BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Marek Jakub**	Personal ID number:	**406714**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Area Coverage for Aerial Vehicles in the Presence of Obstacles

Bachelor’s thesis title in Czech:

Úloha pokrývání oblastí vzdušnými robotickými prostředky

Guidelines:

1. Familiarize yourself with existing algorithms for area coverage with aerial vehicles [1, 2, 3].
2. Propose an extension of existing algorithms to be applied in three-dimensional space and Dubins Airplane model [4].
3. Generalize the cellular decomposition approach [5] to consider obstacles around the covered area.
4. Evaluate the proposed approach and compare it with existing selected algorithm(s).
5. Verify the proposed approach for area coverage in a simulation and eventually using real vehicles.

Bibliography / sources:

[1] Galceran, Enric, and Marc Carreras. "A survey on coverage path planning for robotics." Robotics and Autonomous Systems 61.12 (2013): 1258-1276.
[2] Perez-Imaz, Hector IA, et al. "Multi-robot 3D coverage path planning for First Responders teams." IEEE International Conference on Automation Science and Engineering (CASE), 2016, pp. 1374-1379.
[3] Maza, Ivan, and Anibal Ollero. "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms." Distributed Autonomous Robotic Systems 6. Springer, Tokyo, 2007. pp. 221-230.
[4] Hamidreza Chitsaz and Steven M. LaValle, "Time-optimal paths for a dubins airplane," in 46th IEEE Conference on Decision and Control. IEEE, 2007, pp. 2379-2384.
[5] Xu, Anqi, Chatavut Viriyasuthee, and Ioannis Rekleitis. "Efficient complete coverage of a known arbitrary environment with applications to aerial operations." Autonomous Robots 36.4 (2014): 365-381.

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Jan Faigl, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **12.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

doc. Ing. Jan Faigl, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student’s signature

CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's thesis

Area Coverage for Aerial Vehicles in the Presence of Obstacles

Jakub Marek

June 2018
Supervisor: doc. Ing. Jan Faigl, Ph.D.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague on 11 May 2015

. . . .
Jakub Marek

Acknowledgement

I would like to thank to doc. Ing. Jan Faigl, Ph.D. for his help and supervision of my bachelor thesis, his advices and his patience. Also, I would like to thank to Ing. Petr Váňa for consultations and related work.

Abstract

This thesis is focused on area coverage problem of given convex shaped area surrounded by the obstacles by fixed-wing Unmanned Aerial Vehicles (UAV). Goal of area coverage problem is to find shortest covering trajectory feasible for given vehicle which covers the area completely. The area is considered to be covered if all its points are visited by given vehicle. Point of the area can be covered by vehicle's coverage device. Existing methods are solving area coverage problem by decomposing area to be covered into obstacle-free cells and covering each cell individually. However, obstacles stays arround the area, and therefore, we propose a method which considers it and determines solution with a respect to the obstacles and motion constraints of UAV. The proposed method uses Dubins Airplane Model to satisfy motion constraints of the UAV and to find best corresponding solution while all possible collisions with the obstacles surrounding the area are avoided.

Key words: Area coverage problem; fixed-wing UAV coverage

Abstrakt

Tato práce se zabývá úlohou pokrývání dané konvexní plochy obklopené překážkami Bezpilotním Létajícím Prostředkem (UAV) s pevným křídlem. Cílem pokrývací úlohy je nalézt nejkratší pokrývací trajektorii, která je vhodná pro prostředek, jímž je plocha pokrývaná. Plocha je považována za pokrytou, pokud byli všechny její body navštíveny pokrývacím zařízením. Existující metody řeší pokrývací úlohu rozdělením zadané plochy na menší buňky, jež překážky neobsahují. Nicméně, překážky zůstávají okolo pokrývané plochy a tudíž navrhujeme metodu, která se těmto překážkám vyhýbá a zároveň respektuje omezení spojená s používáním UAV. Navrhovaná metoda používá Dubinsův Model Letadla k určení nejlepšího řešení, které respektuje omezení daná UAV a které se vyhne všem překážkám obklopujících pokrývanou plochu.

Klíčová slova: Úloha pokrývání; Pokrývání s UAV s pevným křídlem

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In modern society, area coverage problems are connected with mowing of the gardens [10], cleaning of households and warehouses by autonomous vacuum cleaners [11], demining of formal war zones [12], painting of car components in autonomous factories [13], mapping of given area and searching for survivors after disaster [6], agricultural activities and many more. With the rapid development of localization systems, sensing devices, artificial intelligence, computational systems, and robotics in recent years, more capable *Unmanned Aerial Vehicles* (UAVs) are produced. Present systems are smaller, require less energy and weightless. Thanks to this, more tasks can be now performed by autonomous robots or unmanned aerial vehicles.

In this thesis, we consider a coverage planning problem to determine cost efficient trajectory for a UAV with the minimum turning radius that is requested to fly at the specified height above the terrain to cover the requested field (e.g., for crop spraying) while the field is surrounded by obstacles. Therefore, the problem being addressed consists of not only to determine the optimal coverage strategy but it also includes careful planning of the turning maneuvers to avoid obstacles while keeping the coverage maximal together with the safety of the vehicle.

The goal of area coverage is to cover the given area utilizing the sensing range of the robot within sensing radius of the robot while the length of the traveled distance is minimized. the problem can be seen as a variant of the classical *traveling salesman problem* (TSP) [14]. Thus, the coverage path planning is a challenging problem because the TSP is known to be NP-hard which makes it computationally very demanding, especially for finding the optimal solution for large instances.

In recent years, we have witnessed fast growth, improvement of UAVs and increased demand for solutions to various tasks including the herein addressed coverage problem. Many solutions for coverage problem for UAVs can be found in the literature, where we can identify two main classes of the vehicles. The first class represents copters like flying vehicles that also includes Vertical Take-off and Landing (VTOL) vehicles. The second class includes the fixed-wing flying vehicles that are similar to airplanes. The main advantage of copter and VTOL vehicles is that it can stop the flight, change its direction or altitude and then continue in a new direction. On the other hand, they are slower than fixed-wing UAV and cannot travel at long distances in the case of electric power, which is currently the dominated type of power for these small and micro aerial vehicles.

(a) Example of VTOL UAV. Credits: The Register[1]

(b) Example of copter UAV. Credits: Wikimedia[2]

Figure 1.1: Examples of UAVs.

Fixed-wing UAVs are non-holonomic vehicles, therefore its control and cover path planning can be considered as more complicated because it requires non-zero forward velocity and the motion has to respect the minimum turning radius, and therefore, the planning has to respect the vehicle motion constraints.

In this thesis, we focus on area coverage problem where the area to be covered can be represented by a convex polygon that is surrounded by obstacles by fixed-wing UAV. Thus, the vehicle is requested to cover the area at the given height above the ground, avoid obstacles, and keep a safe distance from the obstacles. Motivational scenario for the addressed problem is crop spraying for a field "hidden" inside the forest. The problem formulation can also be applied to find the cover path of a polygon retrieved by topological decomposition ([5]), because each such a polygon would be also surrounded by obstacles.

Figure 1.2: An example of a fixed-wing UAV. Credits: Wikimedia[3]

[1] https://regmedia.co.uk/2015/02/26/skyprowler_teaser.jpg?x=1200&y=794

[2] https://upload.wikimedia.org/wikipedia/commons/a/af/WMCH_Drone.jpg

[3] https://upload.wikimedia.org/wikipedia/commons/0/0d/Lehmann_Aviation_L-A_series_2016.png

The rest of this thesis is organized as follows. An overview of the existing approaches to the area coverage problem is presented in the following chapter. In Chapter 3, the problem statement is presented. Chapter 4 explains proposed method. Results are reported in Chapter 5 and finally concluding remarks are in Chapter 6.

CHAPTER 2

Related work

The Area coverage problem is widely described in the existing literature and various methods have been proposed. In general, the area coverage problem is the task which should meet the following requirements formulated in [3]:

- The Vehicle or robot is requested to cover area completely by covering all points in the given area.
- The Cover path has to avoid all obstacles.

Moreover, the cover path should minimize the selected optimization criteria to provide high-quality solutions. The quality indicator can be the shortest path, the fastest path, the most economic path, etc. Cover of the regular shapes can be based on predefined covering trajectories, e.g., mowing the lawn pattern. In the case the requested area to be covered is not of that regular shape, it is for example non-convex, then the area can be decomposed into simple regions such as squares, hexagons, etc. Each such a region can be then covered by primitive motions [3, 7, 15, 6, 8].

Part of the area is considered to be covered once it has been visited. In case of the vacuum cleaners, combine harvesters and similar robots, the captured area is equal to the whole body or frame of the robot.t On the other hand, e.g., for the UAVs, a part of the area is usually captured by a sensing device, for example by the onboard camera. Existing algorithms can be categorized, depending on how they approach and solve the coverage problem. In the rest of this chapter, the selected approaches are described regarding the categorization to offline/online planning and representation of the environment to graph based cellular decomposition based approaches.

2.1 Offline vs Online planning

Offline area coverage methods assume that complete information about the environment to be covered is available. However, the offline methods are naturally the only one, that may guarantee to find trajectory [3]. If a map of the environment is not available, it can be firstly created, e.g., by robotic exploration approach. Then, the map can be used to find the optimal solution. However, for dynamically changing environments, offline methods cannot be used, or at least the guarantee would not hold in such cases. Offline coverage path planning can be found in the automotive industry to create the cover path to spray car components that are precisely described by its design and the cover tasks are done repetitively in the well-known environment [13].

Using online methods, the robot does not need to have apriori knowledge about the environment to be covered. The robot has to collect data using sensors and plan it's path in real-time. It is hand down to it's sensing capabilities. Because of lack of apriori knowledge of the environment, these methods may not guarantee to find the optimal solution with respect to some global performance indicator such as the shortest/fastest covering path/trajectory because one can always find the scenario which will make the robot to do unoptimal decisions [16]. Because of that, these methods are sometimes called sensor-based coverage and they are capable to address the area coverage problem in dynamic environments.

2.2 Grid-based methods

One of the existing approaches is to represent the area by a grid composed of uniform grid cells, which has been firstly proposed in [17]. The authors consider a mobile robot with a sonar array to map the robot's surroundings represented by the grid. Each cell contains a value from the range $[-1, 1]$, representing the cell is probably empty (i.e., values lesser than 0), unmapped (values equal to 0) or probably occupied with the values greater than 0.

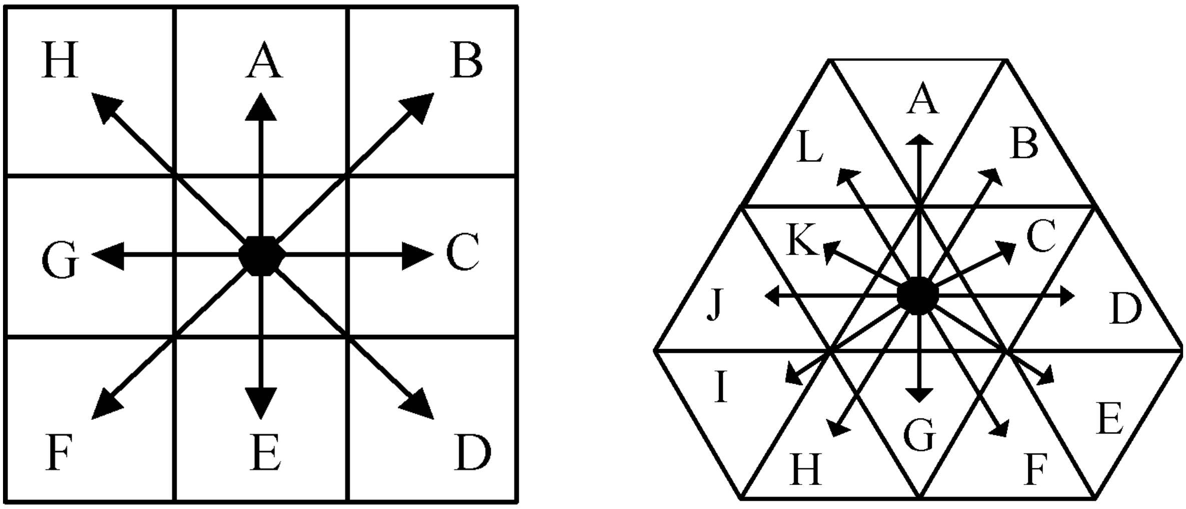
In area coverage scenarios, the cells values can be apriori known but grid representation can be also used by online methods where a robot collects data from sensors and based on the sensor data, the surrounding cells are updated and the requested cover path is replanned.

Regarding the coverage problem, the size of the cells can be the same as the robot's cover radius. The cell is considered to be covered once it is visited by the robot. The Final coverage trajectory is desired to visit all the cells to completely cover the requested area without visiting any cell more than once. Even though square-shaped cells are the most natural choice, other shapes such as triangles or hexagons have been considered as well [1].

Triangular shaped cells were proposed in [1] with the motivation to achieve better resolution than with the square cells of similar size [3]. Besides, the grid composed of squared cells allows 8-neighborhood, while triangular cells allow to considered 12-neighborhood as can be seen at 2.1, and thus the robot can choose from more directions to better fit the covering strategy. The triangular grid has been used for cleaning robots in the area with unknown obstacles and it is reported [1] that the triangular grid allows better maneuverability to avoid obstacles and more flexibility in following the optimal path. High-resolution grids such as triangular grids are advantageous for robots which are able to perform fine movement adjustments but because most of the field robots are unable to control their movement with required precision, it might not be worthy of spending computational effort in high-resolution grid [3].

Because cells are only approximating the real shape of real objects and boundaries of the area to be covered, [16] categorized the grid-based methods are classified as "approximate cellular decompositions." Grid-based methods are also resolution-complete [3] because cells uniform shape does not allow to precisely match real-world objects and shape of the environment and thus finer grid resolution.

The reported advantages of grid-based methods are that using a grid map is simple to implement, represent, and maintain [18]. On the other hand, the main disadvantage is a large memory consumption with an increase of resolution [1], which grows exponentially with the size of the area [18]. Besides, as any metric map, it requires accurate localization of the robot's



(a) Squared grid allows 8 neighborhoods to visit.

(b) Triangular grid allows 12 neighborhoods to visit.

Figure 2.1: Display of possible neighborhoods to visit in square and triangular grids, taken from [1].

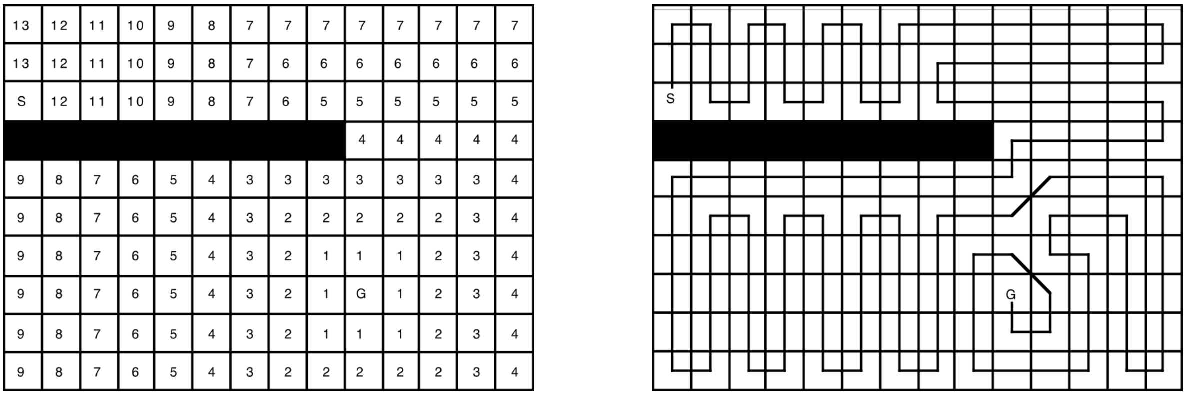
position and planning can be inefficient in comparison with a topological representation of the robot working environment [18]. Nevertheless, a metric map is required for precise coverage anyway, therefore existing solutions to the area coverage problem for the grid-based map of the environment will be introduced in following parts of this text.

2.2.1 Distance transform complete coverage

Using distance transform in the solution of the coverage area problem has been proposed in[19], to find a complete coverage path but not necessarily the optimal one. The proposed algorithm creates a coverage path between the given *start* and *goal* cells, which is suitable, e.g., for vacuum cleaning robots with multiple areas to be cleaned and the starting point can be defined as an entrance to the room and goal point as entrance to next room, so the final coverage path connects all rooms.

The coverage path is created in two steps. In the first step, a distance wavefront is propagated from the given goal location to the start location through the free space of the grid map of the environment. Once the distance value is assigned to each cell, the path starts to be determined from the *start* to the *goal* location following the steepest ascend. The robot then follows the determined sequence of cells until it reaches the goal. The wavefront is propagated around the obstacles, and therefore, it is ensured that robot does not collide with the obstacles. An example of the wavefront and final coverage path are illustrated in Figure 2.2.

The main disadvantage of this approach is that the final path has many turns and the average straight segment of the path is relatively short. Another disadvantage is that the robot's position has to be precise to avoid obstacles and to be able to follow the determined path but with the number of turns, the error in the navigation inevitably increases. One of the proposed solutions to address the navigation problem is to put external navigation beacons into the covered area. Besides, the algorithm can be improved by introducing following contours of the environment and minimizing the number of the turns of the cover path.



13	12	11	10	9	8	7	7	7	7	7	7	7	7
13	12	11	10	9	8	7	6	6	6	6	6	6	6
S	12	11	10	9	8	7	6	5	5	5	5	5	5
■	■	■	■	■	■	■	■	■	4	4	4	4	4
9	8	7	6	5	4	3	3	3	3	3	3	3	4
9	8	7	6	5	4	3	2	2	2	2	2	3	4
9	8	7	6	5	4	3	2	1	1	1	2	3	4
9	8	7	6	5	4	3	2	1	G	1	2	3	4
9	8	7	6	5	4	3	2	1	1	1	2	3	4
9	8	7	6	5	4	3	2	2	2	2	2	3	4

(a) Wavefront propagation from the goal to start cell.

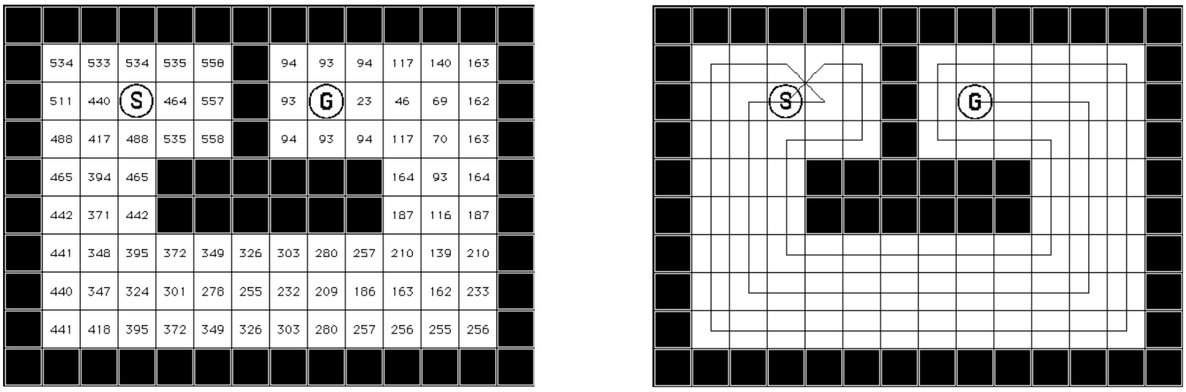
(b) Final cover path from the start to goal cell.

Figure 2.2: Example of the distance wavefront propagation and the cover path determined by the approach proposed in [2].

The improved algorithm adds the third step into the original algorithm to propagate another distance wavefront from the obstacles. The final value of the cell is computed from the modified distance wavefront from the goal defined as

$$PT(c) = min_{p \in P} \left(length(p) + \sum_{c_i \in p} \alpha \; obstacle(c_i) \right), \tag{2.1}$$

where P represents a set of all paths from the cell c to the *goal*, and therefore, $p \in P$ is a single path. The function $obstacle(c)$ is cost the function using the values generated by the wavefront from the obstacles. Finally, $\alpha \geq 0$ represents the weight which determines how much the final path avoids obstacles. An example of the resulting path created by this modified algorithm is displayed in Figure 2.3.



■	■	■	■	■	■	■	■	■	■	■	■	■	■
■	534	533	534	535	558	■	94	93	94	117	140	163	■
■	511	440	S	464	557	■	93	G	23	46	69	162	■
■	488	417	488	535	558	■	94	93	94	117	70	163	■
■	465	394	465	■	■	■	■	■	■	164	93	164	■
■	442	371	442	■	■	■	■	■	■	187	116	187	■
■	441	348	395	372	349	326	303	280	257	210	139	210	■
■	440	347	324	301	278	255	232	209	186	163	162	233	■
■	441	418	395	372	349	326	303	280	257	256	255	256	■
■	■	■	■	■	■	■	■	■	■	■	■	■	■

(a) The improved wavefront propagation from the goal and the obstacles to the start cell.

(b) The improved final path from the start to goal cell.

Figure 2.3: Example of the improved distance wavefront propagation and the improved cover path determined by the approach proposed in [2].

2.2.2 Spanning trees

In [20], Gabriely and Rimon propose a solution to the complete coverage problem in square grid environments based on the new decomposition of the grid and creating a spanning tree connecting the centers of the grid cells. In the first step, the original grid is decomposed into a new grid, which contains four square cells instead of one original square cell. Only the orthogonal directions are allowed, and therefore, the robot can move only in four directions. In the second step, the spanning tree is created from the given starting cell in the counter-clockwise order to "visit" all four adjacency cells if possible. The second step is repeated until all the cells are not connected by the spanning tree. Once all the cells are connected, the cover path can be created from the start cell in a clockwise or counter-clockwise order and the path ends again in the start cell.

The cover path passes all-new smaller cells and follows the spanning tree graph contour, where each node represents the center of the original cell and the edge represents the link between the cells. Only the starting cell has to be given and it has to be a free cell, therefore the algorithm can be considered to be an online method and the method has been also deployed and tested online as follows. After a new cell is explored, orthogonal cells are scanned and spanning tree is updated. Note that the method finds the complete cover path of the area without overlaps, but is not suitable to be used in areas with obstacles.

In the same paper [20], the authors propose an extension suitable for areas with obstacles, but the final trajectory can visit some cells more than once. The Spanning tree method is suitable for 2D grid environments composed only from squared cells. An example of the final path found by the spanning tree based algorithm is depicted in Figure 2.4.

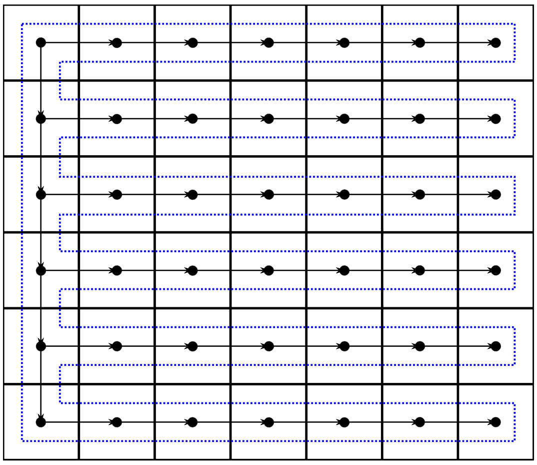


Figure 2.4: An example of the spanning tree and the final cover path of the squared grid environment.

2.2.3 Neural networks

Luo and Yang in [11] and later in [21] propose to utilized neural networks to address the grid-based complete coverage problem. The authors consider sweeping robots in a time-varying environment. The neural network is utilized to provide the robot the next best direction for continuing with the coverage. Since the squared grid is used, only four directions are required to be encoded in the neural network. There are two factors used in a determination of the next cell to be visited.

The first factor is a value obtained from the neural network, where the neurons are connected with neighboring neurons representing the adjacent cells and neural network returns values in the range $[E, -E]$ for each cell, where E is a huge constant parameter. If the value returned from the neuron is equal to E, it means that cell is uncleaned; if the value is equal to 0, the cell is already cleaned; and finally, the value $-E$ represents that the cell contains an obstacle.

The Second influencing factor in the determination of the next cell to visit is the turning angle between the current moving direction and the new possible direction. Because each turn is penalized (i.e., to minimize the number of turns), it is preferred to continue with the already performing turning until it is possible instead of stopping and moving wheels to different turn angle.

These two values are added together for each adjacent cell and cell with the highest value is visited next. The authors deployed the algorithm on multiple robots with different starting positions and let them to completely sweep the area. Each robot was counted as an obstacle in the grid of the other robots.

2.3 Cellular decomposition

Another class of the existing methods is based on a decomposition of the area to be covered into multiple subregions. Each subregion is covered independently using only simple motions. Within this context, the subregions are called cells and they can be convex as well as non-convex, and every cell can have different size and shape. Before a cell is covered, an adjacency graph is created and the optimal order of visits to the cells is found by the exhaustive walk. Note that, the problem of determining the optimal order of visits of the cells is equivalent to the traveling salesman problem (TSP).

Cellular decomposition methods are more efficient than grid methods and provide better solutions with less memory consumption. Besides, the given are to be coved can also be better approximated, e.g., using the specific shape of the coverage trajectories for individual cells. Therefore, the following sections describe existing approaches to the cellular decomposition that are considered related to the motivational problem addressed in this thesis.

2.3.1 Trapezoidal decomposition

Trapezoidal decomposition is an offline method which can cover only areas represented by a planar polygon populated by polygonal obstacles [3]. The area to be covered is decomposed into cells which are determined by vertices of the obstacles [4] and it works as follows. First, the preferred direction must be specified. Once the direction is specified, a perpendicular line to the direction is created. The line is called sweep line and it is moved from one side of the area to opposite one. Each time it intersects with a vertex of the obstacle, the area is split across the line and side of the obstacle. By going through the area completely, it gets decomposed into multiple convex trapezoidal cells. Each cell is remembered as a node in the adjacency graph from which the optimal sequence of visits to the cells is determined and each cell is then covered by the particular cover trajectory.

This method is simple to implement and provides the to complete coverage. Its main disadvantage is that it decomposes area more than it is needed, and therefore, the determined cover path is not optimal. An example of the area decomposition is shown in Figure 2.5.

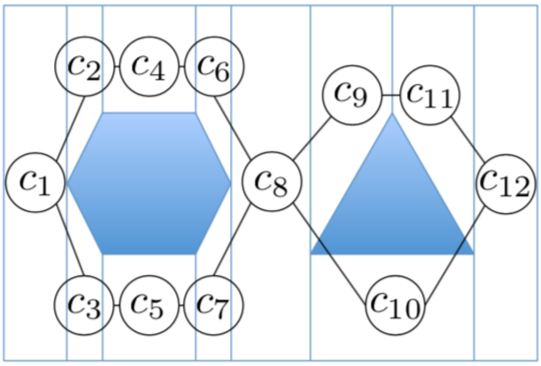
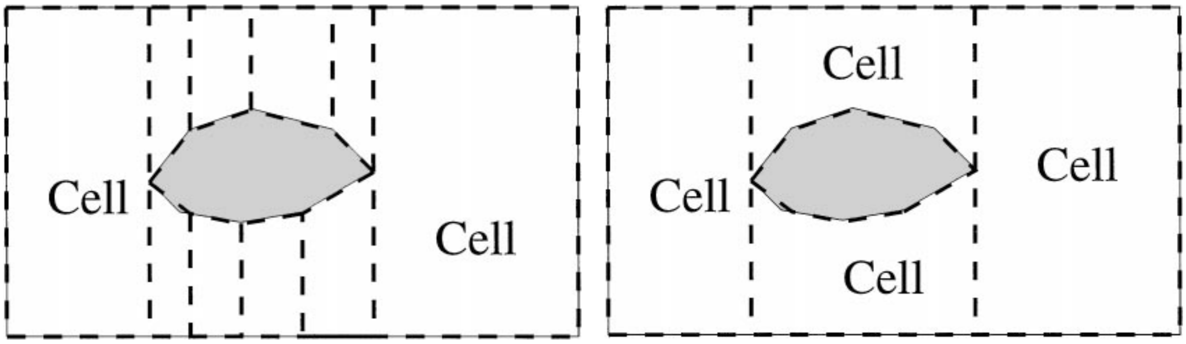


Figure 2.5: Example of the trapezoidal decomposition of the environment containing two obstacles and twelve cells, taken from [3].

2.3.2 Boustrophedon decomposition

Boustrophedon decomposition is a method developed by Choset in [4]. It is similar to the trapezoidal decomposition, but it merges cells into larger ones (if possible), and therefore, it is able to find better solutions. The algorithm searches the area for vertices of obstacles but it divides the area only if the vertical segment cannot be extended above and below the vertex, i.e., if the sweep line intersects with the obstacle in exactly one point then new cells are formed. Therefore, cells are created upon an entry and exit of the obstacle, see Figure 2.6 for an example.

The cells created by the algorithm are not convex but can be covered by simple motions as well. It is the offline method similar to trapezoidal decomposition because it requires apriori knowledge of the obstacles.



(a) Trapezoidal decomposition.

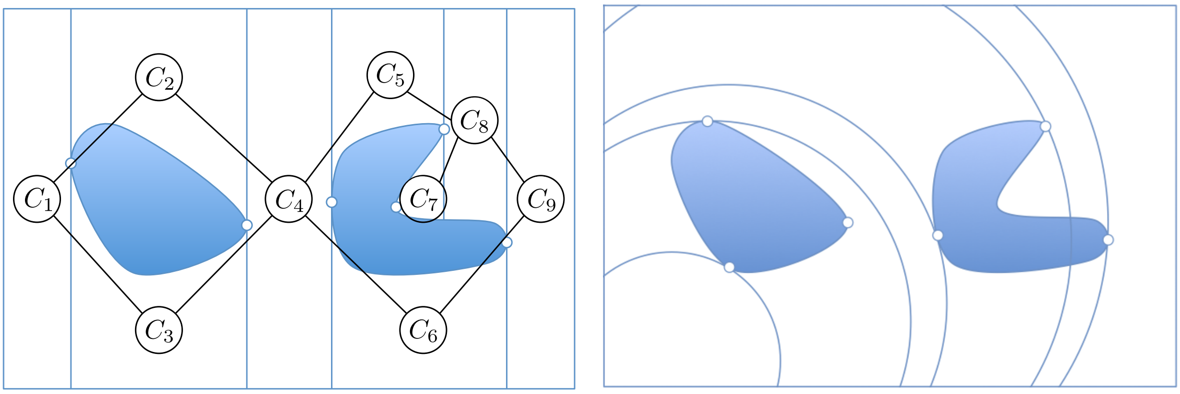
(b) Boustrophedon decomposition.

Figure 2.6: Example of the trapezoidal and boustrophedon decompositions, taken from [4].

2.3.3 Morse-based decomposition

Morse-based decomposition proposed by Acar and Choset [22] is capable of decomposing areas with non-polygonal obstacles, unlike trapezoidal and boustrophedon decomposition methods. Morse function $h : \mathcal{W} \to \mathbb{R}$, where $\mathcal{W}$ is robots workspace has to be defined to find the critical points. The area is then divided at the critical point across a curve defined by Morse function and the boundary of the obstacle. The critical points are thus determined using Morse function restricted to the obstacle boundaries. The critical point is found whenever the function is not differentiable or all its partial derivatives are equal to 0. It can be also imagined as the case where the normal vector to the boundary would be constructed and whenever it is perpendicular to the curvature defined by Morse function, the area is divided across this curvature, e.g., see Figure 2.7. Notice, the boustrophedon decomposition is a special case of Morse function $h(x, y) = x$.

The advantage of Morse-based decomposition is that it can work with non-polygonal obstacles and it can divide the area into cells of various shapes as it only depends on the selected Morse function. Another advantage is that it can be implemented and used online. Unfortunately, this method is not able to determine critical points in rectilinear environment [3].



(a) Morse function decomposition where the Morse function is $h(x, y) = x$.

(b) Morse function decomposition where the Morse function is $h(x, y) = \sqrt{x^2 + y^2}$.

Figure 2.7: An example of the Morse function decomposition of the same area with the use of two different Morse functions, taken from [3].

2.3.4 Topological decomposition

In [5], Wong and Macdonald develop a topological decomposition method that does not require prior knowledge about the environment, and therefore, it can be used online. The topological decomposition method creates unregular subregions and new subregion is created each time a new landmark is visited. The environment is then represented by the graph G, where the nodes $N(G)$ represent subregions and the edges $E(G)$ represent connections between the adjacent subregions. Each edge $E(G)$ also contains a path between the nodes $N(G)$ and approximate distance between them.

There can be four types of the nodes $N(G)$: convex, concave, unexplored and joint. The convex and concave nodes can be covered by zig-zag patterns. The unexplored nodes are not visited and covered yet. The joint nodes are created each time the robot is covering an unexplored subregion. Because it does not know where the subregion ends, it creates the node of the joint type at the end. Once it visits the landmark, the joint is transferred to one of the first three types of the nodes and the subregion is closed.

The algorithm is represented by a finite state machine consisting of three states: normal, travel, and boundary. The robot is in a normal state while it is covering a subregion. Anytime it visits the landmark, it switches into the boundary state, because landmarks can be located only at the borders between subregions. If the robot has to move from the current subregion to the next subregion, it is switched into the travel state.

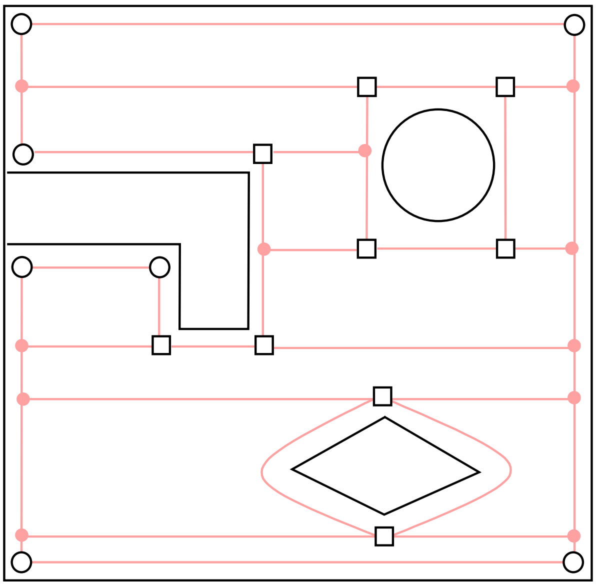


Figure 2.8: An example of topological decomposition of an area with a multiple obstacles of various shapes, taken from [5].

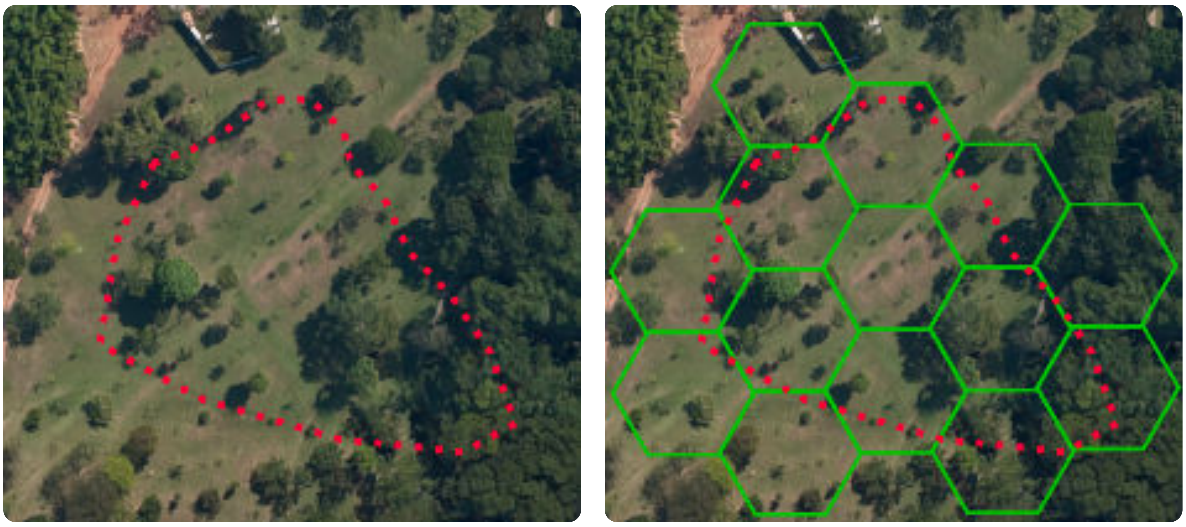
The proposed method requires to always start in the corner of the area. It is not shortcoming prerequisite because a robot can get to the corner by simple motions from any position in the area. The corners are handled as landmarks. Once the robot finds the corner, the graph G is updated and then it switches to the normal mode and starts to cover the newly created subregion. Each time the robot visits the landmark, the graph G is updated. Once the subregion is covered, the robot switches to the travel mode and visits the next subregion. Once all nodes in G are explored and visited, the area coverage is complete. An example of the decomposed area by the topological decomposition is shown in Figure 2.8.

2.3.5 Hexagonal decomposition

In [6], the authors propose a hexagonal decomposition of the user-specified area defined by the polygon P and the height map of the environment. The area is covered by multiple copter-like UAVs and the decomposition is performed as follows.

First, the hexagonal grid H is created and overlayed over the polygon P. Each hexagon has the same radius r. Then a subset of the hexagons H' is defined by the hexagons which are inside of the polygon P or are in a contact with its border. Then, a user can prohibit some hexagons from being covered for example in the case where it contains an obstacle. Each hexagon cell or multiple cells are assigned to the closest UAV localized at the nearby station or entry point. For each assigned set of cells, the shortest order of visits to them is determined and then each cell is covered by lawn moving like motions. Finally, all trajectories are projected on to height map.

This method is an offline method, where the environment and obstacles are supposed to be known prior the planning and the path is pre-computed before the execution. The determined paths are uploaded to the to UAVs and they execute the mission. The algorithm can also be used to find the complete cover path for single UAV. The disadvantage of the method is that hexagons will not cover not only the area inside polygon P but also an area around it. Therefore, the final path is not optimal and robots cover also the area which is not inside of the area of the interest. An example of the hexagonal decomposition is shown in Figure 2.9.



(a) Polygon selection.

(b) Hexagonal decomposition of polygon.

Figure 2.9: An example of hexagonal decomposition of given polygonal area to be covered, taken from [6].

2.3.6 Polygonal decomposition

Maza and Ollero [7] proposed to address area coverage problem by the method originally developed by Hert and Lumelsky [23]. The method decomposes a given convex polygon P into set of n convex subpolygons P_i each with the specified size c_iP, where c_i is a selected from the range $0 < c_i < 1$, and $\sum_{i=1}^{n} c_i = 1$. Each subpolygon P_i must also contain a specific point (site) S_i, which can represent starting point of the cover path for the robot. Each sub-polygon P_i can be covered by simple zig-zag motions as is it is shown in Figure 2.10. The decomposition algorithm works as follows.

The algorithm expects a list of the vertices V which define the polygon P and a list of the sites S. Each site contains a demand on the size of the sub-polygon. Then, a line L is created from the first vertex v_1 to the first site S_1. The starting point of the line is then used as the pivot and the ending point is moved in the counter-clockwise (or clockwise) order until a sub-polygon created by the division of the polygon P by the line L that does not match the demand on the site S_i. If the newly created sub-polygon has more sites, then the sub-polygon has to be large enough to fulfill the demands of all sites and thus it is further decomposed using the same procedure until all the demands of sites are fulfilled.

The polygonal decomposition has been used in [7] to cover the area by a team of helicopters. Each helicopter covers one sub-polygon by zig-zag motions. While Maza and Ollero used helicopters to cover the area, they did not include any obstacles and the whole problem was reduced to the 2D task. Hert and Lumelsky also created an extension to their algorithm to decompose convex polygons with obstacles.

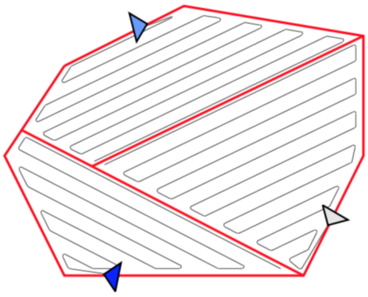


Figure 2.10: An example of polygonal decomposition of the polygon to be covered by helicopters into three subpolygons, taken from [7].

2.4 Random

In addition to the aforementioned deterministic approaches, random walk based approach can be utilized to eventually achieve a complete coverage [3]. The randomized area coverage method can be for example found in cheap autonomous vacuum cleaners. The main advantage of this method is that it does not rely on any expensive sensors nor any expensive computational equipment. The disadvantage is that finding an optimal solution in a limited time is not guaranteed similarly to the guarantee to not visit some parts of the area multiple times. Besides, the method is not suitable for complex 3D environments because the random approach would provide unjustifiable long and time-consuming trajectory.

Gage in [12] studied the randomized approach from the cost/benefit point of view. In his work, he compared the methodological and randomized approaches to localize targets in a given area by a team of robots. He considered imperfect sensors which do not guarantee the detection of the target within the sensing. The reported results show that once the probability of the detection decreases, advantages of the methodological search over the randomized approach diminish. At some point, the randomized approach becomes more efficient because the methodological complete coverage does not guarantee to localize the targets and robots using random approach are less costly as the mentioned above.

2.5 Reference Method

In [8] is proposed a method to find the cover trajectory of given area without the obstacles that work as follows. In the first step, this algorithm decomposes an area into multiple cells using the Boustrophedon cellular decomposition [24] and then finds a Eulerian path connecting all newly created cells. Once the order of visits is decided, each cell is covered by the lawn mowing pattern like trajectory displayed in Figure 2.11. The authors of the algorithm [8] deployed the method with fixed-wing UAV and covered vast areas. The Boustrophedon decomposition is used to determine, which parts of the area can be covered and which contain obstacles.

Because the final trajectory contains "sharp" turns at the end of covering segments, the fixed-wing UAV could not follow it precisely. The authors [8] proposed to minimize the difference between the planned trajectory and the real flight by making a circle at the end of each segment before it is connected to the next one. The difference between the planned and real flight is shown in Figure 2.12.

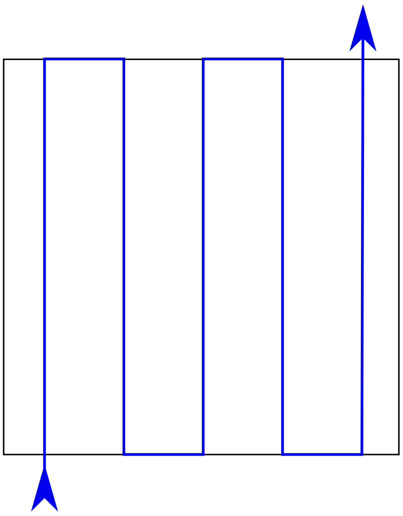
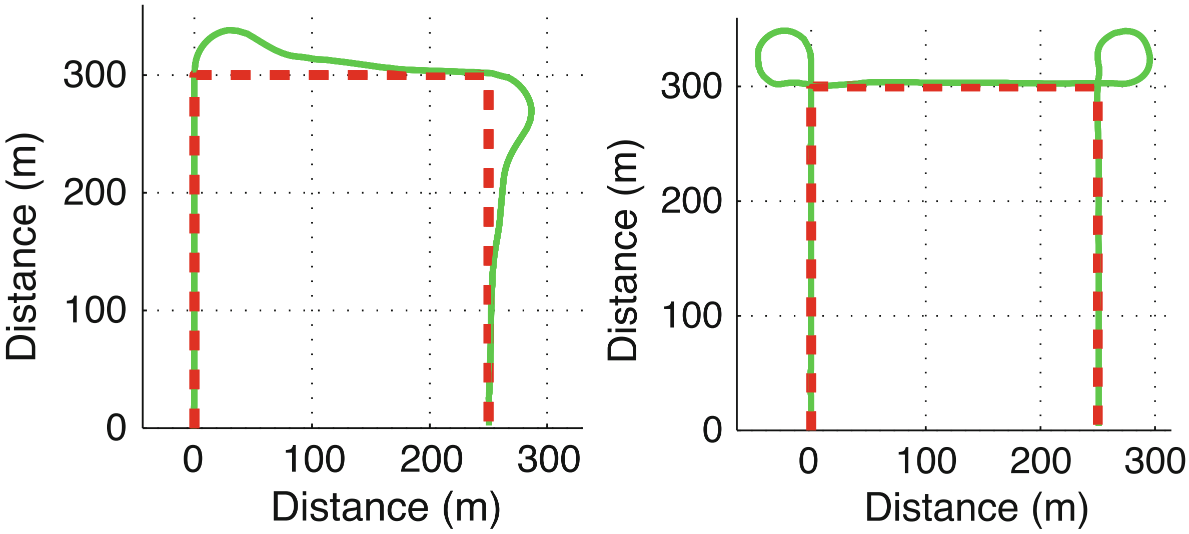


Figure 2.11: Cover path produced by the method proposed in [8].



(a) Direct connection.

(b) Circular connection

Figure 2.12: Difference between planned trajectory and real flight, taken from [8]

If the circular connection is used, the difference between planned trajectory and the real flight is minimized, but the final trajectory is much longer.

2.6 Conclusion

Regarding the presented overview of the existing methods for area coverage problem, the coverage of the non-convex or complicated areas can be addressed by decomposition into regions that can be covered by the best fitting “zig-zag” pattern, i.e., in a dominant direction of the region. The obstacles in the environment are treated in a similar way in the existing methods. Therefore, we restrict the studied problem to coverage of convex region with a dominant direction and without obstacles inside. However, we focus on satisfying the motion constraints of fixed-wing vehicle and tradeoff between the full coverage at the requested height above the terrain and safe distance from the obstacles that are surrounding the area to be covered. The addressed problem is formally defined in the following chapter.

CHAPTER 3

Problem Statement

This Chapter formally defines the addressed area coverage problem with a fixed-wing UAV. The problem is to find an optimal trajectory which visits all points within the given sensing radius inside of the given area. The herein studied problem is considered under the following assumptions:

1. The area to be covered is a convex polygon with
2. *Dominant orientation* of the coverage and
3. it is obstacle free and with only negligible terrain altitude changes.

The UAV is considered to be equipped with a coverage device with a limited range that can be larger than the dimensions of the robot, e.g., camera or LiDAR. Thus, the area covered by a single flight over is limited based on the device limitations and the cover radius r can be defined as

$$r = 2z\sin\left(\frac{\alpha}{2}\right), \tag{3.1}$$

where z is the altitude of the flight and angle α represents the device field of view (FOV). The device is connected to the UAV perpendicularly, and it is downward-looking. The value of r is the actual radius within which the area is covered. Therefore, the coverage of the given convex polygon with the defined dominant direction is considered as coverage of a finite set of covering segments $\mathcal{P}$:

$$\mathcal{P} = \left\{\left\{s_1^1, s_1^2\right\}, \left\{s_2^1, s_2^2\right\}, ..., \left\{s_n^1, s_n^2\right\}\right\} \tag{3.2}$$

$$s_i^1 = \left(p_i^1, p_i^2\right), \tag{3.3}$$

$$s_i^2 = \left(p_i^2, p_i^1\right), \tag{3.4}$$

where each segment is defined by two endpoints $p_i^1, p_i^2 \in \mathbb{R}^3$. All the segments are parallel with the heading defined by the dominant direction of the polygon and are at the same altitude z. Because each segment can be approached by the UAV in two directions, the segments form collinear pairs $c_i = \{s_i^1, s_i^2\}$ with mutually exchangeable endpoints. The non-collinear segments are spaced by the r distance, and thus the UAV covers a part of the area within the $\frac{r}{2}$

distance from a particular segment when following the segment. The reason why the covering segments are defined is that the pattern from the straight segments connected by turns at the border of the area is the most efficient pattern to cover the polygon [7]. The area can be approached from multiple directions; however, the minimal number of the segments needed is for the dominant orientation of the area, which also minimizes the length of the required trajectory [7]. Respecting the dominant orientation also requires the minimal number of the required turns which is desirable because each turn can significantly increase the trajectory length and slows down the coverage.

The coverage plan is defined by the sequence of visits to the segments which can be solved as a variant of the TSP for which we need to determine all possible shortest trajectories connecting all the covering segments by the corresponding maneuvers connecting the endpoints of the segments. Since in general we are requesting 3D trajectories for the fixed-wing vehicle, the shortest maneuvers are determined using the *Dubins Airplane Model* [25] for which the motion of the vehicle can be described as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta\cos\psi \\ \sin\theta\cos\psi \\ \sin\psi \\ u_\theta \rho^{-1} \end{bmatrix}, \tag{3.5}$$

where x, y, z are the position coordinates, θ is the vehicle heading angle, ψ is the vehicle pitch angle, v is as a given constant forward velocity, and $u_\theta \in [-1, 1]$ is the control input. The motion of the vehicle is constrained by:

1. the minimal turning radius ρ,
2. the limited pitch angle, i.e., $\psi \in \langle \psi_{min}, \psi_{max} \rangle$.

However, a simplified version without a dynamic model of the vehicle is considered, and therefore, it is allowed the pitch angle can be changed arbitrarily within the limits ψ_{min} and ψ_{max} and the configuration space of the vehicle can be defined as

$$\mathcal{C} = \mathbb{R}^3 \times \mathbb{S}^1, \tag{3.6}$$

where $\mathbb{S}$ is the space of all possible headings of the vehicle. The shortest possible trajectory covering all the requested polygon can be found as an optimal sequence of the covering segments that are connected by the optimal (and feasible) maneuvers respecting the Dubins Airplane Model (3.5). For $2n$ covering segments including the collinear pairs, the sequence can be described as a permutation Σ

$$\Sigma = (\sigma_1, \sigma_2, ..., \sigma_n), \tag{3.7}$$

where $1 \leq \sigma_i \leq n$ represents mutually exclusive index of the segment pair $c_{\sigma_i} \in \mathcal{P}$. Because each segment can be approached from two directions, it is also required to define a binary sequence describing the direction in which the segment is approached and thus determine the next segment is connected to p_i^1 or p_i^2 which is encoded in the variable d_i

$$d_i \in \{1, 2\}. \tag{3.8}$$

Since all segments have to be passed (in one or the opposite direction), we can consider a set binary sequence D of the same length as the sequence Σ :

$$D = (d_1, d_2, ..., d_n) . \tag{3.9}$$

Solution to the above-described variant of TSP problem leads to minimalization problem of the length of the trajectory T which is composed of $2n$ maneuvers m_i, while only the sequences of visits Σ and directions D can be changed. Because the final cover trajectory has to respect fixed-wing vehicles motion constraints defined by (3.5), the maneuvers are determined by *Dubins maneuvers* [26] further described in Section 4.2. Dubins maneuvers are shortest curves between two points in the plane. Because we consider a three-dimensional environment with the obstacles around, the Dubins maneuvers need to be further optimized to avoid the obstacles and still respect fixed-wing vehicles motion constraints. However, once such a maneuver is determined, its length is fixed and utilized in the solution of the TSP, but in general, maneuver m_i is defined as:

$$m_i : \mathbb{R} \to \mathcal{C}, \tag{3.10}$$

where the definition field of m_i is $\mathcal{D} = \langle 0, 1 \rangle$. Each maneuver is connected to the next covering segment in the sequence and it is represented by the set of points in 3D space. Since the covering trajectory is requested to visit all the covering segments, the sequence of maneuvers consists of covering segments connected by the maneuvers connecting the endpoints of the covering segments. Thus, the solution to TSP problem can be formally defined as

$$\text{minimize}_{\Sigma,D} \; T = \sum_{i=1}^{2n} \mathcal{L}(m_i) \; , \tag{3.11}$$

subject to

$$\begin{aligned}
&m_i(1) = m_{i+1}(0) \, , && \forall i \in 1, \dots, 2n \, , && (3.12)\\
&m_{2i-1} = s_{\sigma_i}^{d_i} \, , && \forall i \in 1, \dots, 2n \, , && (3.13)\\
&m_{2n}(1) = m_1(0) \, , && && (3.14)\\
&\Sigma = (\sigma_1, ...\sigma_n) \, , && 1 \le \sigma_i \le \sigma_n \, , && (3.15)\\
&D = (d_{\sigma_1}, \dots, d_{\sigma_n}) \, , && d_{\sigma_i} \in \{1, 2\} \, , && (3.16)
\end{aligned}$$

where the permutation index σ_i denote the order in which the covering segment is visited and $d_{\sigma_i} \in D$ denotes the particular orientation in which the covering segment is followed. Notice that the maneuvers are fixed (i.e., either covering segment or the connecting maneuver), once they are determined, and therefore, the problem is purely combinatorial to determine the sequence of visits to the covering segments and the orientation in which the segment is approached.

CHAPTER 4

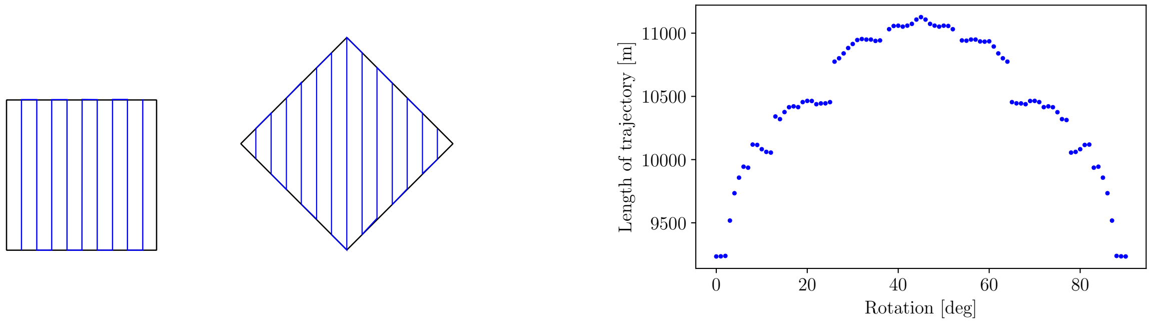
Proposed Solution and Used Methods

This chapter describes the developed solution to the proposed problem formulation of the area coverage problem with the convex polygonal area without any obstacles but with obstacles surrounding it. The proposed method is composed of existing approaches to similar problems. Among the used methods the dominant polygon orientation is determined by the approach of [7], the method for computing optimal maneuvers between the states of the Dubins Airplane model [25]. Problem 3 is considered as a variant of the GTSP that is transformed by the TSP using the Noon-Bean transformation [27], and the final TSP is solved by the existing solver LKH [28].

4.1 Determining the Dominant Orientation of the Polygon to be Covered

In the area coverage problem, the direction from which the area is covered influences the total length of the final coverage path as it is shown in [7]. Unfavorable direction can lead to a path, that consists of the non-optimal number of the straight segments and unnecessarily high number of turns. Because each turn increases the total length of the coverage path, it is therefore important to select the best direction how to approach the coverage of the polygon that we call the dominant direction, and coverage of the polygon by straight segments corresponding to the dominant direction would minimize the number of required turns and thus save the trajectory length. At this point, the optimal connection of the segments is relaxed and determination of the optimal maneuvers connecting the segments and respecting Dubins Airplane Model (3.5) is addressed in Section 4.2.

Probably the most efficient method to the find dominant direction is to find necessary spacing from the knowledge of the coverage radius. Once the spacing is known (i.e., from 3.1), parallel lines covering the area can be determined. The number of parallel lines from which the area is covered is minimal for the dominant direction. However, in general, the area can be approached from an arbitrary direction, and thus the number of possible directions is practically unlimited. Therefore the authors of [7] propose to determine the required number of parallel lines needed to completely cover the whole requested area instead of examining every possible direction. Since we assume the area is of the polygonal shape that is convex, the authors proposed to determine the dominant direction considering coverage of the area parallel segments that are parallel to the particular sides of the polygon. Thus all sides of the polygon are iteratively examined and the side with the minimum required segments is selected as the dominant direction. An example of the influence of cover direction to the length of the coverage path is shown in Figure 4.1.



(a) Influence of direction on the number of turns for 0 and 45 degrees.

(b) Influence of the direction on the path length.

Figure 4.1: Influence of selected direction on path length with zig-zag pattern of the squared shaped area with the one side $500m$ long.

4.2 Trajectory planning

Once the dominant direction is selected, the parallel segments can be defined. Each segment has to be connected by a turning maneuver with other segments to create the requested continuous cover path.

First, we assume there are no obstacles around the area to be covered and the feasible optimal trajectories connecting the segments are found as Dubins maneuver. Dubins maneuver is defined in 2D space and it is proven [26] that the shortest trajectory between two points with the prescribed heading of the vehicle consists only from arcs of the minimum turning radius ρ and straight lines.

Dubins maneuvers are determined as follows. Having the start point and end point together with the required headings, four circles are created. The radius of each circle is equal to ρ and the position of its center lies at the orthogonals to the heading vectors and it goes through the beginning and end points of the maneuver. Then, all the tangents between all four circles are created. The vehicle then follows one of the circles until it reaches the tangent point on it. Then it stops to follow it and starts to follow straight line leading to the next tangent point located at one of the circles connected to the endpoint. Once it reaches the next tangent point, it again starts to follow the circle until it reaches the endpoint. An example of creating Duibins maneuver is depicted in Figure 4.2.

There are four possible paths between two points in the 2D space depending on the prescribed headings at the points and also the mutual distance between the points they can consist of the left-handed or right-handed arc that is connected to the straight line connected to an arc that can again be left-handed or right-handed that forms the so-called CSC ma-

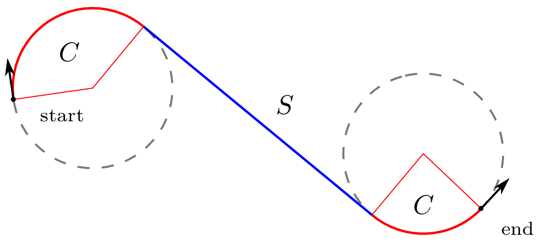
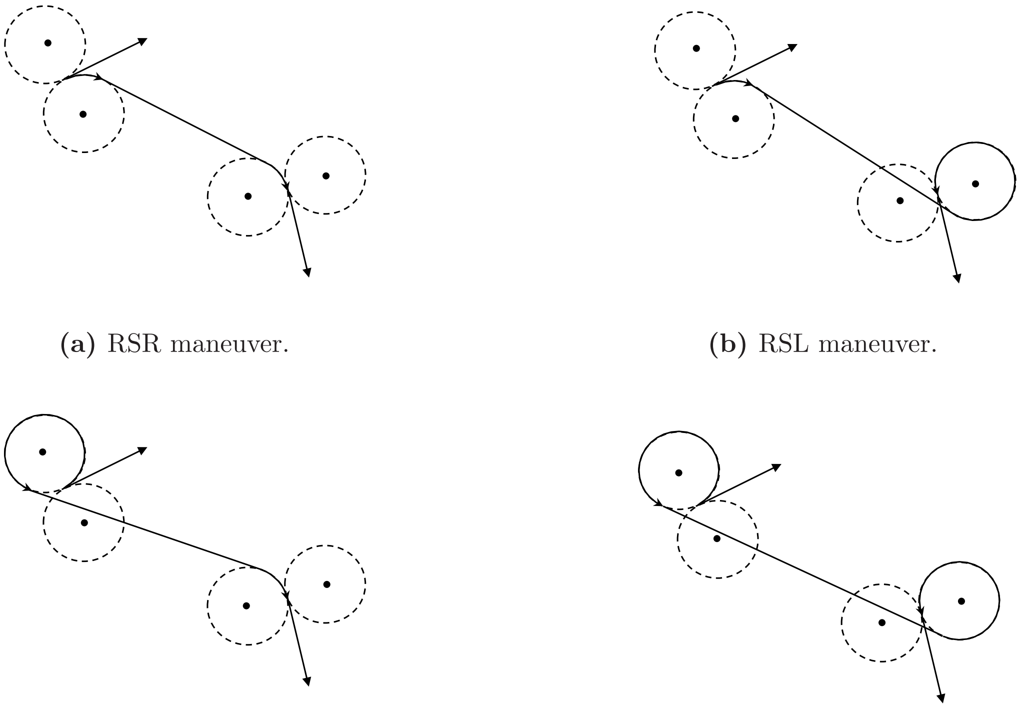


Figure 4.2: An example of Dubins maneuver construction

neuver. A combination of the left-handed arcs, right-handed arcs, and the straight line can create maneuvers, which are referred to as LSL, LSR, RSL, and RSR. All four kinds of the maneuvers are displayed in Figure 4.3. The shortest maneuver is called Dubins path (Dubins maneuver). There also exists special cases of maneuver, which consists of three arcs only, i.e., the so-called CCC maneuver. It is created only when the starting point and final point are too close, therefore there is not enough space between them to connect arcs by the straight line. In the area coverage scenario presented in this work, this kind of maneuver is not used, because it is more efficient to connect segments which do not require to be connected by this maneuver, i.e., connecting two segments by the CCC maneuver will always be longer than connecting the first segment with another segment with the CSC maneuver.



(a) RSR maneuver.

(b) RSL maneuver.

(c) LSR maneuver.

(d) LSL maneuver.

Figure 4.3: Types of Duibins maneuvers, taken from [9]

The Dubins maneuvers are created between all segments and an example is shown in Figure

4.4. Because the final cover path is determined from all these maneuvers, the corresponding lengths of the maneuvers are stored in matrices, i.e, distances in the distance matrix **D** and the corresponding maneuvers in the maneuver matrix **M**. Note that the covering segments define the heading of the vehicle when approach and also leaving the segment, therefore, the closed form solution can be utilized and determining such 2D maneuvers is not a computational issue.

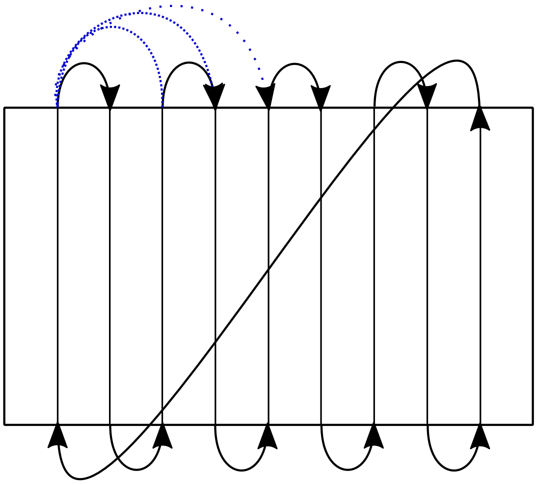


Figure 4.4: Creation of dubins maneuvers between all instersection points and determination of the final path.

4.3 Obstacle avoidance

The area to be covered is obstacles free; however, the obstacles are around, and therefore, it is necessary to determine maneuvers that are free of possible collisions with the obstacles. Because each maneuver is created only between points defined by the intersections of the segments and the border of the polygonal area P to be covered, only an altitude above the terrain is considered while maneuvers are created. Thus, each maneuver from the maneuver matrix **M** is checked for a collision with any obstacle surrounding the area. If the maneuver intersects with an obstacle, it is adjusted to avoid a possible collision. The collision check is performed for a sampled maneuver and every such sampled point is evaluated for possible collision with the obstacles. Each colliding point is then risen up to a new altitude level without collision and thus the whole maneuver is forced to have the flight altitude above any obstacle. Such a change of the maneuver violates the motion constraints of the fixed-wing vehicle, albeit multi-rotor UAV can follow it. After the maneuver was changed, from its points a convex envelope is created. Therefore a convex envelope [29] is created to satisfy the motion constraints of Dubins Airplane model. In particular, we do not modify the turning radius, but the climbing/diving angles have to respect the limited pitch angle. The convex envelope is created in the following three steps.

In the first step, the point of the maneuver that is at the highest altitude is found. The point is used to split the maneuver into two parts. Each part can be covered by the envelope separately. The first part is considered to be ascending part of the maneuver, i.e., the UAV has to climb towards the highest altitude of the maneuver, while the second part is descending to reach the height above the terrain required for the coverage.

In the second step, the significant points in each part are found. A significant point is determined from the ascending (descending) vector $\vec{w}$:

$$\vec{w} = (w_x, w_y) \tag{4.1}$$
$$w_x = \delta \tag{4.2}$$
$$w_y = \delta \tan(\psi), \tag{4.3}$$

where δ is the sample distance and ψ is the ascending (descending) pitch angle. Because each maneuver is sampled and each sample point is at the distance δ from the neighboring sample points, the convex envelope can be created from the points in the $x - z$ plane which is depicted in Figure 4.5.

Then, each point from given part is iteratively selected. A new vector $\vec{q}$ is created from the previous point to newly selected one:

$$\vec{q} = (q_x, q_y), \tag{4.4}$$
$$q_x = p_{i_x} - p_{(i-1)_x}, \tag{4.5}$$
$$q_y = p_{i_y} - p_{(i-1)_y}. \tag{4.6}$$

A cross product between the vectors $\vec{w}$ and $\vec{q}$ is counted and if the result is greater then 0, it means that the point is at the higher altitude than the previous one. Such a point is considered to be a significant point and it is stored in the list L. The last step of searching for the significant points is the examination of the list L to discard any points from L which are below or at the same height as the newly found significant point. It is again done using the cross product between the vector from the current point in the list L to the newly added point and the vector from the previous point to the current point, which are both in L. This process is displayed in Figure 4.5.

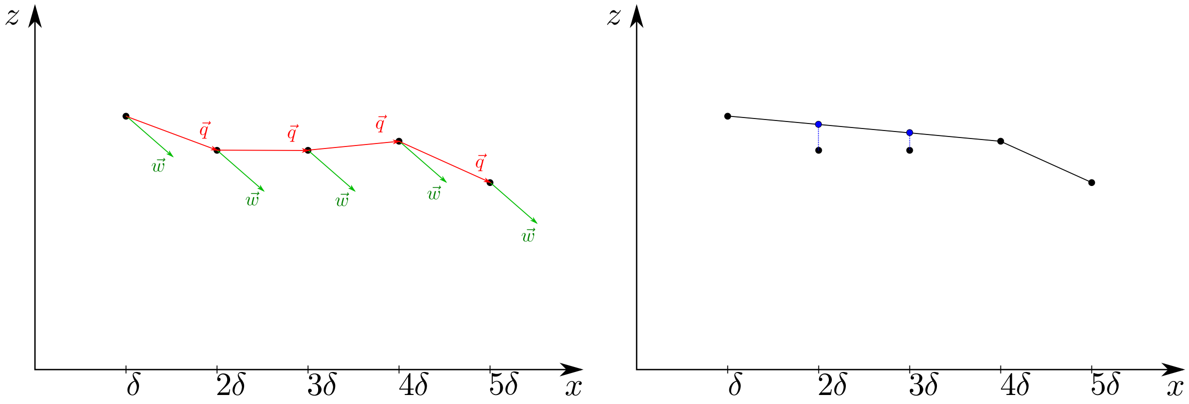
The last step is to go through each part of the maneuver and to rise all insignificant points to the new altitude; so, the maneuver respects the constraints of the vehicle and forms a continuous path between the significant points. The final convex envelope of one part of the maneuver is shown in Figure 4.5.

After all the maneuvers are checked and adjusted to be collision-free, it is necessary to update the distance matrix $\mathbf{D}$ because the length of the maneuvers may change and also the adjusted maneuvers are updated in the maneuver matrix $\mathbf{M}$.

4.4 Segment prolongation

Because maneuvers can be modified to be collision-free, and thus can be at different altitude, it may happen that the maneuvers and segments do not represent a continuous trajectory. It is, therefore, necessary to make the trajectory continuous by adjusting the covering segment and start ascending along the collision-free maneuver earlier than at the original segment endpoint to get the required altitude above the obstacle that is around the area. Reciprocally, a similar adjustment can be needed for descending part from a relatively high altitude.

If such a situation occurs, it means that the particular segment will not be completely covered, because the vehicle needs to gain the safe altitude above the obstacles. Because of maximization of the coverage, it may be more suitable to prefer a longer trajectory but with

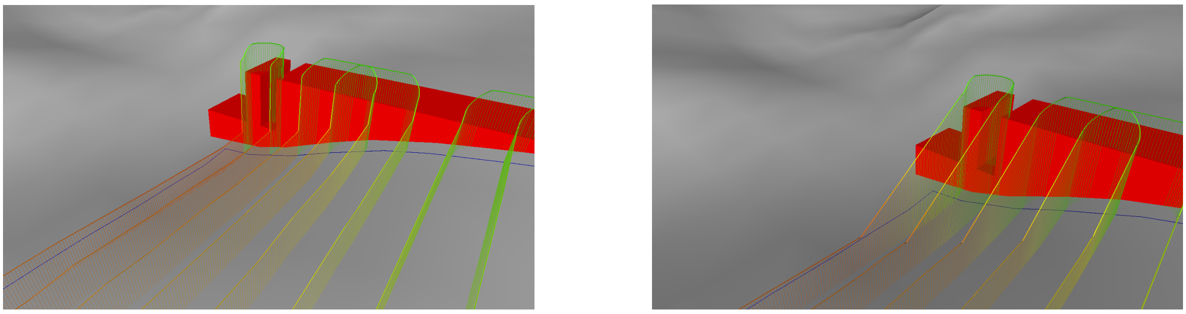


(a) Determination of significant points from the result of cross product between vectors $\vec{w}$ and $\vec{q}$.

(b) Raise of the insignificant points to correct altitude.

Figure 4.5: Example of determination of convex envelope of the maneuver in $x - z$ plane.

more coverage of the original covering segment. Therefore we define the *penalization constant* k to reflect how much the trajectory has to be prolonged that can be added to the length of the maneuver and distance matrix, and thus rather prefer connections of the segments that do not need a reduction of the covering segment; An example of the obstacle-free maneuver prolongation is depicted in Figure 4.6.



(a) Unchanged segments

(b) Changed segments

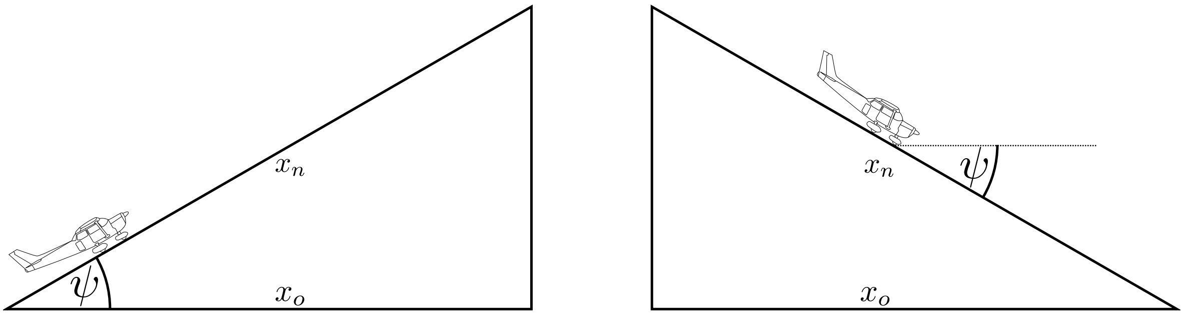
Figure 4.6: Situation where the segment has to be edited because the maneuvers were changed to avoid the obstacles with the respect to the fixed-wing vehicle constraints.

The penalization of the prolonged maneuver to respect reduction of the coverage is expressed as

$$p = k\left(x_n - x_o\right), \tag{4.7}$$

where p is the penalization, k is the penalization constant, and x_n and x_o are the new and original lengths of covering segment, respectively. Situation is displayed in Figure 4.7.

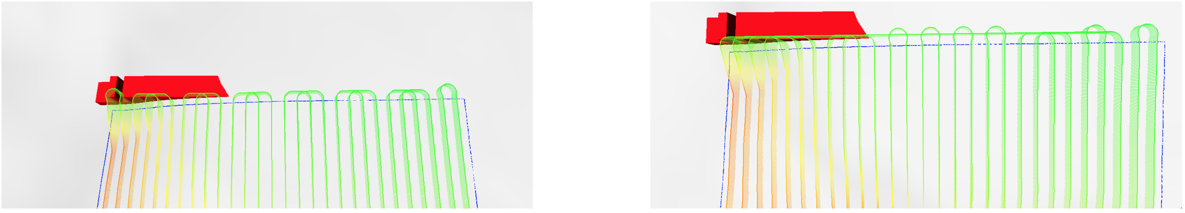
An example how the penalization parameter k influence the penalization and the final found solution is depicted in Figure 4.8.



(a) Segment prolongation because of neccesary ascending.

(b) Segment prolongation because of neccesary descending.

Figure 4.7: Segment prolongation.



(a) Penalization constant $k = 1$.

(b) Penalization constant $k = 50$.

Figure 4.8: Influence of the penalization constant k on the selected cover trajectory.

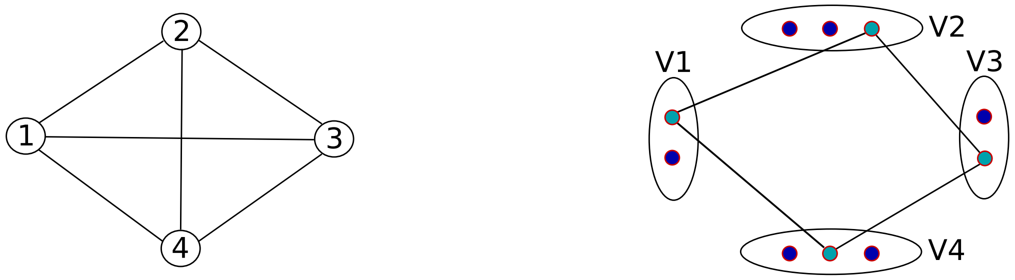
4.5 Traveling salesman problem

Once the *dominant direction* is determined and all maneuvers are created and adjusted to be free of collisions with obstacles, the problem is to determine the sequence of visits to the covering segments together with the selection of the particular maneuvers connecting the segments. The problem is considered as the *Traveling salesman problem* (TSP) which can be formulated as follows.

Given a "Cost matrix" of the dimension $n \times n$, $C = (c_{ij})$, where c_{ij} represents the cost of getting from the point i to the point j, find a permutation $(\sigma_1, \sigma_2, \cdots, \sigma_n)$ that minimazes the quantity $c_{\sigma_1\sigma_2} + c_{\sigma_2\sigma_3} + \cdots + c_{\sigma_n\sigma_1}$.

However, in the addresed area coverage problem, it is required to plan the path visiting the covering segments that can be approached by two opposite orientations. Therefore, the problem is not to visit all the given targets as in the TSP, but to select the orientation for each pair of collinear covering segments, and the problem becomes the *Generalized Traveling Salesman Problem* (GTSP). The GTSP can be formulated as follows. Having a given set of clusters, each cluster with at least one target. The problem is to determine the shortest tour visiting all clusters such that only one target is visited from each cluster. In the area coverage problem, the targets are equal to the pairs of the collinear covering segments, and thus, the problem is to choose the particular direction how the covering segment is approached. An example of instances of the TSP and GTSP is shown in Figure 4.9.

The GTSP created as a part of the coverage area problem is transformed to the Asymmetric Traveling Salesman Problem (ATSP) that is solved by the Helsgaun's implementation [28] of the Lin-Kerninghan heuristic.



(a) TSP.

(b) GTSP.

Figure 4.9: Example of TSP and GTSP.

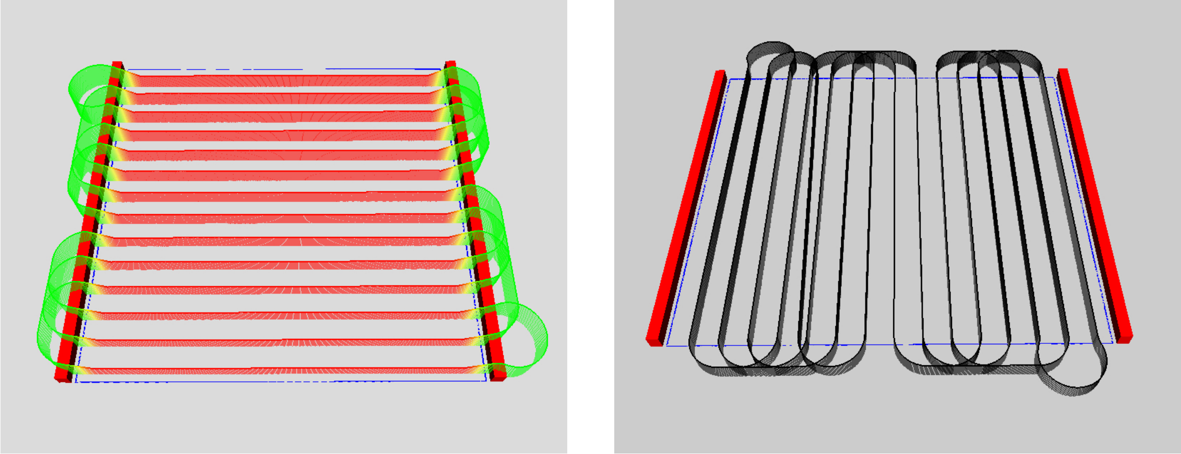
4.6 Improving Solution Quality by Considering Alternative Coverage Directions

The proposed solution of the addressed area coverage problem is based on the idea of the dominant orientation proposed by [7] to determine the shortest trajectory that provides complete coverage of the given area. However, in the herein addressed problem, we have to deal with the surrounding obstacles, and therefore, the maneuvers connecting the covering segments can be adjusted to different altitude to avoid collisions with the obstacles (see Section 4.3). Then, it might be necessary to shorten the covering segment to reach the requested altitude because of the limited climb/dive angle of the considered Dubins Airplane model. Therefore, it may happen that the final trajectory does not provide full coverage of the requested area to be covered, see Figure 4.10. Since full coverage is preferred in our solution over the length of the covering trajectory, we propose to consider also another orientation how to approach the covering using the pattern consisting of parallel segments. Since evaluation of the full range of all possible orientations would be computationally too demanding, based on the idea of [7] and the convex shaped polygon, we propose to evaluate all orientation defined by the sides of the convex polygon.

An example of improved coverage by considering different than dominant direction is shown in Figure 4.10, where the area to be covered is 500 m × 550 m large. Because of obstacles on two opposite sides in the dominant orientation, it is necessary to prolongate the maneuvers and adjust the ascending/descending of the vehicle. Even though the sum of the prolongations yields the covering trajectory with the length shorter than the trajectory for perpendicular coverage, only 94.6% of the are is covered in comparison to 100% coverage for a bit longer trajectory for the coverage in the direction perpendicular to the dominant direction.

4.7 Implementation

The proposed method is implemented in C++ with the use of multiple libraries developed by AI Center at CTU FEE. The implementation leverages on opendubins library and extends it for the purpose of the area coverage problem. Results are displayed in 3D Gui built on openGL library. The computational time of a maneuver is linearly dependent on its length. Set of all maneuvers is transformed into GTSP and solved by a method with complexity $\mathcal{O}(n^2)$.



(a) Maneuvers had to be prolonged and area is not fully covered.

(b) Area is fully covered but the covering trajectory is longer.

Figure 4.10: Example of shorter trajectory for the dominant orientation that is shorted but do not provide full coverage while using different orientation provides a bit longer trajectory but with full coverage.

CHAPTER 5

Results

This Chapter studies different aspects of the proposed method and its ability to cover the given area. Four scenarios have been designed to evaluate properties of the proposed method and compare it with an existing approach. For all scenarios, one airplane model and one flight altitude have been utilized. The used airplane model is inspired by real fixed-wing UAV used at the AI Center of FEE CTU, see Figure 5.1. Its constraints have been measured during real deployment. After several flights, parameters of the model have been adjusted to ensure that flight and trajectory produced by the proposed method is feasible for the fixed-wing UAV and the onboard controller can handle the trajectory with sufficient precision. The identified airplane model constraints are summarized in Table 5.1.

Table 5.1: Airplane model

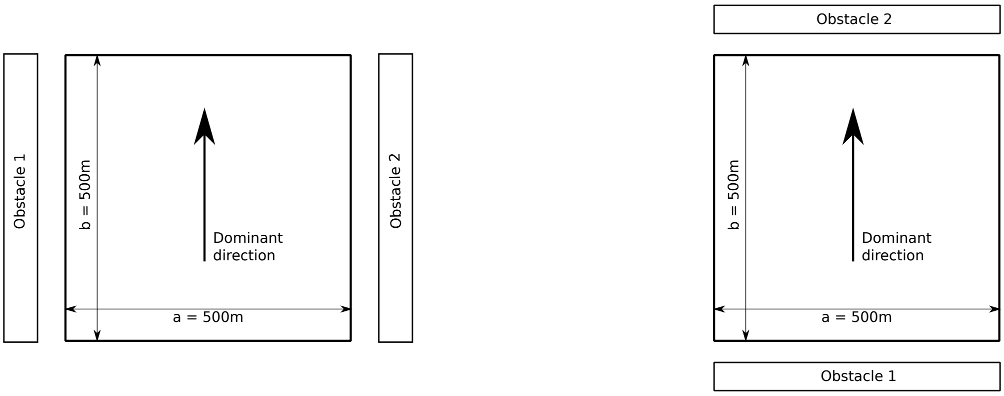
Altitude	20 m
Turn radius	40 m
Climb angle	30°
Dive angle	30°
Cover angle	35°

Each group of scenarios is focused on different aspect of the proposed method. The first group is focused on the influence of the shape of the area to the length of the trajectory and the results are reported in Section 5.1. The second group studies the influence of the size of the minimal turn radius on the length of the final trajectory and the results are reported in Section 5.2. The third group describes the influence of the height of the obstacles surrounding the area to the amount of the covered area. In Section 5.3, we show how the amount of covered area changes with increasing height of obstacles nearby its borders. The last group is focused on the influence of the obstacles on coverage and dominant direction. This group of scenarios shows, how the selection of the dominant direction can be influenced to maximize the covered area under the particular circumstances. The results are reported in Section 5.4. In Section 5.5, we report on the comparison of the proposed method with the existing algorithm. The chapter is concluded by several examples of the final cover trajectories found by the proposed approach in various areas. The solutions are shown in Section 5.6

Figure 5.1: Considered fixed-wing UAV of AI Center FEE CTU.

5.1 Influence of the shape of area on trajectory

The first group of scenarios studies the influence of the shape of the polygon on the length of the found cover trajectory. The length of the trajectory is mainly influenced by the number of turns which the UAV has to perform to cover the given area. Therefore we propose to study the influence of a sequence of problem instances with the area of the fixed volume. The area is of a rectangular shape defined by the lengths of its sides denoted a and b. The particular instances are created as follows. The length of the side a is decreased and the length of the side b is increased to maintain the same volume of the area. All computations of these scenarios have been performed in two variants.



(a) Obstacles are placed in non dominant direction aside of area.

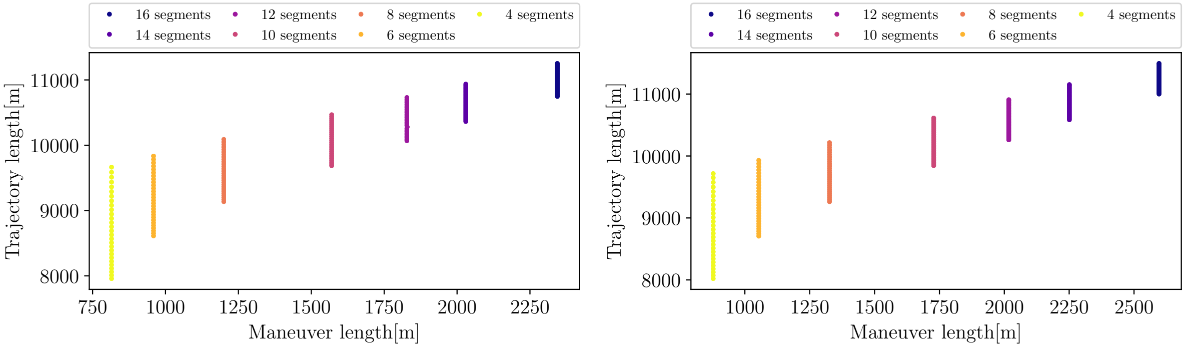
(b) Obstacles block dominant direction aside of area.

Figure 5.2: Basic test area setups.

The first variant contained obstacles in non-dominant direction only, and therefore, the final cover trajectory does not interfere with it and it is not influenced by the obstacles. In the second variant, two obstacles are situated in the dominant direction below and above the area, so, the final trajectory is adjusted to avoid them. The obstacles have the same length as the corresponding sides of the area. In a real scenario, these obstacles can, for example, represent a forests or buildings surrounding the area to be covered. The basic setup of the areas is depicted in Figure 5.2:

The initial area contains 17 covering segments, and therefore, 17 connecting maneuvers are needed to fully cover the area. This initial area is then alternated 400 times and the final setup needs just 3 covering segments, and therefore, 3 maneuvers to be covered.

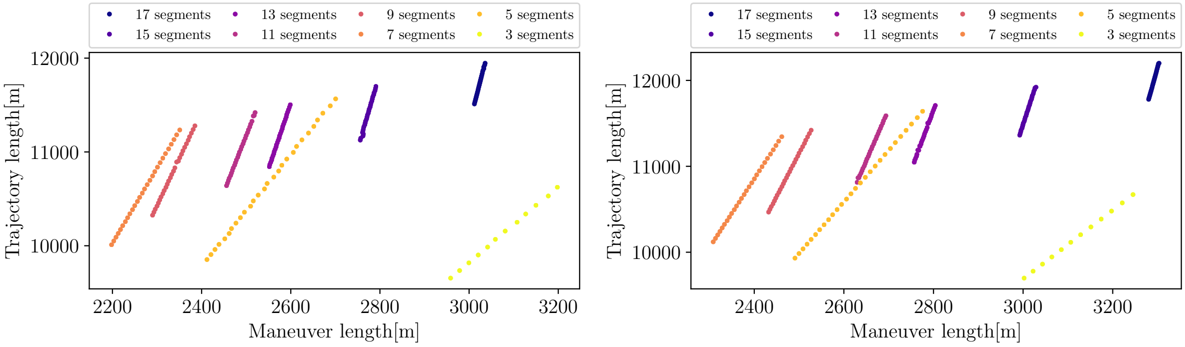
The influence of the shape of an area is depicted in Figure 5.3 and Figure 5.4. Resulting data are split into two groups. The first group represents those cases when the area needed an even number of segments to be covered, see Figure 5.3. In contrast, the second group is composed only of those cases when the odd number of segments is needed to compose the final coverage trajectory, Figure 5.4.



(a) Area without obstacles in dominant direction. **(b)** Area with obstacles in the dominant direction.

Figure 5.3: Influence of the shape of the area on trajectory length of the areas, where the even number of segments was required to cover the area.

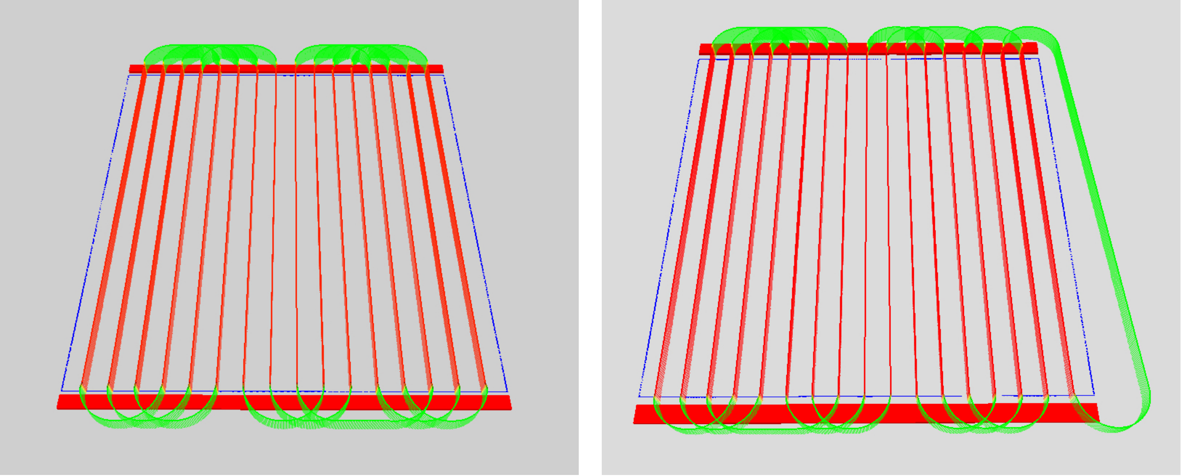
As can be seen in Figure 5.3, in cases where an even number of segments is needed, the shape of an area does not significantly influence the length of connecting maneuvers. The length of segments is influenced only and it does not influence the total length of the trajectory.



(a) Area without obstacles in dominant direction. **(b)** Area with obstacles in dominant direction.

Figure 5.4: Influence of the shape of the area on trajectory length of the areas, where the odd number of segments was required to cover the area.

As can be seen in Figure 5.4, when an odd number of segments is required, the shape of the area influences the length of the maneuvers as well. Since the odd number of segments is needed, the same odd number of connecting maneuvers are needed, one maneuver has to connect opposite sides of the area to create *eulearian path*. This maneuver crosses the area and its length thus depends on the shape of the area. Examples of the representative solutions showing these two cases are shown in Figure 5.5.



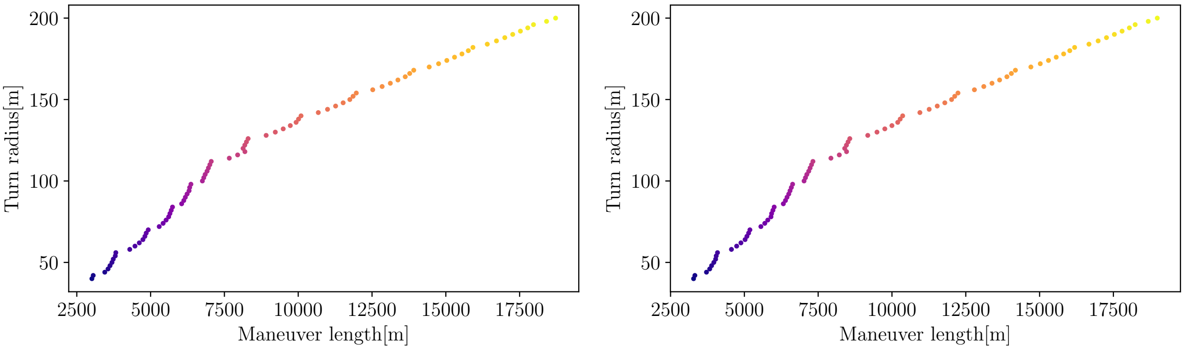
(a) The case where the even number of maneuvers is required.

(b) The case where the odd number of maneuvers is required.

Figure 5.5: The representative solutions of the covered areas where the even and the odd number of segments is required to fully cover the area.

5.2 Influence of the Minimal Turning Radius on Trajectory

The next group of scenarios demonstrates the influence of the minimum turning radius of the UAV on the length of the trajectory. In these scenarios, the area stayed the same and its shape is unchanged. The setup of the area is the same as in the initial setup of the 1st group, i.e., the rectangular area of the dimensions 500 m × 500 m (squared shaped). In this group, the obstacles are placed in the dominant direction and in the non-dominant direction, aside from the area. The minimal turning radius ρ is incremented from 20 m by one meter up to 100 m.



(a) Area with no obstacles in the dominant direction.

(b) Area blocked by the obstacles in the dominant direction.

Figure 5.6: Influence of the minimum turning radius on the length of cover trajectory.

The influence of the minimum turning radius ρ is displayed in Figure 5.6. The trajectory is prolonged with increasing ρ. The length of the segments is constant and increase of the trajectory length is caused only by the prolongation of connecting maneuvers. In some cases, the maneuver length is increased more significantly than in the previous cases. It is caused by the maneuvers from the previous solution cannot connect the same segments as before because for the increased radius ρ, it is not possible to perform “sharp” enough turn as before and a new solution is therefore created and selected.

5.3 Influence of the Obstacle Height on Coverage of an Area

In the third group of scenarios, the influence of the obstacle height on the total amount of the covered area is studied. A part of the area is considered to be covered only if the UAV is at a specific height above the ground. If it is higher, then the area is considered not be covered, therefore we can expect that for increased height of the obstacles, the connecting maneuvers have to be prolongated, and thus the covering segment at the desired altitude is shortened. Note that when the area is surrounded by obstacles completely from all sides, it might not be possible to completely cover the area by the fixed-wing UAV. We consider a slightly different setup of the area and obstacles to always have an odd number of segments without the Eulerian path that may significantly influence the length of the find cover trajectory. Besides, the obstacles are placed 5 m from the sides of the area and they completely block the dominant direction of the area. The area constraints and parameters of the obstacles are depicted in Table 5.2, which referes to the area configuration as in Figure 5.2b:

Table 5.2: Scenario Parameters for Increasing Obstacle Height

Side a	475 m
Side b	500 m
Total size	0,237 km^2
Obstacle 1 height	$\langle 0, 400 \rangle$ m
Obstacle 2 height	$\langle 0, 400 \rangle$ m

The influence of the height of the obstacles is shown in Figure 5.7. For a low height of the obstacles, the area is completely covered, but once the height of the obstacle exceeds 5 meters, the coverage of the area starts to decrease linearly to zero for the about 334 m height obstacles. If an area would be completely surrounded by the obstacles from all sides, the proposed method does not provide the cover path with full coverage. However, since it is requested to cover the area from the particular altitude, such a solution does not exist.

5.4 Influence of the Penalization on the Total Coverage and Dominant Direction

In some cases, it is not possible to completely cover the given area in the dominant direction, because of obstacles surrounding the area. Such a final trajectory is still the shortest possible, but from the point of view of the amount of covered area, it is not optimal. In such a scenario, a different direction can provide higher coverage at the cost of the longer cover path. The proposed method can find this solution if penalization constant is raised enough.

Two different areas are considered in studying the influence of the penalization parameter. The dimensions of the first area are selected such that the area can be covered by an even

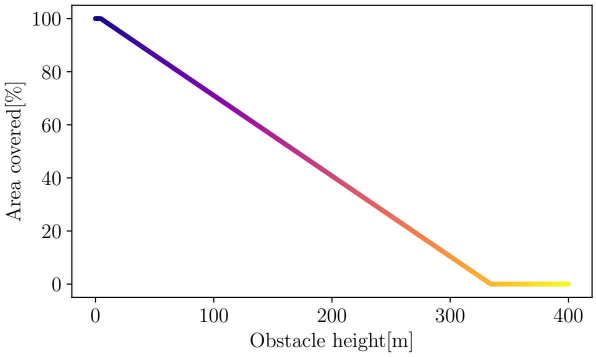


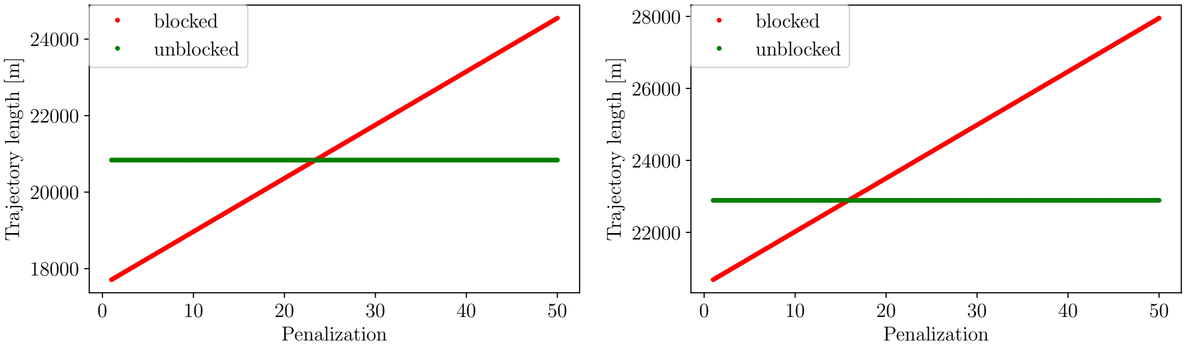
Figure 5.7: Influence of the obstacle height on the coverage of the area.

number of the segments in the dominant direction. The dimensions of the second area are selected such that the area can be fully covered by an odd number of segments in the dominant direction. Both areas have two times longer dimension in the non-dominant direction and both have obstacles blocking the dominant direction. Parameters of the areas are summarized in Table 5.3.

Table 5.3: Penalization areas setups

Area 1 (Even)	
Side a	475 m
Side b	950 m
Total size	0,45125 km^2
Obstacle 1 height	20 m
Obstacle 2 height	20 m
Area 2 (Odd)	
Side a	500 m
Side b	1000 m
Total size	0.5 km^2
Obstacle 1 height	20 m
Obstacle 2 height	20 m

The results on the trajectory length for particular areas and increasing penalization parameter are depicted in Figure 5.8. As it can be seen, the penalization for prolonged (and thus not completely covered) segments are added to the length of the final trajectory and at some point, the trajectory in the dominant direction becomes longer than a trajectory in a non-dominant direction, which stays the same because it is not influenced by the obstacles. Hence a solution with the longer path but with full coverage is found. In these scenarios, the total amount of the covered area is 0.975%, if the area is being covered in dominant direction. If the non-dominant direction is selected, the area is covered completely.



(a) Area covered by the even number of segments. **(b)** Area covered by the odd number of segments.

Figure 5.8: Influence of the penalization on trajectory length and dominant direction.

5.5 Comparison with Reference Solution

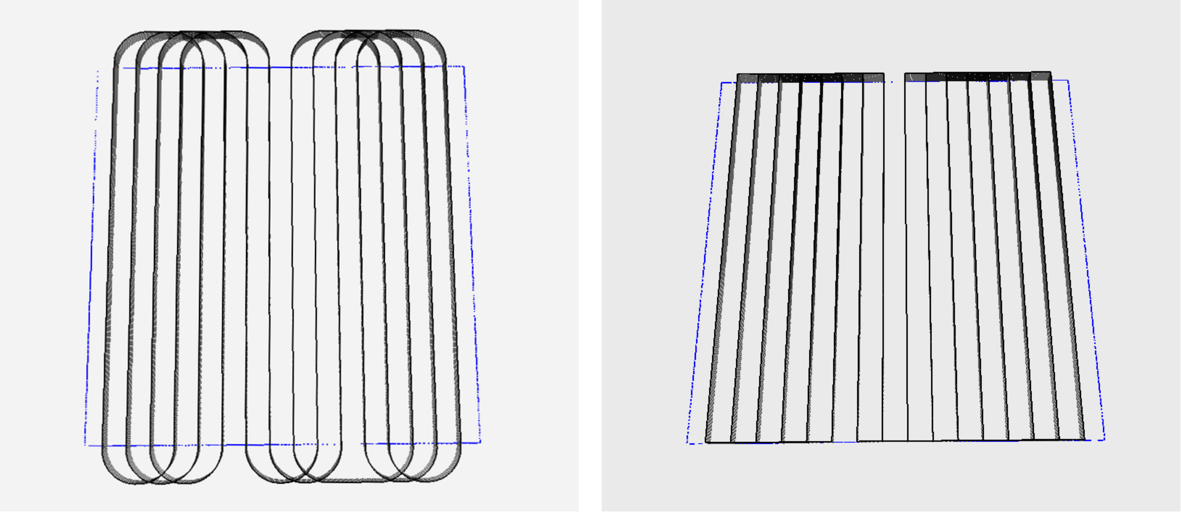
In [8] is proposed the method which solves the area coverage problem of an area surrounded by the obstacles being covered by a UAV. Therefore the proposed method is compared with the approach [8] described in Chapter 2 that is further denoted as reference algorithm. Two scenarios were designed to show differences between the methods. The first scenario does not have the surrounding obstacles, while obstacles around all sides of the area are in the second scenario. Because the algorithm proposed by [8] does not avoid obstacles, an area surrounded by obstacles has to be reduced to avoid collisions of the UAV with the obstacles. It is because the Boustrophedon decomposition is used to determine the cells of the area and cells which contain an obstacle are not covered but are also not being actively avoided. Because of that, the reduction of the area to be covered is made such that the maximum of the area is covered while the fixed-wing UAV stays in the desired safe distance from the obstacles surrounding the area. The reduction is equal to the minimum turning radius of the fixed-wing UAV in each direction of each segment.

The area has the same size as very first setup shown in Figure 5.2, but the surrounding obstacles are around all sides and they are five meters far from the edge of the area and their height is 20 m. The reduction of the area is made only for the case of the surrounding obstacles and only for the reference algorithm. Each scenario is considered in 400 variants for decreasing the length of the side a about one meter while the length of the side b is increased such that the volume of the area is the same, i.e., we follow the first group of evaluation scenarios reported in Section 5.1.

Examples of found solutions without obstacles and with the surrounding obstacles are depicted in Figure 5.9 and Figure 5.10, respectively, for both evaluated method (the proposed and reference).

The area surrounded by the obstacles is explicitly reduced for the reference algorithm [8]. On the other hand, the proposed algorithm reduced the coverage by prolongation of the connecting maneuvers at the cost of shortening the fly at desired altitude along the covering segment. The achieved coverage in the percentation of the complete coverage by the proposed and reference algorithms is depicted in Figure 5.11.

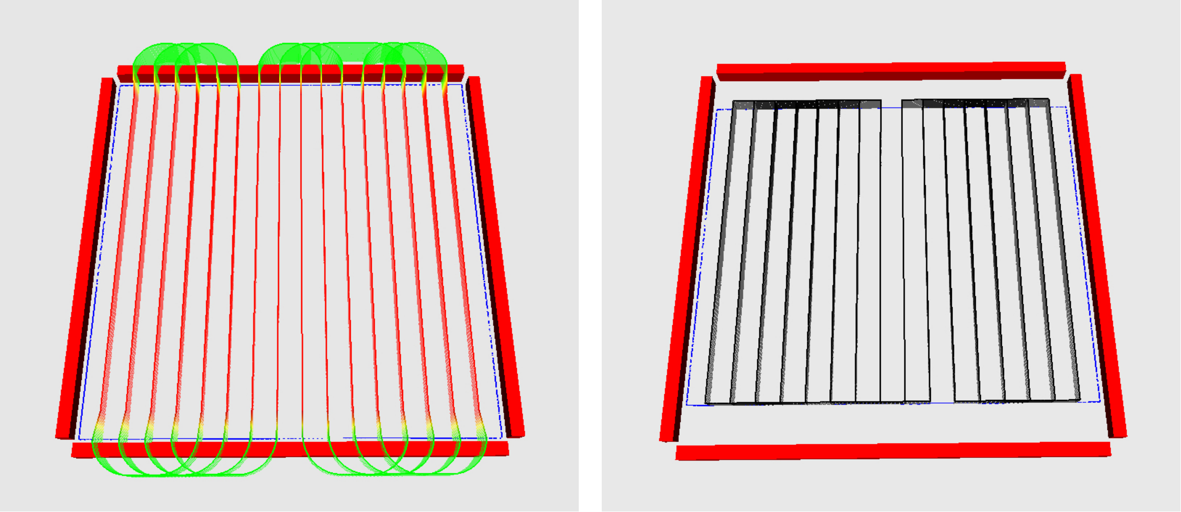
In case of presence of the obstacles, the area was reduced for algorithm proposed by [8]. Following graph shows the results of the total amount of covered area by both algorithms. The x-axis contains the ratio of the length of the side a to the side b.



(a) Proposed method.

(b) Reference method

Figure 5.9: Final paths in case of no obstacles surrounding the area.



(a) Proposed method.

(b) Reference method

Figure 5.10: Final paths in case of the area surrounded by obstacles.

As it can be seen in Figure 5.11, the proposed method covers more amount of area in all variants of the testing scenario than the reference method proposed by [8]. The proposed method has several advantages over the reference method [8]. The main advantage is that the proposed method is fully 3D offline planning method solving the given area coverage problem with the presence of obstacles surrounding the area. The proposed method plans the cover trajectory that satisfies the motion constraints of the Dubins airplane. The method proposed by [8] does not take in consideration the motion constraints and the produced cover trajectory contains "sharp" connections of the covering segments. These connections cannot be precisely followed by fixed-wing UAV and real trajectory differs from planned one. Also, the final trajectory is strongly dependent on the utilized path (trajectory) following controller. The proposed method directly produces the desired cover path that is feasible for the given

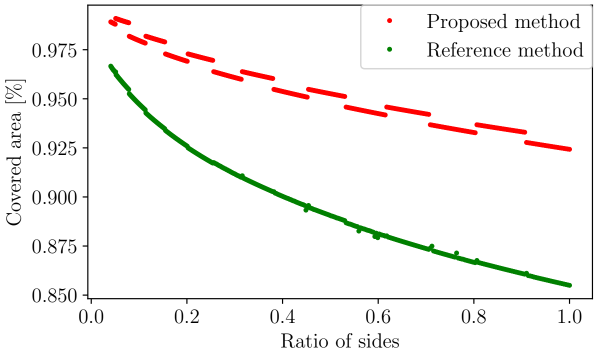


Figure 5.11: Comparison of the amount of covered area by both algorithms.

fixed-wing UAV and thus it is less dependent on the controller. Besides, the proposed method provides higher coverage of the area in comparison to the required reduction for the reference method [8].

5.6 Selected Solutions of the Area Coverage Problem

Several solutions found by the proposed method are presented in this section to further support advantages of the method and ability to solve different instances of the area coverage problem. First two instances are covered in the dominant direction that is selected manually because of showing the influence of surrounding obstacles to the solution. The last instance is intentionally covered in the non-dominant direction. Thus, all areas are considered without and with the (at least two) obstacles.

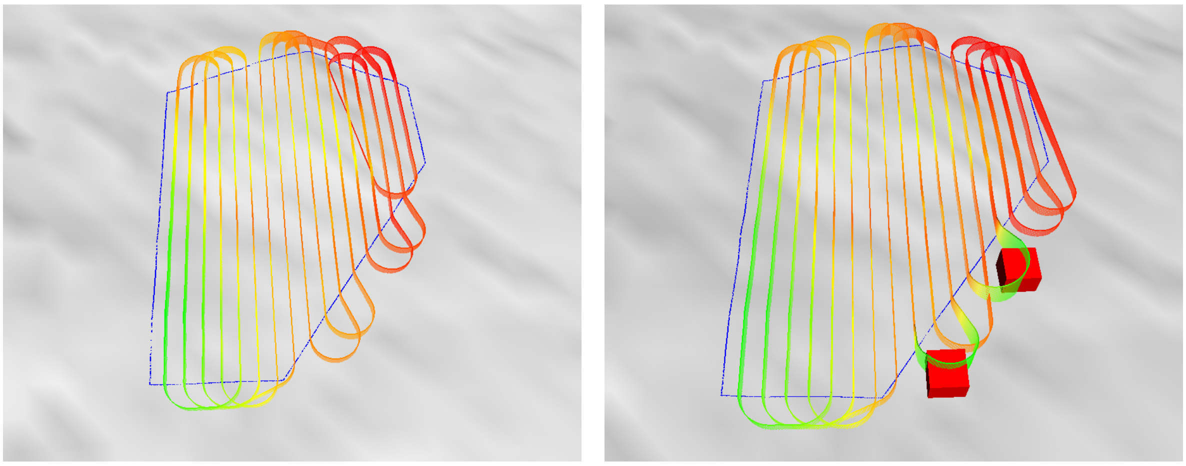


Figure 5.12: The first selected area to be covered.

The first instance is asymmetric area and found solutions are depicted in Figure 5.12, where it can be seen that various turns maneuvers are created. Once the obstacles are presented, basically the same solution is determined with few adjustments. Length of covering trajectory is 12419.7 m in case of the area without obstacles and it is prolonged to 12458.7 m after the obstacles are introduced. Total coverage decreases from 100% to 99% if no penalization is introduced, and therefore, dominant direction stays the same.

The second area is symmetric but not rectangular and it is shown in Figure 5.13. It is also covered in the dominant direction but the resulting cover path is not symmetrical, because spacing from one side is longer than from the second one. It is caused by the fact that the width of the area is not equal to an integer multiple of the UAV cover radius r.

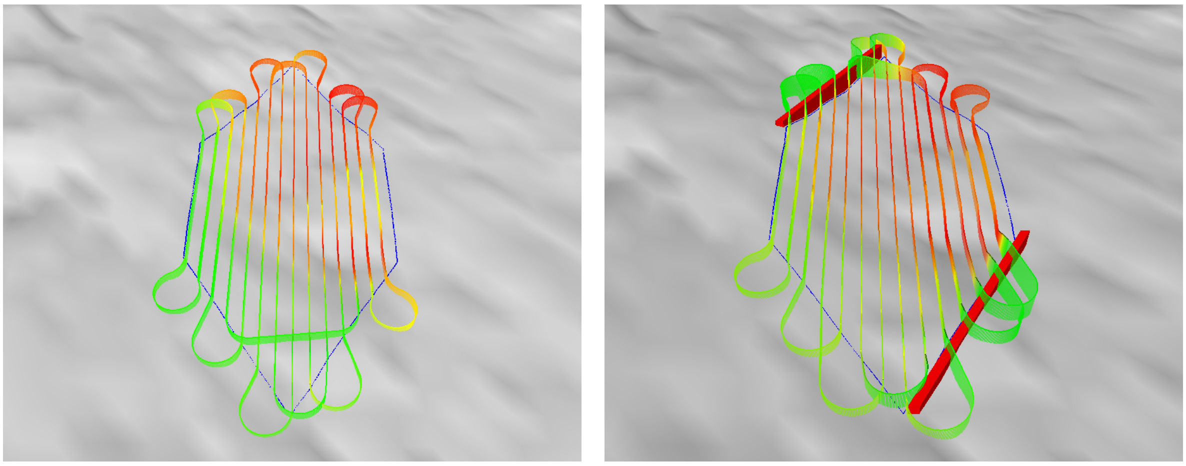


Figure 5.13: The second selected area to be covered.

In the second case, the length of final covering trajectory is 11487.7 m if no obstacles are surrounding the area and increases to 11589.8 m once the obstacles are introduced. The coverage is decreased from 100% to 95.6%.

The last presented scenario is depicted in Figure 5.14, where the dominant direction is selected to be parallel to the axis of the symmetry and not a direction of one of the sides of the convex polygonal area. It can be noticed that the obstacles do not represent a significant issue; however, some of the connecting maneuvers are of the CCC type with the zero length of the first C segment.

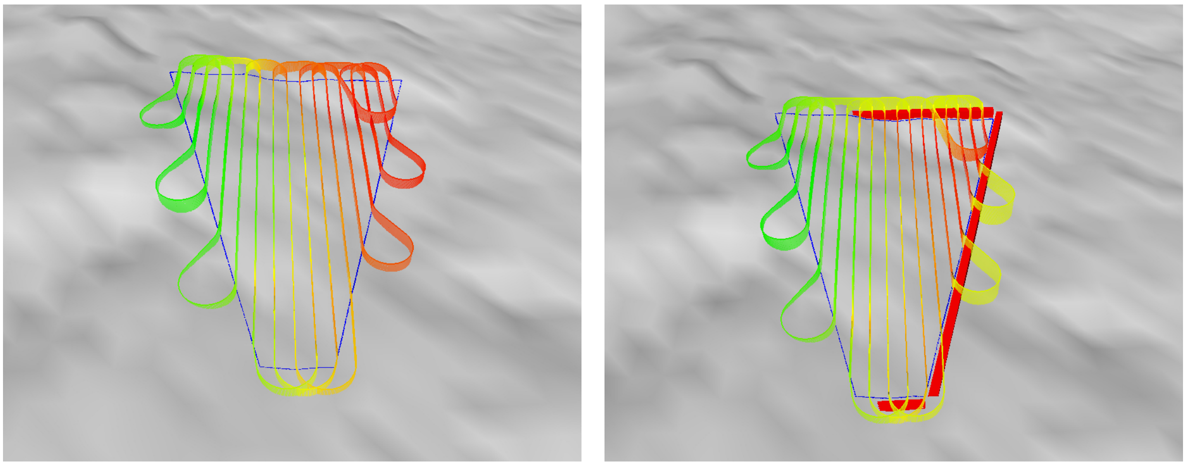


Figure 5.14: The third selected area to be covered.

The covering trajectory length is 10681.9 m and increases to 10758.5 m after the obstacles are introduced. In this case, the coverage decreases from 100% to 96.3%.

CHAPTER 6

Conclusion

In this thesis, we study the area coverage problem where a fixed-wing vehicle is requested to cover an area that can be surrounded by relatively tall obstacles. Based on the review of the state-the-art-methods in coverage planning, we consider it is sufficient to limit the case to cover the convex polygonal area without the obstacles, because the existing approaches utilized some form of decomposition to convex areas without obstacles, and thus the problem of surrounding obstacles still needs to be addressed. Moreover, the motion constraints of the vehicle have to be considered as well. However, the most of the existing methods are not well suitable for the motion constraints of the fixed-wing vehicles that need cover trajectory respecting the minimum turning radius of the vehicle. Therefore, we propose a novel solution to address all the required constraints.

The proposed method follows the idea of the existing coverage algorithms that prefer to use a regular coverage pattern that is applied in the dominant direction of the area to be covered. The preferred pattern is composed of parallel covering segments that are connected by turn maneuvers at the end and beginning of each segment. To find the best turn maneuvers connecting the segments that respect the fixed-wing vehicle motion constraints, the Dubins paths are used. Dubins paths are the shortest possible connections between two points in plane respecting Dubins vehicle model.

The proposed method first determines Dubins maneuvers between end-points of covering segments and then checks each maneuver for possible collisions with the obstacles. If a maneuver collides with an obstacle, it is moved to the safe height above the obstacle and polished to still respect the fixed-wing vehicle motion constraints. The change of colliding maneuvers can cause discontinuity of the final cover trajectory, and therefore, each such a segment that is connected to the changed maneuver has to be checked. In the case that it is disconnected from the consecutive maneuver of the determined covering trajectory, the segment has to be changed, so the fixed-wing vehicle starts to ascend before continuing to the maneuver or descend after leaving the maneuver to connect with the consecutive segment. Such changes of the final covering trajectory can decrease the coverage of the area.

The proposed method is focused on the minimalization of the length of the covering trajectory with respect to the motion constraints of the fixed-wing vehicle while any collisions with the obstacles are avoided. To solve the problem with the total coverage of the area, the changed final trajectory is penalized, so the covering trajectories with the less covered area are artificially prolonged, and therefore, a different solution which would be longer otherwise is preferred instead. The proposed method is capable to find the shortest cover trajectory of an area represented by the real height map of the real environment surrounded by the real obstacles if exists.

The proposed method has been evaluated on the data using the real-height maps in Section 5.6 and results of its comparison with the reference method are reported in Section 5.5. In a case that the surrounding obstacles are too tall, the proposed method might not be able to find the solution because the fixed-wing vehicle would not be able to avoid the obstacles surrounding the area. Therefore, we would like to extend the proposed method in the near future to allow it to find the solution of the addressed area coverage problem in case of the tall obstacles surrounding the area. Also, we consider determining a different solution to the covering maneuvers instead of the utilized Dubins path which would be still respecting the motion constraints.

Bibliography

[1] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, “Complete coverage navigation of cleaning robots using triangular-cell-based map,” *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, pp. 718–726, 2004.

[2] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, “Planning paths of complete coverage of an unstructured environment by a mobile robot,” in *International Conference on Advanced Robotics*, vol. 13, pp. 533–538, 1993.

[3] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[4] H. Choset, “Coverage of known spaces: The boustrophedon cellular decomposition,” *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[5] S. C. Wong and B. A. MacDonald, “A topological coverage algorithm for mobile robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1685–1690, 2003.

[6] H. I. Perez-Imaz, P. A. Rezeck, D. G. Macharet, and M. F. Campos, “Multi-robot 3d coverage path planning for first responders teams,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1374–1379, 2016.

[7] I. Maza and A. Ollero, “Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms,” in *Distributed Autonomous Robotic Systems 6*, pp. 221–230, Springer, 2007.

[8] A. Xu, C. Viriyasuthee, and I. Rekleitis, “Efficient complete coverage of a known arbitrary environment with applications to aerial operations,” *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.

[9] M. Owen, R. W. Beard, and T. W. McLain, “Implementing dubins airplane paths on fixed-wing uavs,” in *Handbook of Unmanned Aerial Vehicles*, pp. 1677–1701, Springer, 2015.

[10] Z. L. Cao, Y. Huang, and E. L. Hall, “Region filling operations with random obstacle avoidance for mobile robots,” *Journal of Field Robotics*, vol. 5, no. 2, pp. 87–102, 1988.

[11] C. Luo and S. X. Yang, “A real-time cooperative sweeping strategy for multiple cleaning robots,” in *IEEE International Symposium on Intelligent Control*, pp. 660–665, IEEE, 2002.

[12] D. W. Gage, “Randomized search strategies with imperfect sensors,” in *Mobile Robots VIII*, vol. 2058, pp. 270–280, International Society for Optics and Photonics, 1994.

[13] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, “Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage,” *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 107–120, 2009.

[14] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study.* Princeton university press, 2006.

[15] A. Xu, C. Viriyasuthee, and I. Rekleitis, “Optimal complete terrain coverage using an unmanned aerial vehicle,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2513–2519, 2011.

[16] H. Choset, “Coverage for robotics–a survey of recent results,” *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[17] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 116–121, IEEE, 1985.

[18] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[19] A. Zelinsky, “A mobile robot exploration algorithm,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 6, pp. 707–717, 1992.

[20] Y. Gabriely and E. Rimon, “Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 954–960, IEEE, 2002.

[21] S. X. Yang and C. Luo, “A neural network approach to complete coverage path planning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 718–724, 2004.

[22] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, “Morse decompositions for coverage tasks,” *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.

[23] S. Hert and V. Lumelsky, “Polygon area decomposition for multiple-robot workspace division,” *International Journal of Computational Geometry & Applications*, vol. 8, no. 04, pp. 437–466, 1998.

[24] H. Choset and P. Pignon, “Coverage path planning: The boustrophedon cellular decomposition,” in *Field and service robotics*, pp. 203–209, Springer, 1998.

[25] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *46th IEEE Conference on Decision and Control*, pp. 2379–2384, IEEE, 2007.

[26] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[27] C. E. Noon and J. C. Bean, "A lagrangian based approach for the asymmetric generalized traveling salesman problem," *Operations Research*, vol. 39, no. 4, pp. 623–632, 1991.

[28] K. Helsgaun, "An effective implementation of the lin–kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[29] "Algorithm implementation/geometry/convex hull/monotone chain." `https://en.wikibooks.org/wiki/Algorithm_Implementation/Geometry/Convex_hull/Monotone_chain` [Online] Accessed on: 22.5.2018.