area during path rearrangement.

7.2 Performance with varying turning radius

As the turning radius increases, all heuristic algorithms experience a significant decline in performance, except for the iterative method, which shows a remarkable ability to adapt to the changing constraints. Among the other algorithms, the simple local search method manages to reach decent solutions due to its initial up-down straight line pattern, which is inherently more suited to the problem's resolution. However, even this method struggles as the turning radius increases, leading to less optimal paths and higher computational costs.

The TSP relaxation method, while effective under smaller turning radii, loses much of its utility with larger turning radii. The increased constraint fundamentally alters the nature of the solution, forcing way-points into positions that require more curved paths. This results in sub-optimal solutions and diminished efficiency, making the TSP method less appealing under these conditions.

The boustrophedon method initially performs well, especially when the turning radius matches the UAV's vision radius. It efficiently covers areas with only minor issues at the path extremities. However, as the turning radius increases, the effectiveness of this method declines. The necessity for curved Dubins paths often causes the UAV to extend outside the designated area, reducing the overall efficiency and coverage quality.

In contrast, the iterative method stands out for its adaptability to increasing turning radii. It effectively balances the stricter turning constraints with the need for comprehensive area coverage. As the turning radius increases, the iterative method adjusts by shortening the path length and increasing execution speed. Although there is a slight reduction in coverage efficiency, the method maintains a better balance between constraint adherence and area coverage compared to the other algorithms.

In summary, while most heuristic methods struggle with increased turning radii, the iterative method demonstrates a unique ability to adapt to these constraints. The simple local search, TSP relaxation, and boustrophedon methods all face significant challenges and reduced effectiveness as the turning radius grows. Only the iterative method manages to maintain a reasonable balance between path optimisation and coverage efficiency, making it the most robust option under varying turning radius conditions.

7.3 Performance with obstacles

The introduction of obstacles has a limited impact on path length and computation time for most methods, with some notable exceptions. Among these, the boustrophedon method faces significant challenges. This method struggles with efficiently rearranging paths to navigate around obstacles. Its inherent rigidity and the inflexibility of its sub-solutions hinder its ability to adapt dynamically, resulting in difficulties when attempting to circumvent obstacles effectively.
Most heuristic methodologies manage to perform adequately even with the presence of obstacles. However, the simple local search method demonstrates clear limitations in this context. It proves ineffective at avoiding smaller obstacles, as its straightforward, linear path planning does not account for or adjust to such impediments adequately.
The iterative method, while still performing reasonably well, shows some shortcomings compared to other approaches. It manages to navigate around obstacles but falls short in terms of area coverage. The iterative method's primary strength lies in its adaptability, but this comes at the cost of slightly reduced coverage efficiency when compared to other methods that handle obstacles more seamlessly.

In summary, while the addition of obstacles generally has a modest effect on path length and computation time for most algorithms, the boustrophedon method is notably disadvantaged due to its lack of adaptability. The simple local search method struggles with small obstacles, and the iterative approach, despite its adaptability, exhibits reduced efficiency in area coverage compared to other methods. Each method's effectiveness in dealing with obstacles varies, highlighting the importance of considering both obstacle navigation and overall coverage in path planning

7.4 Global analysis

In this part of the report the performance of the different path planning methods are discussed, evaluating their strengths and weaknesses across the different scenarios.

Heuristic methods vary widely in effectiveness depending on problem constraints. For example, the genetic algorithm often under-performs relative to other heuristic approaches, resulting in longer paths and requiring more computation time. While heuristic methods handle convex areas relatively well, they struggle with irregularly shaped regions and larger turning radii. Their ability to navigate around obstacles also varies, with some methods being more adaptable than others. Generally, heuristic methods are less efficient than the TSP relaxation approach.

The TSP relaxation method is effective in delivering good-quality solutions with minimal computation time, particularly in scenarios involving convex areas or fewer obstacles. It strikes a balance between solution quality and efficiency. However, its performance diminishes with increasing turning radii, as it struggles to adapt to more stringent path curvature constraints. This results in less optimal solutions and reduced overall effectiveness under such conditions.

The boustrophedon method excels in efficiently constructing solutions and performs well when the turning radius is equal to the UAV's vision radius. It is particularly fast in creating solutions and covers areas effectively. Nevertheless, it struggles with navigating around obstacles due to its rigidity and inflexibility. Additionally, as the turning radius increases, the method can extend paths outside the designated area, reducing its effectiveness in complex or irregularly shaped environments.

The iterative method demonstrates exceptional adaptability to varying constraints, such as increasing turning radii and the presence of obstacles. It effectively manages the trade-off between path curvature constraints and area coverage. As the turning radius increases, the iterative method adjusts by shortening path lengths and improving execution speed. Although it may show slightly reduced efficiency in coverage compared to other methods, its ability to adapt to changing conditions makes it the most robust option for managing increased turning radii.

In summary, the TSP relaxation method is optimal for environments with complex shapes and obstacles, providing a good balance between quality and computational efficiency. In contrast, the iterative method excels in scenarios with increased turning radii, showing the best adaptability to stricter constraints. While heuristic methods generally offer lower efficiency, and the boustrophedon method provides faster solution construction or better performance in convex areas, both face significant challenges with complex shapes, large turning radii, and obstacle navigation.

8 Conclusion

In conclusion, this report outlines a comprehensive methodology for improving path planning for Fixed-Wing UAVs in obstacle-rich environments. Our approach integrates several key components: point placement, subdivision of areas into convex sub-cells for improved computational efficiency, and advanced tour length computation to enable global Dubins path analysis. We employed a diverse set of algorithms, including heuristic techniques, TSP relaxation, Boustrophedon path planning, and iterative methods, to evaluate their effectiveness across different test environments. These environments varied in terms of zone types, turning radii, and obstacle density, allowing for a robust assessment of each method's performance and adaptability.

Our findings highlight the TSP relaxation method and the iterative method as the most effective approaches. The TSP relaxation method is particularly effective in convex areas with minimal obstacles, providing a balanced trade-off between solution quality and computational efficiency. Conversely, the iterative method demonstrates exceptional adaptability to complex, obstacle-rich environments and varying turning radii, making it the most robust choice for handling stringent operational constraints.

Despite these advancements, there are several promising avenues for future research. Extending these methods to three-dimensional space could improve mission planning by incorporating altitude variations. Additionally, the development of real-time adaptive algorithms capable of responding to dynamic conditions, such as adverse weather or sudden obstacles, could significantly enhance operational flexibility. Exploring multi-UAV coordination strategies could also address challenges related to large-area coverage and the limitations of individual UAVs in terms of energy and manoeuvrability.

Overall, this project establishes a strong foundation for advancing UAV path planning. It provides valuable insights and methodologies that pave the way for addressing more complex and dynamic real-world challenges, further pushing the boundaries of autonomous aerial operations.

Bibliography

1. Xin Yu and J. Y. Hung, "A genetic algorithm for the Dubins Traveling Salesman Problem," 2012 IEEE International Symposium on Industrial Electronics, Hangzhou, China, 2012, pp. 1256-1261, doi: 10.1109/ISIE.2012.6237270.

2. Babel, L. New heuristic algorithms for the Dubins traveling salesman problem. J Heuristics 26, 503–530 (2020). https://doi.org/10.1007/s10732-020-09440-2

3. J. Ny, E. Feron and E. Frazzoli, "On the Dubins Traveling Salesman Problem," in IEEE Transactions on Automatic Control, vol. 57, no. 1, pp. 265-270, Jan. 2012, doi: 10.1109/TAC.2011.2166311

4. Isaacs, Jason T., and João P. Hespanha. 2013. "Dubins Traveling Salesman Problem with Neighborhoods: A Graph-Based Approach" Algorithms 6, no. 1: 84-99. https://doi.org/10.3390/a601008

5. P. Váňa and J. Faigl, "On the Dubins Traveling Salesman Problem with Neighborhoods," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 4029-4034, doi: 10.1109/IROS.2015.7353945.

6. Bassolillo, Salvatore Rosario, Gennaro Raspaolo, Luciano Blasi, Egidio D'Amato, and Immacolata Notaro. 2024. "Path Planning for Fixed-Wing Unmanned Aerial Vehicles: An Integrated Approach with Theta* and Clothoids" Drones 8, no. 2: 62. https://doi.org/10.3390/drones8020062

7. Coombes, Matthew; Fletcher, Tom; Chen, Wen-Hua; Liu, Cunjia (2019). Decomposition-based mission planning for fixed-wing UAVs surveying in wind. Loughborough University. Journal contribution. https://hdl.handle.net/2134/10318751.v1

8. Walker, Andrew. "Dubins-Curves: An Open Implementation of Shortest Paths for the Forward Only Car." GitHub, 2008–. https://github.com/AndrewWalker/Dubins-Curves.

9. Stützle T., Heuristic optimisation (INFO-H413), Université Libre de Bruxelles (ULB), Bruxelles, Belgium.

10. De Smet Y., Recherche opérationnelle (INFO-H3000), Université Libre de Bruxelles (ULB), Bruxelles, Belgium.

11. Bersini H., Techniques of artificial intelligence (INFO-H410)), Université Libre de Bruxelles (ULB), Bruxelles, Belgium.

12. F M Puspita et al 2020 J. Phys.: Conf. Ser. 1480 012029

13. L. H. Nam, L. Huang, X. J. Li and J. F. Xu, "An approach for coverage path planning for UAVs," 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC), Auckland, New Zealand, 2016, pp. 411-416, doi: 10.1109/AMC.2016.7496385.

14. Bouzid, Yasser, Yasmina Bestaoui, and Houria Siguerdidjane. “Quadrotor-UAV Optimal Coverage Path Planning in Cluttered Environment with a Limited Onboard Energy.” IEEE, 2017. 979–984. Web.

15. Cook W., "Concorde TSP Solver". Concorde Home, March 2015, https://www.math.uwaterloo.ca/tsp/concorde/index.html.

16. Vankerschaver J., "pyconcorde", Github, 2nd Jully 2023, https://github.com/jvkersch/pyconcorde

17. dmishin, "tsp-solver", Github, 22th July 2020, https://github.com/dmishin/tsp-solver

18. Mulvad S., "fast-tsp", Github, 15th August 2023, https://github.com/shmulvad/fast-tsp

19. luanleonardo and Goulart F, "python-tsp", Github, 9th september 2023, https://github.com/fillipe-gsm/python-tsp