ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Marek** Jméno: **Jakub** Osobní číslo: **406714**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Plánování trajektorií s omezenou křivostí v úloze pokrývání vzdušnými prostředky

Název diplomové práce anglicky:

Coverage path planning with curvature-constrained aerial vehicles

Pokyny pro vypracování:

6. Seznamte se plánovací úlohou pokrývání [1, 2, 3, 4, 5, 6] a variantami
se vzdušnými prostředky [7, 8].
7. Formulujte problém pokrývání oblastí jako dvoufázové řešení s detekcí
možných vysoce náročných oblastí a následným podrobným pokrytím
náročných oblastí.
8. Navrhněte formulaci druhé fáze řešení jako variantu úlohy obchodního
cestujícího s okolím pro Dubinsovo vozidlo [9].
9. Posoudit vliv distribuce prioritních oblastí na celkové náklady na
pokrytí, například spotřebovanou energii [10].
10. Navrhněte experimentální ověření vyvinutých řešení s reálným
vzdušným prostředkem typu křídlo.

Seznam doporučené literatury:

[1] Galceran, E. and Carreras, M.: A survey on coverage path planning for robotics, Robotics and Autonomous Systems, 61(12):1258-1276, 2013. doi: 10.1016/j.robot.2013.09.004
[2] Almadhoun, R., Taha, T., Seneviratne, L., and Zweiri, Y.: A survey on multi-robot coverage path planning for model reconstruction and mapping, SN Applied Sciences, 1(8):847, 2019. doi: 10.1007/s42452-019-0872-y
[3] Yu, X., Roppel, T., and Hung, J.: An optimization approach for planning robotic field coverage, IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, 2015, pp. 004032-004039. doi: 10.1109/IECON.2015.7392728
[4] Bochkarev, S. and Smith, S.: On minimizing turns in robot coverage path planning, 2016 IEEE International Conference on Automation Science and Engineering (CASE), 2016, pp. 1237-1242. doi: 10.1109/COASE.2016.7743548
[5] Schirmer, R., Biber, P., and Stachniss, C.: Coverage Path Planning in Belief Space, 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 7604-7610. doi: 10.1109/ICRA.2019.8793969
[6] Maza, I. and Ollero, A.: Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, Distributed Autonomous Robotic Systems 6, 2007, pp. 221-230.doi: 10.1007/978-4-431-35873-2_22
[7] Lewis, J., Edwards, W., Benson, K., Rekleitis, I., and O&#39;Kane, J.: Semi-boustrophedon coverage with a dubins vehicle, 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 5630-5637. doi: 10.1109/IROS.2017.8206451
[8] Karapetyan, N., Moulton, J., Lewis, J., Quattrini Li, A., O&#39;Kane, J., and Rekleitis, I.: Multi-robot Dubins Coverage with Autonomous Surface Vehicles, 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 2373-2379. doi: 10.1109/ICRA.2018.8460661
[9] Faigl, J., Váňa, P., Pěnička, R., and Saska, M.: Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles, Journal of Field Robotics, 36(1):270-301, 2019, doi:10.1002/rob.21823
[10] Yu, K., O’Kane, J., and Tokekar, P.: Coverage of an Environment Using Energy-

Constrained Unmanned Aerial Vehicles, 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3259-3265. doi: 10.1109/ICRA.2019.8794150.[

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Jan Faigl, Ph.D., centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **12.02.2020** Termín odevzdání diplomové práce: **14.08.2020**

Platnost zadání diplomové práce: **30.09.2021**

prof. Ing. Jan Faigl, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání	Podpis studenta

CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

Department of Computers

Master's thesis

Coverage path planning with curvature-constrained aerial vehicles

Jakub Marek

August 2020

Supervisor: prof. 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 14 August 2020

. . . .
Jakub Marek

Acknowledgement

I would like to thank to prof. Ing. Jan Faigl, Ph.D. for his energy and his patience during supervision of my master's thesis. Also, I would like to thank to Ing. Petr Váňa for consultations and the insight to the related work.

Abstract

This thesis is focused on the area coverage problem divided into two phases by the fixed-wing Unmanned Aerial Vehicle (UAV). The classical approach to the area coverage problem is to capture the whole area at once. Therefore, the coverage is performed at roughly the same altitude with approximately the same resolution. The two-phase approach proposed in this thesis allows us to perform a fast scan of the whole area and detect its regions that require further inspection. The second phase of the proposed approach involves the high-resolution coverage of multiple disconnected regions. This thesis proposes a method solving the multiple-region coverage with the use of the re-entries into the regions and compares this approach with the classical approach not involving the re-entries.

Key words: Multi-region coverage; area coverage problem; fixed-wing UAV coverage

Abstrakt

Tato práce se zabývá úlohou pokrývání daného území rozdělené na dvě fáze za použití Bezpilotního Létajícího Prostředku (UAV) s pevným křídlem. Klasickým přístupem k úloze pokrývání je zachytit celé území najednou. Pokrývání je tudíž provedeno po celém území v přibližně stejné výšce se stejným rozlišením. Dvoufázový přístup navržený v této práci umožňuje provést rychlé pokrytí celého území a souběžnou detekci regionů, které vyžadují detailnější pokrytí. Druhá fáze navrhovaného přístupu zahrnuje pokrytí několika oddělených regionů s vyšším rozlišením. Tato práce navrhuje metodu řešící problém pokrývání více oddělených regionů s využitím opakovaných přeletů a porovnává výsledky s klasickým přístupem, který opakované přelety neumožňuje.

Klíčová slova: Pokrývání více regionů; úloha pokrývání; Pokrývání s UAV s pevným křídlem

Contents

List of Figures

List of Tables

CHAPTER 1

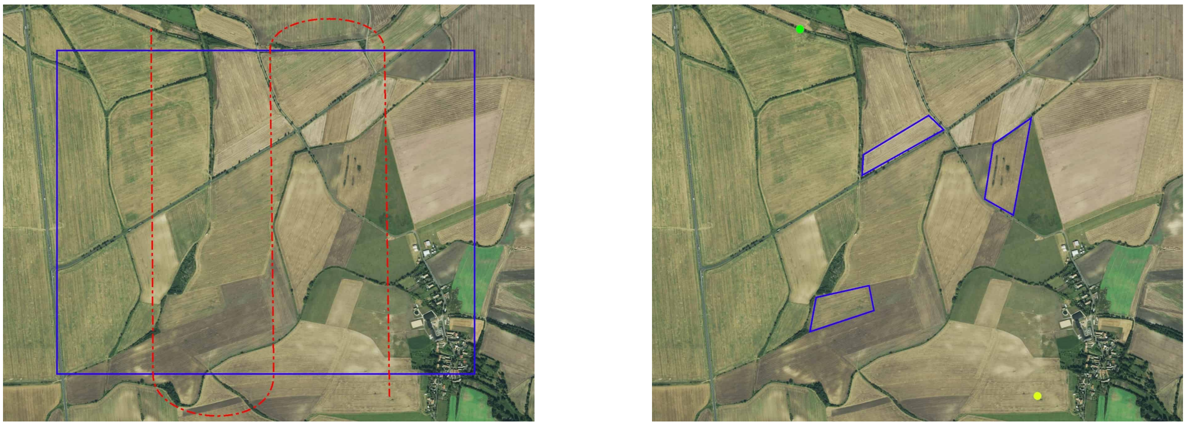
Introduction

This work is focused on the problem of area coverage with fixed-wing *Unmanned Aerial Vehicles* (UAVs). The field of UAVs experienced rapid growth in recent years and opened new horizons and commercial opportunities. The UAVs became accessible and nowadays can be encountered in various tasks like the surveillance [2], agriculture [3], environment monitoring [4], border control, property monitoring, infrastructure monitoring, disaster management [5], medical supplies transportation [6], goods transportation, de-mining of the previous war zones [7], and more [6], [8], [9]. UAVs' usage brings multiple advantages like increased safety because no pilots are on-board, decreased cost, and possible future automation of the task. Also, an improved precision can be achieved, because a UAV can fly in a low altitude such that no human pilot would dare to do so. One of the less-discussed benefits is that in some scenarios, less CO_2 emissions are produced by UAVs [10]. Similarly to the classical airplanes, the operation of UAVs in commercial applications is limited by the official regulations [6] and it is also needed to take into account limited operational time and payload of the nowadays UAVs. There are existing approaches that are focused on the extension of the operational time of the UAVs, for example, through the cooperation with the unmanned ground stations (UGV) [11], [12] or by installing super-tin photovoltaic panels into the wings of the UAVs [13].

The area coverage problem is to capture the area by the attached sensor, e.g., the camera, most efficiently w.r.t. the given criteria, such as to find the shortest/fastest path [14], [8]. In addition to approaches that cover the area with roughly the same details, practical applications where the same level of detail is not required can be found in precision agriculture. We can argue that not all the parts of the area have to be affected by abnormal conditions, for example, by a drought or a disease affecting the plants. Therefore, it is not resourceful and adequate to cover the whole area in great detail. Hence, the area coverage problem can be divided into two phases further referred to as

- The scanning phase,
- The collection phase.

The scanning phase, visualized in Figure 1.1a, is complete coverage of the given area from the relatively high altitude to detect regions of interest that need further investigation. In the collection phase, the UAV is requested to visit all the regions detected in the scanning phase and cover them in greater detail, see example in Figure 1.1b. The idea of two-phase area coverage is to quickly identify possible regions of interest and then collect detailed data to minimize the coverage cost. However, the collection phase can be utilized in a standalone scenario, e.g., for the crop spraying of multiple fields with Z-37, depicted in Figure 1.2a.



(a) A depiction of the scanning phase of the area coverage with a draft of possible coverage trajectory (red) to detect regions of interest.

(b) The collection phase of the area coverage with three regions of interest (blue), the starting (yellow) and ending (green) locations.

Figure 1.1: The examples of the two-phase area coverage problem.

The herein proposed approach to the area coverage problem is feasible for both the fixed-wing UAVs and airplanes. Both of the vehicles share similar motion constraints and can be modeled as Dubins vehicle [15]. Although generalization of the Dubins vehicle model to the three-dimensions space exists, the addressed problem can be transferred into two dimensions only. We argue that it is a realistic simplification because there are existing terrain-following methods [16, 17], and many agricultural areas are roughly flat. Therefore it is possible to keep the same altitude above them with a reasonable bias.

(a) Z-37 - an airplane used in the agriculture.
Credits: letadlanaplatne.cz[1]

(b) UAV of AIC of FEE CTU.
Credits: Jakub Sláma

Figure 1.2: Examples of UAVs.

The division of the coverage problem into the scanning phase and the collection phase

[1] `https://www.letadlanaplatne.cz/wp-content/uploads/2018/04/Zl%C3%ADn-Z-37-Čmelák.jpg`

presented in this thesis is based on the formulation of the collection phase as the multi-region area coverage problem. Here, it is worth noting that the scanning phase containing the single region is a single area coverage problem that can be considered as well-addressed by existing approaches overviewed in Chapter 2.2. The proposed method treats the multi-region coverage problem as the *Generalized Traveling Salesman Problem* (GTSP). The proposed formulation as the GTSP allows multiple visits and re-entries to the regions and supports the examination of the relationship between the distance of the regions and selected orientation of the individual regions' coverage patterns on the effectiveness of the re-entries. Notice that enabling re-entries to the regions brings increased complexity of the problem because the coverage path over all the regions has to be considered at once without the possibility to decompose the problem to the determination of the tour over the regions and then separate the determination of the coverage of the individual regions. Despite the increased complexity of the problem, we consider such a formulation advantageous because the coverage paths with the re-entries are shorter than the coverage paths without them, shown in the reported results. In addition to the empirical evaluation, we also propose the deployment of the developed method on the testing fixed-wing platform depicted in Figure 1.2b, to further validate the feasibility of the solutions in experimental scenarios. The presented work is motivated by the deployment of an aerial vehicle in area coverage scenarios, where also robotic vehicles such as autonomous UAVs can be deployed. Therefore, the words vehicle and robot are used as synonyms in the rest of this thesis.

The rest of the thesis is structured as follows. The existing approaches to the single area coverage problem and multi-region coverage problem are presented in Chapter 2. The addressed area coverage problem as the two-phase coverage is formally defined in Chapter 3. The proposed method is presented in Chapter 4 with the evaluation results reported in Chapter 5. The Chapter 6 contains the description of the proposed testing platform and the results of the real deployment. Finally, Chapter 7 is devoted to the discussion of the proposed method and conclusion of the presented work.

CHAPTER 2

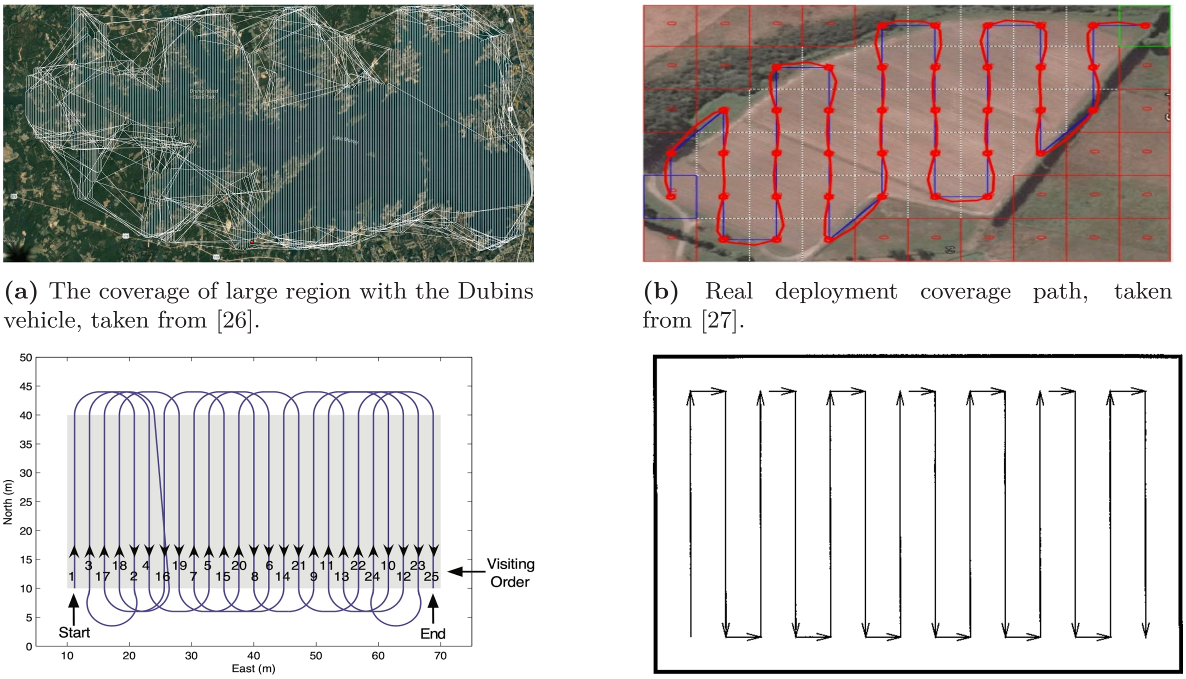
Related Work

The area coverage problem, sometimes also referred to as the *Coverage Path Planning* (CPP) [8], is a well-known task, and various solutions exist in the literature. The task, which is mostly comprehended as area coverage, involves one region only. The region usually can include obstacles, and in general, it does not need to be convex. A few methods for the extended area coverage problem, which includes multiple disconnected regions to be covered, were proposed in the existing literature. Therefore, we summarize the existing approaches in the three following sections as follows. The first section describes the general area coverage problem and existing approaches to modeling the fixed-wing UAV with the sensing capabilities. The second section describes existing methods for the area coverage problem with a single region. The third section is devoted to the methods capable of solving the problem with multiple regions.

2.1 Area Coverage Problem

The area coverage problem is a problem to find a cost-efficient path for a mobile robot or other vehicles concerning the objective function such that the vehicle's sensor or a tool visits all the points of the given area while avoiding collision with obstacles [6, 14, 8]. A vehicle can be equipped with a range sensor such as cameras, and therefore, the problem can also be treated as a problem to find a path for the vehicle such that its sensor footprint visits all the points of the given area. Practical tasks that can be formulated as area coverage problem includes floor cleaning [18], de-mining [7], lawn-moving [19], oceanographic mapping [20], terrain mapping [21], precision agriculture [1], arable farming [22] and others. Some of the existing approaches use the grid to represent the area, and for the convenience, the cells of the grid are usually of the same size as the footprint of the robot [14]. If such a grid is used, then it is enough to find the trajectory visiting all the grid cells. Few examples of the final coverage trajectory are displayed in Figure 2.1.

The CPP can be formulated as a variant of the *Traveling Salesman Problem* (TSP), a well-known combinatorial optimization problem solving the following task [23]. Given a complete graph with the rated edges, find a minimum cost cycle connecting all its vertices and visiting each vertex exactly once. The TSP is known to be NP-hard, but there are many existing solvers available such as the LKH [24] and the Concorde solver [25].



(a) The coverage of large region with the Dubins vehicle, taken from [26].

(b) Real deployment coverage path, taken from [27].

(c) Coverage pattern merging the boustrophedon pattern and dubins paths, taken from [28].

(d) Mostly used coverage pattern - the boustrophedon pattern, sometimes called zig-zag pattern, taken from [29].

Figure 2.1: Examples of coverage trajectories.

In this work, the coverage is assumed to be performed with fixed-wing UAV, and therefore, it is required to find the coverage trajectory such as all the points below the UAV are captured by camera present on board. Existing approaches for the coverage with fixed-wing UAV can adopt the grid representation of the environment or the decomposition of the area that leads to parallel straight coverage lines followed by the UAV. The optimal distance between the lines can be determined by the perception model described in the existing literature [30]. Because fixed-wing UAV is in the motion all the time in the 3D space, its footprint is determined by its altitude above the ground, field of view of the camera, and the camera's orientation with respect to the frame of the UAV. For simplicity, the camera is usually attached perpendicularly to the forward motion axis at the center of the UAV. Such positioning of the camera forms a rectangular footprint, which further simplifies the path calculation.

It is usually assumed that the UAV can keep the constant altitude above the ground, and therefore, the footprint stays constant. That can be further supported by terrain-following methods [17, 16]. Also, there are existing coverage approaches involving multi-rotor UAVs [21] and fixed-wing UAVs [31] under photogrammetric constraints proving the feasibility of such requirements. Therefore, the trajectory determination can be simplified to the plane only.

A sufficient model of the vehicle is needed to determine a feasible coverage path. Since the fixed-wing UAV is a non-holonomic vehicle, and the problem is reduced to the plane, the Dubins vehicle model [15] is utilized in the existing literature. Dubins vehicle is constrained by its minimal turning radius and constant forward velocity. There are also existing variants of the Dubins vehicle model that include acceleration limits and the maximum and minimum velocity of the vehicle with varying turning radius [32] or Dubins airplane model extended to three-dimensional space with limited maximum climb and dive angles [33].

The solution of the area coverage problem considers the given motion and perception models of the vehicle and computes the feasible coverage trajectory of the given area. Since the fixed-wing UAV is in motion all the time, the trajectory must be continuous, and it should visit every point in the area exactly once and avoid all the obstacles. In the survey written by Galceran et al. [8], we can find the following requirements regarding the area coverage in general.

1. Robot must move through all the points in the target area, covering it completely.
2. Robot must fill the region without overlapping paths.
3. Continuous and sequential operation without any repetition of paths is required.
4. Robot must avoid all obstacles.
5. Simple motion trajectories (e.g., straight lines or circles) should be used (for simplicity in control).
6. An *"optimal"* path is desired under available conditions.

Similar conditions are proposed also in [18] and a simplified version for the multi-rotor UAVs in [34]. As described in the following sections, regarding the fixed-wing UAV, it is often impossible to achieve some of the requirements. Regarding Requirement 1, due to the presence of obstacles, it is often not possible to cover every point in the area, or at least it is not possible to keep the same resolution. Regarding Requirement 2, realistic scenarios can require to have at least some minimum overlap of the paths for more straightforward map reconstruction from the captured images [21]. For Requirement 6, we need to describe the objective function. The CPP is an optimization problem, and the objective function usually used minimizes the total length of the final coverage path [15]. Recent articles address a different objective function, such as minimizing the total coverage time under energy and recharging constraints [35]. Another possible approach, given the battery capacity, is to minimize the number of recharges or visits to the charging station. Therefore the final coverage path should be close to optimum regarding the selected objective function. In the following two sections, a brief description of the existing methods is provided with respect to the established taxonomy.

2.2 Single Region Coverage

The existing planning methods are focused on the area coverage of a single region of arbitrary shape that can contain obstacles and holes and thus not be necessarily convex. The existing methods are divided into categories w.r.t. the original taxonomy defined in the survey written by Choset et al. [14]. Before the actual coverage path is determined, the region to be covered is pre-processed, and decomposed into non-overlapping cells or represented to allow methodological coverage path calculation. In [14], the CPP methods are divided into five categories from the point of view of the used area pre-processing approach as heuristic and randomized approaches, approximate cellular decompositions, semi-approximate decompositions, and exact cellular decompositions. The methods can be further labeled online, sometimes referred to as real-time, offline, and multi-robot methods independently on the main categories. The survey by Galceran et al. [8] and the most recent survey by Otto et al. [6] respect the original taxonomy, albeit it is sometimes difficult to fit the new methods into one of the existing categories.

Therefore, in the following, we add a new category to area coverage for the methods covering multiple disconnected regions. Each category is explained in a standalone section. Methods that are not feasible to UAV coverage (even with modifications), like contact sensor-based methods, are not included.

2.3 Online and Offline Methods

The offline methods compute the coverage path before the deployment. It is required to have complete knowledge about the area a priori to the deployment, e.g., a terrain map with the information about the obstacles. The offline methods can afford to explore the state space of the solutions more diligently and cut the wrong solutions immediately. Therefore, the computed solution is optimal or close to the optimum. The vehicle is requested only to follow the computed trajectory, and therefore, no special equipment is needed. Unfortunately, the knowledge of the area cannot always be available [14].

The online methods do not require any knowledge about the area a priori, except the area's boundary, which is a clear advantage. The actual coverage path during the online coverage is usually forcing advantageous coverage pattern or heuristic behavior and is adjusted every time a new landmark or obstacle is discovered. The disadvantage of the online methods is that the final solution is usually not optimal. A scenario with the obstacles can always be found, such that the online algorithm is forced to backtrack or overlap. Therefore, online solutions are not optimal with respect to the value of the objective function [14]. Nevertheless, as it is proposed in [36], the online methods are considered to be competitive to the offline methods, if the value of the computed solution can be achieved by the multiplication of the value produced by the offline method.

2.4 Heuristic and Randomized Approaches

The heuristic and randomized approaches are performing the coverage without any preprocessing of the region. These approaches are considered online and consist of a combination of simple heuristics and a random search. As it is pointed out in [14], simple heuristics such as wall following can create a behavior that can lead to imperfect coverage. The purely randomized coverage approach does not even need to have the internal representation of the region. Given enough time, it can still lead to sufficient coverage without any costly sensors and computational resources for performing localization on board of the vehicle.

The randomized coverage is analyzed regarding the cost/benefit in [14]. In simulated coverage scenarios with a robot team, the achieved ratio between the randomized and methodological approaches is reported to be 5 to 1. In such cases, if the coverage robots without localization can be constructed for one-fifth of the cost, the randomized approach may be a suitable option.

More arguments for the heuristic and randomized approaches are proposed in [37] that is focused on the area de-mining. In the proposed scenarios, the mine is discovered with a probability lower than 1, if the robot gets to its proximity. Therefore, even the methodological coverage does not guarantee to discover all the mines in the area. The lower the probability of detection, the more of the methodological approach's advantages diminish in comparison with a randomized approach.

The heuristic and randomized approaches' main advantages are the low cost of the robots and required equipment since an internal representation is not needed, low computational requirements are sufficient for the coverage path computation and fast deployment.

2.5 Approximate Cellular Decomposition

The approximate cellular decomposition methods are also referred to as grid-based methods [8]. In this method, the region to be covered is represented as a grid composed of homogeneous non-intersecting cells, the region can be non-convex, and it can contain arbitrary obstacles and holes. The union of the cells does not precisely reproduce the original environment. The shape and size of the cells do not allow to exactly copy the shape of the obstacles, holes, and the boundary of the region, which is why these methods are considered to be approximating [14]. The cell is considered to be covered once the mobile robot enters it, and thus the cell is assumed to be the robot footprint. The grid cells usually have a rectangular shape, but triangular-shaped cells [18] or hexagonal cells [34] are used as well. The actual coverage path is calculated so that all cells are visited exactly once, if possible. The existing methods differ in how the grid traversal is performed, and the actual coverage path computation. Six selected representative approximate cellular methods are overviewed in the following sections.

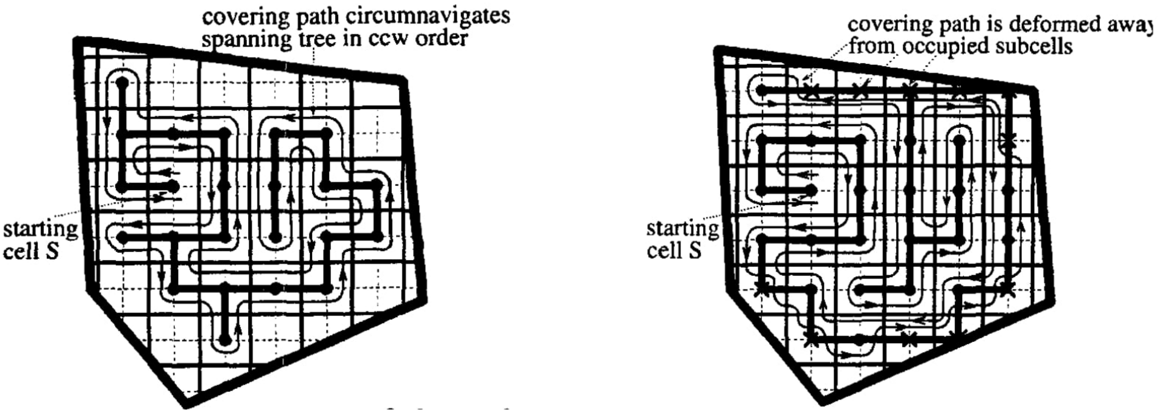
2.5.1 Spiral Spanning Tree Coverage

In [38], an online CPP algorithm, is proposed using a grid representation of the planar non-convex region with stationary obstacles. The algorithm is designed for mobile robots with a footprint of the square shape and size D. The area is first divided into the grid consisting of the cells of the size $2D$; therefore, each cell of the grid contains four sub-cells of the size of the robot footprint D. The robot has a given start cell and must completely cover the whole area. The path is determined by following iteratively constructed spanning tree of the graph induced by the grid representation. It is allowed to move the robot in orthogonal directions only and without the rotation. Therefore, the edge of the spanning tree can connect the orthogonal cells only. In every step, it is determined which edge connecting the actual cell, and the neighboring cell is added to the existing spanning tree. The edge can connect the actual cell with the new cell only. The cell is labeled as new only if all its sub-cells are unvisited; otherwise, it is labeled as an old cell. The spanning tree is iteratively constructed counter-clockwise, and the robot must follow its edging on the right side.

The original spiral *Spanning Tree Coverage* (STC), proposed by the same authors of [38], prohibits visitation of any partially occupied cells by the obstacles that have been addressed in [38]. In this version, it is allowed to visit partially occupied cells, but the main disadvantage is that some parts of the final path overlap and visit the same sub-cells more than once. Examples of the results provided by both algorithm variants are depicted in Figure 2.2.

2.5.2 Wavefront Algorithm

An offline coverage method based on wavefront propagation is proposed in [39]. The method operates above a grid representing environment with rectangular shaped cells and with the presence of the obstacles based on the distance transform path planning methodology. The algorithm was experimentally deployed on the Yamabico mobile robot, and it was also deployed on a UAV by L. Nam et al. [27]. The algorithm produces the complete coverage path of the given area leading from the given start location of the robot to the robot's given goal location. The free cells in the area are firstly rated by a value representing their distance from the goal cell by propagating the wavefront. The coverage path is then constructed by following the


covering path circumnavigates
spanning tree in ccw order
starting
cell S
covering path is deformed away
from occupied subcells
starting
cell S

(a) Example of the original version of the STC algorithm prohibiting coverage of partially occupied cells, taken from [38]

(b) Example of the extended version of the STC algorithm allowing coverage of partially occupied cells, taken from [38]

Figure 2.2: Examples of coverage paths found by the Spanning Tree Coverage (STC) algorithm [38].

steepest descent from the start location to the goal location. The robot has to remember which cells were already visited and must have access to the values of the neighboring cells. The advantage of this algorithm is that the start and the end position of the robot can differ. It is beneficial in situations like sweeping the floor in the warehouses, where the robot starts at the entrance and finishes at the end, ready to continue to the next warehouse. Also, if a different function for the propagation of the wavefront is used, different behavior can be induced.

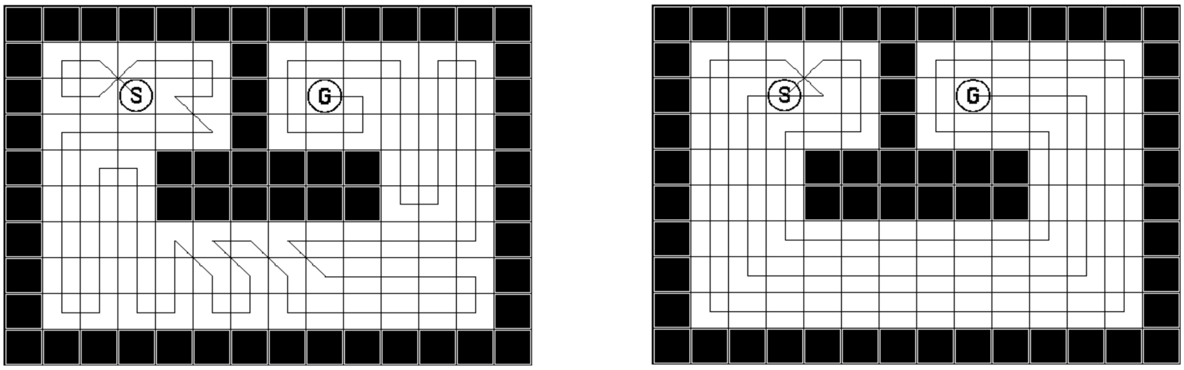
On the other hand, the original algorithm produces a coverage path that contains many turns that can cause problems with the localization. Therefore the authors proposed an extended version of the algorithm where the distance transform is replaced by the path transform consisting of two steps. Every obstacle becomes a goal in the first step, and the distance wavefront from the obstacles is propagated through the grid. In the second step, the actual value of each cell is computed as:

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

where P denotes the set of all possible paths from the cell c to the goal. The constant α represents the value how much is the robot avoiding the obstacles, the function obstacle(x) returns the value from the first distance wavefront from the obstacles and the function length(x) returns the length of the path p from the given cell to the goal. Final paths produced by both versions of the algorithm are depicted in Figure 2.3.

2.5.3 Backtracking Spiral Algorithm

The *Backtracking Spiral Algorithm* (BSA), originally proposed in [40], is an offline method for square-shaped grid regions. The main idea of the BSA is to cover the coarse-grained grid based on modified exhaustive depth-first search producing spiral-like paths. The final coverage path is composed of multiple simple spirals connected by the traversals over already covered cells. The traversal overlaps are unavoidable as each spiral is nesting to itself until the central point is achieved with no way to get to the beginning of the next spiral without crossing already



(a) Coverage path produced by the distance transform version of the wavefront algorithm, taken from [39].

(b) Coverage path produced by the path transform version of the wavefront algorithm, taken from [39].

Figure 2.3: Examples of final coverage trajectories determined by the wavefront algorithm [39].

covered paths. The original algorithm was unable to cover the partially occupied cells and therefore, it has been extended to also cover the partially occupied cells [41], see Figure 2.4.

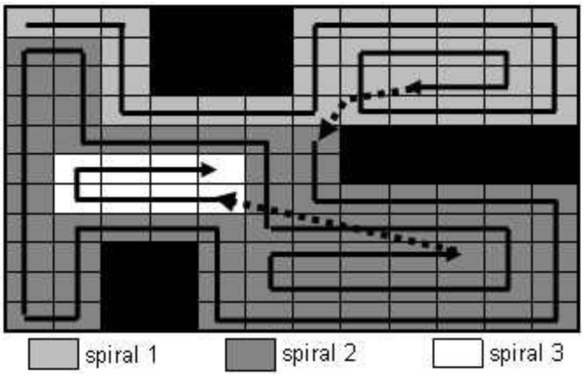
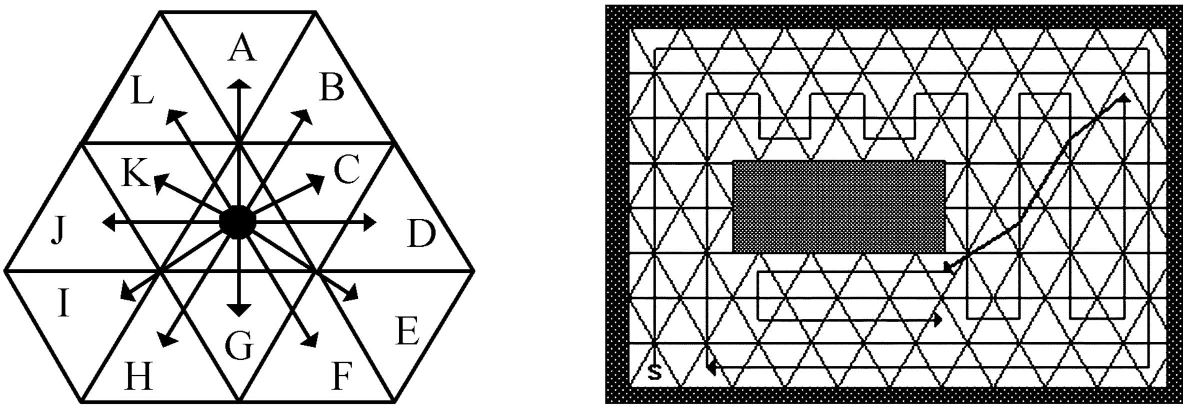


Figure 2.4: Example of the coverage path computed by the BSA, taken from [40].

2.5.4 Triangular-Cell-Based Representation

A method proposed in [18], similar to the previous one, is a distance-transform-based path-planning method with a new representation of the region. Unlike the wavefront algorithm [39], this method is designed to be used online without prior knowledge about the environment. The map is represented by the triangular cells instead of the rectangular that allows planning smoother paths with the improved approximation of the region and stationary obstacles. Unlike the rectangles, which allows 4 to 8 directions for the next move, the triangular cells allow 12 directions for the next move, and therefore, finer movement adjustments, see Figure 2.5a. The authors of the method also modified the distance function to avoid the same values of the cells. In the original distance wavefront, the value starts at one and is increased by every step, which produces many cells with the same values, and the final trajectory is spiral-like with many turns. A new function increases the value by one only if the heading to the next cell is the same. If the heading changes, the value can be increased by 3 or 4.

The method determines the coverage path in three steps. In the first step, the wall-following is performed, since the method is entirely online. Once the boundary is determined, the triangular map representation of the inside of the area is calculated. In the last step, the map is covered using the movement templates. The algorithm distinguishes the covered cells and the uncovered cells, and the actual coverage of the region is performed using predefined movement templates to perform the traversal through the grid and do not leave any uncovered cells. The example of the final path is depicted in Figure 2.5b.



(a) Triangular representation of the cells of the grid allows 12 directions for the next move, taken from [18].

(b) Final coverage path computed by the online wavefront algorithm with the triangular representation, taken from [18].

Figure 2.5: Examples of triangular cell and a triangular grid [18].

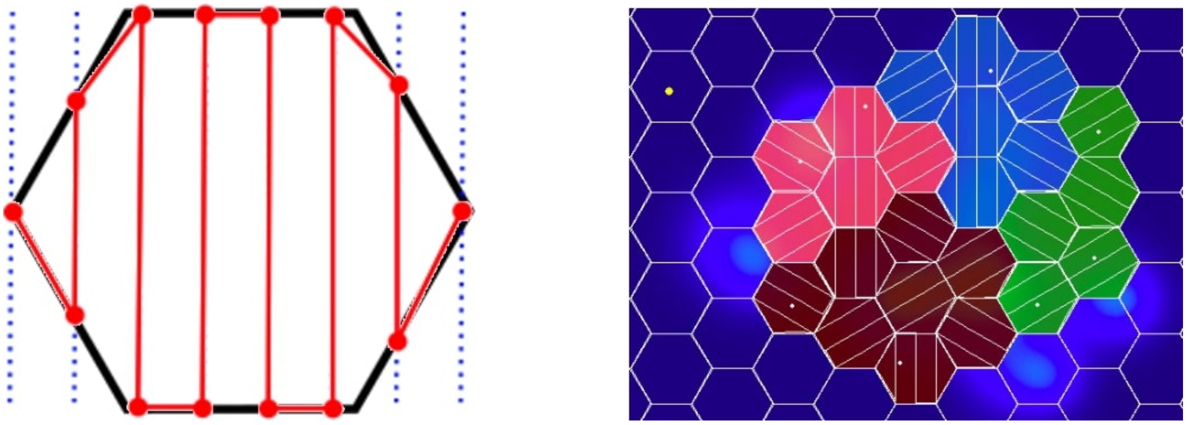
2.5.5 Hexagonal based Representation

A method designed for multi-rotor UAVs is proposed in [34] based on the hexagonal grid representation of the region to be covered. The method was verified in simulations and outdoor experiments. Unlike the previously mentioned methods, the method [34] operates in the 3D environment representation and distributes the coverage among a team of UAVs. The input of the algorithm is the terrain map and a bounded area. The area is approximated by the hexagonal cells, and a group of cells is assigned to each UAV involved. The *Traveling Salesman Problem* (TSP) for each assigned group of cells is solved, and each cell is covered by the lawn mover pattern [8] as follows.

Lawn mover pattern, also known as the boustrophedon pattern, zig-zag pattern, or sweep-motion, consists of connected straight lines. Each line goes from one boundary determined either by the area, end of the cell, or the obstacle to the opposite boundary or the obstacle wall. There is a space between each line equal to the footprint's size; therefore, if the pattern is followed, no portion of the visited area overlaps. It is a methodological and straightforward approach to the coverage problem with many advantages. It is also feasible to be used for the coverage with the fixed-wing UAVs with necessary modifications, further detailed in this chapter. The examples of the sweep-motion pattern and the coverage path proposed by the method are depicted in Figure 2.6.

2.5.6 Neural Network-based Area Coverage

Online grid-based coverage path determination using neural networks is proposed in [42]. The grid consists of the square-shaped cells, and each cell is assigned to one neuron in the

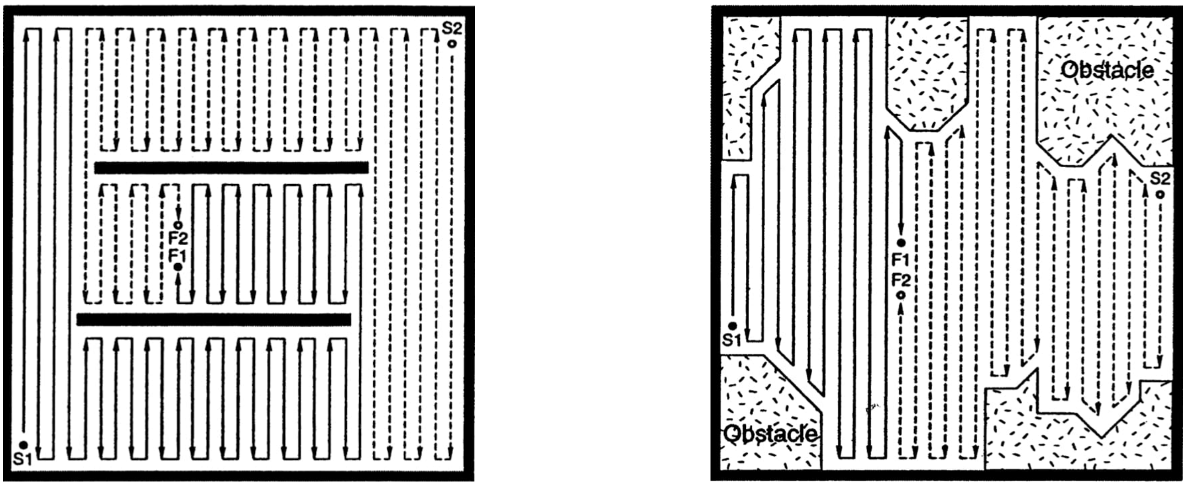


(a) Boustrophedon pattern projected onto the hexagonal cell, taken from [34].

(b) Hexagonal cells divided among the team of UAVs for the coverage, taken from [34].

Figure 2.6: Hexagonal grid pre-processing of the area to be covered and determination of the coverage path [34].

network. Due to the structure of the network and used functions, the activated neurons represent unvisited areas and globally propagate through the network. Also, the obstacles and visited areas represent in-active neurons and propagate only locally. The actual path is determined from the previous location and heading of the vehicle and activity of the neural network. Basically, the vehicle is attracted by unvisited areas and repelled by the obstacles, and thus avoiding them. The final trajectory is smooth and minimizing the number of turns. The main advantage of the method is that it can successfully navigate the vehicle through a non-stationary environment. Thus, obstacles can move, and visited areas can again become unvisited. An example of the resulting path is depicted in Figure 2.7.



(a) First example of the coverage path for two robots computed by the neural network, taken from [42].

(b) Second example of the coverage path for two robots computed by the neural network, taken from [42].

Figure 2.7: Examples of the final coverage paths of the indoor and outdoor environment determined by the neural network [42].

2.6 Semi-approximate Decomposition

The semi-approximate decomposition methods can be seen as an extension of the approximate-cellular decomposition methods. The cell representation is still used, but the cells are not organized into the homogeneous grid anymore. Instead, one dimension of the cell is fixed, the width usually, and the other dimension can be of arbitrary size with respect to the location of the obstacles in the region. Moreover, the shape of the boundary in another dimension can also be arbitrary with respect to the obstacles and the boundary of the region.

A semi-approximate decomposition method is proposed by Hert et al. in [20]. The method is feasible for the coverage of the non-convex planar regions with the presence of the obstacles. The method determines the cell around the starting position of the vehicle and starts its coverage with the sweep-motion pattern. If another uncovered cell (called inlet) is discovered during the coverage, the on-going coverage of the current cell is interrupted, and the newly discovered inlet is covered immediately, using the same pattern and rules. The vehicle must remember the interruption point; so, it can return to it after the inlet is completely covered and finish the coverage. The algorithm is recursive, and therefore, if the inlet inside the inlet is discovered, it is also covered by application of the same rules. During the coverage, the obstacle can be discovered, called an island. The cells around the islands are covered with slightly modified methods for the coverage of the inlets. Because the vehicle must return to the interruption points, some parts of the region may be visited more than once. An example of the final coverage path found by the semi-approximate method is depicted in Figure 2.8.

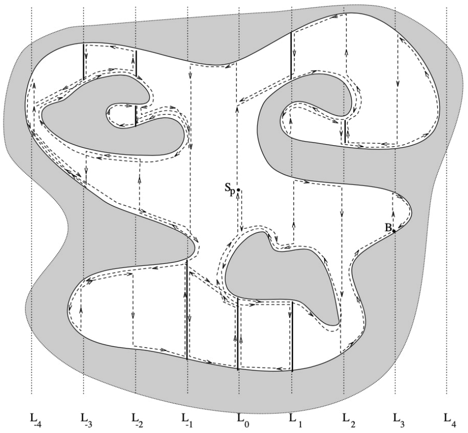


Figure 2.8: An example of coverage path produced by semi-approximate approach, taken from [20].

2.7 Exact Cellular Decomposition

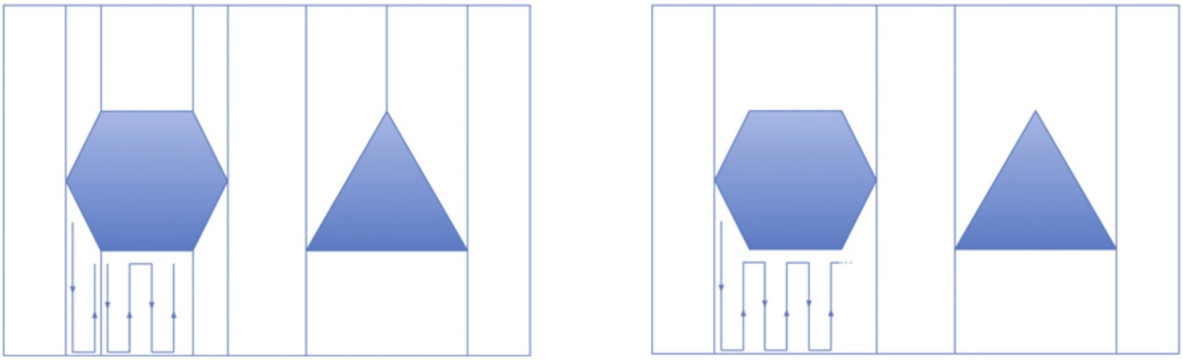
Exact cellular decomposition further generalizes the cell-based decomposition approach to accurately represent the region to be covered with no requirements on the fixed width/height or unified shape of the cells. There are several existing approaches, each performing a different decomposition of the region into the cells. However, the common trait is that the cells are non-intersecting, their union accurately depicts the region, and the individual cells can be covered by simple sweep-motions, i.e., by the boustrophedon pattern depicted in Figure 2.1d. The task is to divide the area using one of the decomposition strategies and create an adjacency graph of the cells. The walk-through then determines the final coverage path over each node in the corresponding adjacency graph [14, 8]. Four selected representative decompositions are detailed in the following paragraphs.

2.7.1 Trapezoidal Decomposition

The trapezoidal decomposition [43] is a simple decomposition method suitable for the complete coverage of planar non-convex polygonal regions. The method decomposes the region into simple trapezoidal cells determined by the motion of the sweep line, the boundary of the region, and the critical points determined by the vertices of the obstacles. The sweep line is moved through the region in a sweep direction, and each time it intersects the vertex of the obstacle, new cells are created. Each cell represents a node in an adjacency graph for which the optimal sequence of the visits is computed. Resulting cells have a trapezoidal shape, and the sweep-motion covers each of them. The example of possible decomposition of the region is depicted in Figure 2.9a.

2.7.2 Boustrophedon Decomposition

The Boustrophedon decomposition described in [29] addresses the problem of the trapezoidal decomposition that produces more cells than it is necessary and could be easily merged, such as neighboring cells that are not divided by any obstacle or boundary. In the Boustrophedon decomposition, cells are merged during the sweep line's motion through the area, if a new cell is created, and the connectivity in the corresponding adjacency graph stays the same or decreases. Since fewer cells are produced in comparison with the trapezoidal decomposition, the corresponding adjacency graph contains fewer nodes, and final decomposition allows optimal coverage pattern. An example of the division and the final coverage path is depicted in Figure 2.9b.



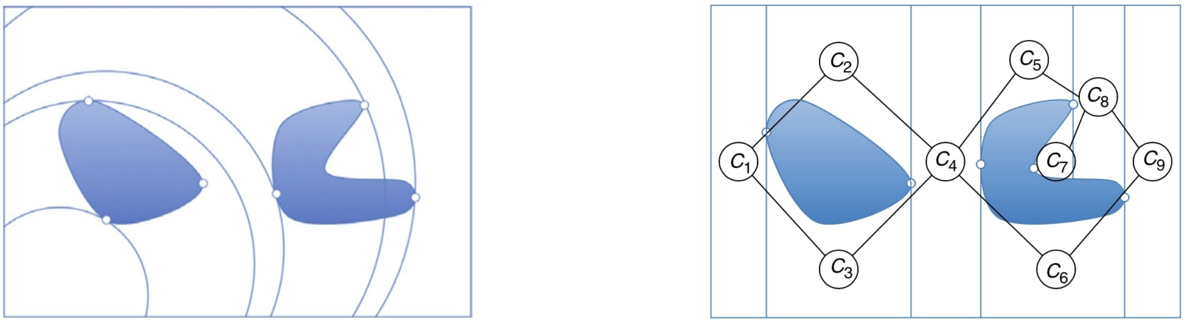
(a) An example of the region decomposed by the Trapezoidal decomposition method, taken from [8].

(b) An example of the region decomposed by the Boustrophedon decomposition method, taken from [8].

Figure 2.9: Examples of the Trapezoidal and Boustrophedon decomposition methods [8].

2.7.3 Morse-based Cellular Decomposition

The Morse-based cellular decomposition [44] is a general exact-cellular decomposition method based on critical points defined by the selected Morse function [45] and the obstacle boundary. For example, we can achieve the Boustrophedon decomposition described in Section 2.7.2, using appropriate Morse function. The method divides the region based on the sweep line determined by the selected Morse function and by calculating the critical points bounded by the boundary of the obstacle. Different functions lead to a different decomposition of the region that can be advantageous for different coverage patterns, e.g., the spiral pattern depicted in Figure 2.10a. It is theoretically proven that the approach could be generalized into n-dimensional space [8], and a coverage method based on Morse functions was developed for online coverage in [46]. An example of a possible region decomposition is depicted in Figure 2.10b.



(a) Morse-based decomposition for the spiral coverage pattern, taken from [8].

(b) An example of the Morse-based decomposition method, taken from [8].

Figure 2.10: Examples of the Morse-based decomposition [44].

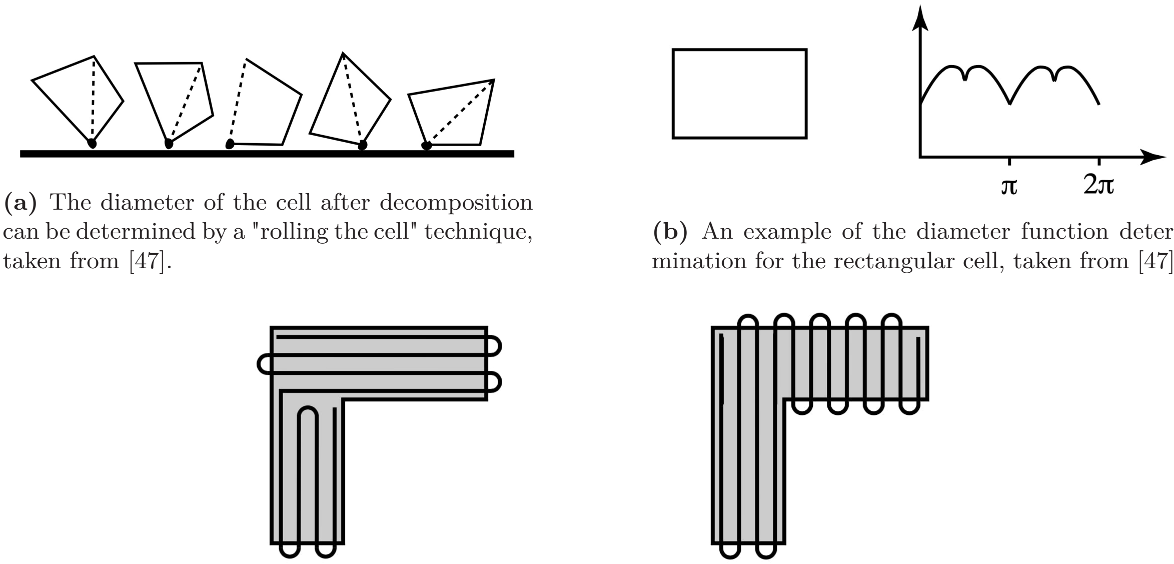
2.7.4 Minimal Sum of Altitudes Decomposition

The aforementioned exact cellular decomposition methods are based on the sweep motion division of the region and coverage of each cell in the same direction as the direction of the sweep motion. Since the often-used sweep-motion pattern consists of straight connect lines, the author of [47] reason that different sweep directions produce the straight lines of coverage pattern with roughly the same length but a different number of turns. The turns are disadvantageous not only because it slows down the coverage time and can multiply error caused by the localization, but it also prolongs the final coverage path. Therefore, the number of turns should be minimized. Based on that reasoning, the authors proposed a method called the *Minimal Sum of Altitudes* (MSA) decomposition.

The MSA searches for a region decomposition such that the sum of the altitudes of all cells is minimal. Thus the decomposition minimizes the number of turns, where the altitudes measured relative to the sweep direction assigned to the particular cell [47]. The author presents how the cell diameter influences the number of turns, where for a longer diameter, shorter straight lines have to be followed, and thus more turns need to be performed. For a fixed-wing UAV, the problem is increased if the vehicle cannot connect to the next neighboring straight line. An example of the MSA decomposition and MSA diameter function is shown in Figure 2.11.

2.8 Multi-robot Coverage

Many real-life scenarios consider the deployment of multiple robots to solve the given task, and the area coverage problem is no exception. For example, the Hexagonal based representation



(a) The diameter of the cell after decomposition can be determined by a "rolling the cell" technique, taken from [47].

(b) An example of the diameter function determination for the rectangular cell, taken from [47].

(c) Decomposition of the region into two cells with different sweep direction(on the left) and decomposition of the same region with just only one sweep direction (on the right), taken from [47].

Figure 2.11: The illustrations of the MSA decomposition [47].

overviewed in Section 2.5.5 is motivated by a disaster management scenario, where groups of cells are divided between multiple UAVs to decrease the coverage time and rescue survivors quickly [34]. Hence, the deployment of multiple vehicles is motivated to decrease the total time to finish the task. Moreover, during the real deployment, many other practical problems surface, such as reliability and robustness of the localization, a failure rate of the robots and equipment, and random events that can influence the mission and cannot be prevented entirely. Assigning multiple UAVs can reduce the impact of such issues.

The problem encountered in multi-robot coverage is the determination and assignment of the workload between the team members. The actual proportion of the work should be similar or should consider each team member's capabilities as deployed UAVs can have different battery capacity, different footprint, etc. In the rest of this section, selected representative multi-robot coverage approaches are presented to provide an overview of the existing methods.

2.8.1 Polygonal Area Decomposition

Polygonal area decomposition [48] is an exact cellular decomposition method designed for a team of UAVs that respects the robots' capabilities. The capabilities are represented by the desired size of the area to be covered where the target region should be decomposed to the cells of the particular size, and each such a cell is then assigned to the robot. Simultaneously, each robot covers exactly one cell only. The required sizes can be determined with respect to each team member's actual capabilities, like the battery capacity, flight speed, and footprint size. Moreover, each robot has a starting point, referred to as its site. The original method [48] can divide the polygonal convex regions only, but the extended version can also address non-convex regions. The method is based on a sweep line with recursive divide-and-conquer techniques. The final cells are convex and suitable for the sweep-motion coverage. An example of the decomposed region is depicted in Figure 2.12.

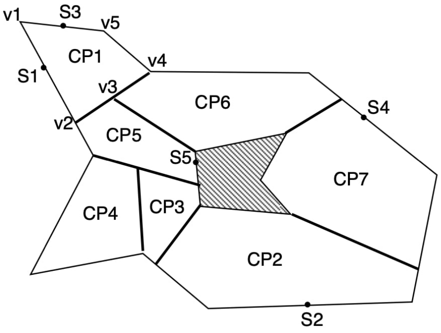
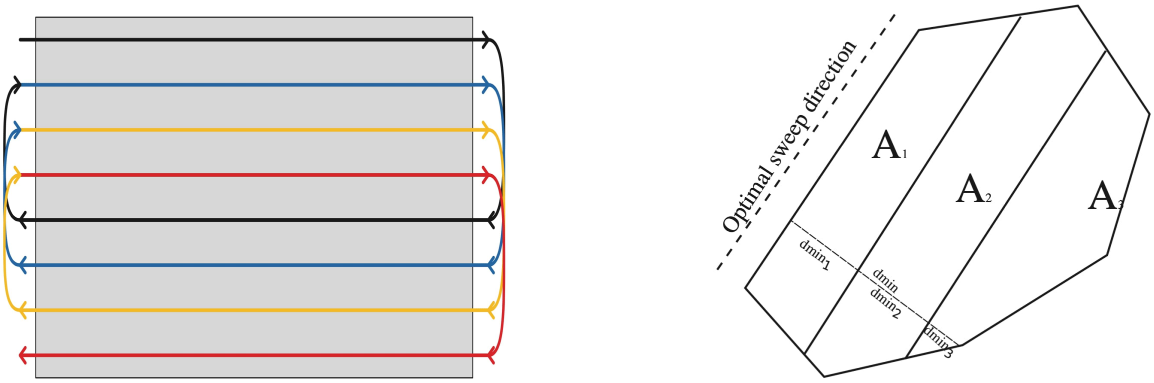


Figure 2.12: An example of the region decomposed into 7 convex polygons with five entry sites, taken from [48].

2.8.2 Optimal Sweep Direction and Zamboni Pattern Coverage

Multi-robot coverage with the homogenous team is proposed in [49] that is suitable for the coverage of non-convex polygonal regions without the presence of obstacles. In the first step, the non-convex region is transferred to the convex region using the convex envelope. The size of the new region is enlarged, but the authors argue that the extra amount of coverage is usually low enough to be relaxed and can always be diminished by the implementation of the exact cellular decomposition method. For a new region, the optimal sweep direction is computed by searching the minimum value of the diameter function [47]. Once the optimal sweep direction is found, the region is divided into similarly sized sub-regions respecting the direction. For each sub-region, the coverage path is determined, and then the sub-region is assigned to the UAV. The actual coverage pattern used is a modification of the Zamboni pattern. The Zamboni pattern is called after the cars used to resurface the ice in the ice-hockey stadiums, and an example of such a pattern is depicted in Figure 2.13a. The method [49] allows the repeated coverage of specified coverage segments; therefore, it simulates the surveillance of the given region. It was also experimentally verified with the fixed-wing UAVs. An example of the decomposed region into three sub-regions is depicted in Figure 2.13b.



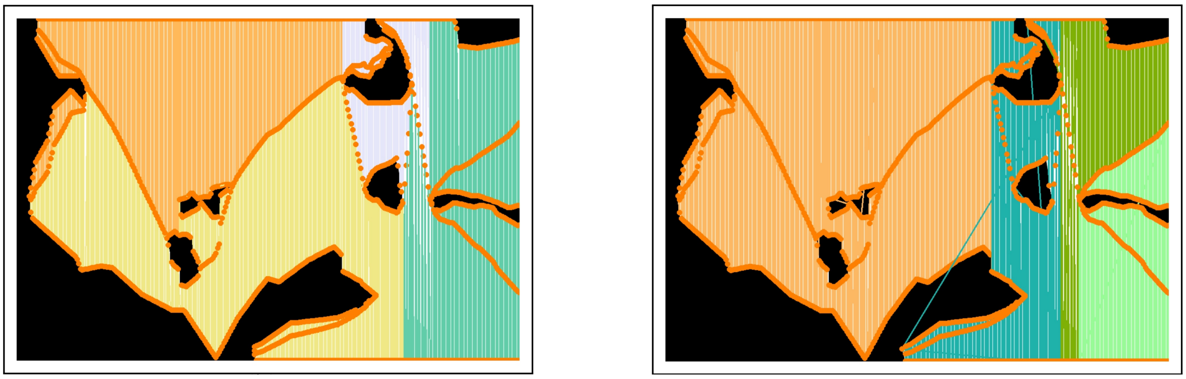
(a) The depiction of the Zambonni pattern, taken from [49].

(b) Example of optimal decomposition, taken from [49].

Figure 2.13: Ilustration of the Zambonni pattern and the optimal decomposition.

2.8.3 Boustrophedon Decomposition-based Efficient Coverage

A variant of the Boustrophedon decomposition for a team of fixed-wing UAVs is proposed in [50] to perform complete coverage of non-convex regions with the obstacles of the arbitrary shape. Firstly, the given region is decomposed using the Boustrophedon exact-cellular decomposition. In the following step, the coverage of the global task is determined. The actual coverage pattern is determined by the *Efficient Complete Coverage Algorithm* (ECC) [51] that is based on the computation of the coverage segments with respect to the footprint of the UAV and selected sweep direction with the following determination of the Euler path visiting all the segments. Once the Euler path is determined, it is divided among the robots in a fashion that each robot covers a similarly sized area. The Euler path is computed with respect to the global starting and end locations of the whole team. The authors also proposed the second variant of the algorithm that firstly divides the region and assigns the cells to individual robots. Then, individual coverage paths are determined. Examples of the decomposition are depicted in Figure 2.14.



(a) An example of the region decomposed using the coverage with clustering route method, taken from [50].

(b) An example of the region decomposed using the coverage with area clustering method, taken from [50].

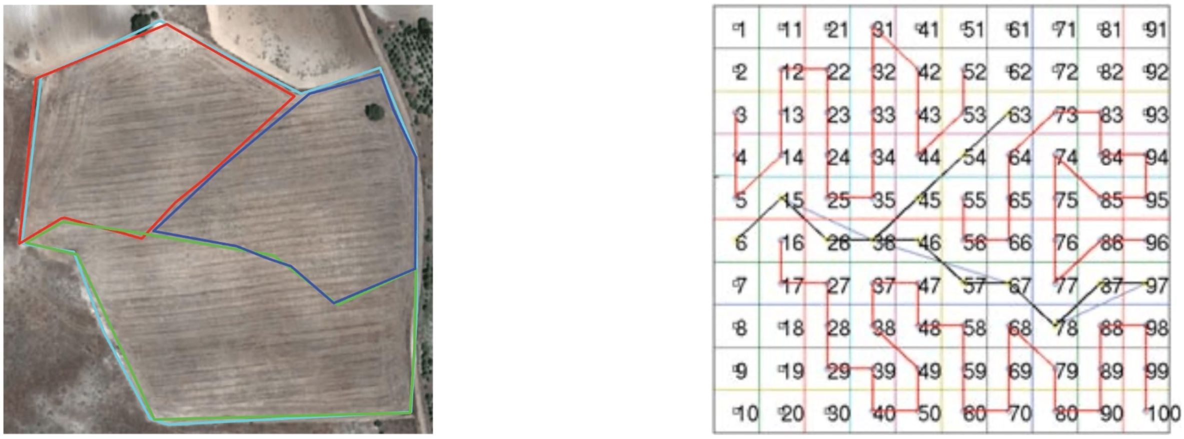
Figure 2.14: An examples of the boustrophedon-based coverage decomposition [50].

2.8.4 Game Theory-based Multi-robot Coverage

Game theory-based multi-robot coverage is proposed in [1], where the best division among the UAVs with non-equal capabilities is motivated by covering as much area as the individual vehicles can, and thus maximizing their reward function. The sub-regions are represented as cells of the rectangular-shaped grid, and the UAVs negotiate and offer to cover as much area as possible with respect to their abilities. If a UAV is not satisfied, a counter-offer is proposed with a different setup. In each iteration of the negotiation, the area is decreased by a specified step, and the negotiation continues until all UAVs are satisfied. The team of the UAVs was deployed in a scenario over the vineyards to verify the method experimentally, and the resulting coverage path is depicted in Figure 2.15.

2.9 Multiple Regions Coverage

In this thesis, we are focused on multi-region coverage, and regarding the overview of the existing methods presented in the previous sections, the single region coverage problem can



(a) Region decomposition after negotiation among the UAV agents, taken from [1].

(b) The region discretization and actual coverage path of each UAV agent, taken from [1].

Figure 2.15: An example of the area coverage solved by the game theory based method [1].

be considered as well-addressed with many practical approaches. Thus, existing groundwork support generalization of the methods to the coverage problem with multiple disconnected regions, where any single region can be addressed as a single area coverage problem. Since the multi-region coverage attracts the research community's attention relatively recently, only a few papers can be found in the literature. Among many real-life scenarios of multi-region coverage, we present three selected cases to motivate the addressed problem further and show an example of the application, a prior overview of the existing approaches.

An example of the multiple disconnected regions distant from each other from a few dozens of meters to dozens of kilometers can be found in the search-and-rescue when a tsunami hit the east coast of Japan in 2011 and damaged or destroyed over one million buildings–120 000 were destroyed, 278 000 were half-destroyed, and 726 000 were partially destroyed [52]. As shown in Figure 2.16, dozens of cities spread ashore were damaged, and people living in them seriously endangered. The most critical is the monitoring of the cities and densely populated locations, where the monitoring mission can be formulated as the multi-region coverage problem.

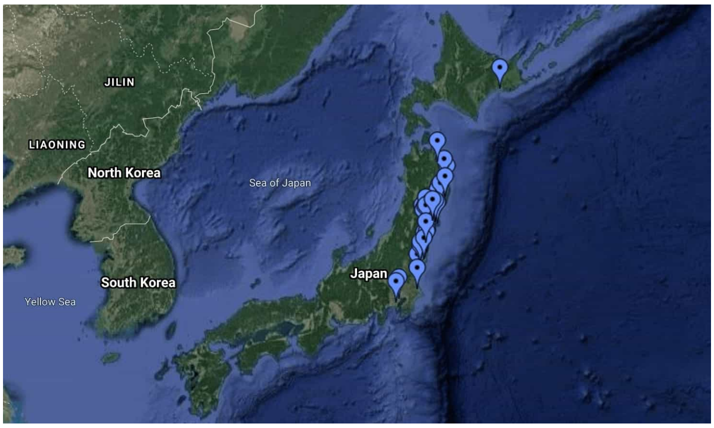


Figure 2.16: Impacted Japan cities by the tsunami in 2011, credits: Google[1]

The next example of the natural disaster and motivation for the area coverage is wildfire management [5]. Natural fires can occur in multiple regions simultaneously and spread across the forests and farms, and it is of high importance to monitor the situation. As the authors of [5] argue, it is advantageous and cost-efficient to deploy multiple fixed-wing UAVs to capture aerial images of influenced regions. As in the previous example, the regions can be dozens of kilometers distant.

Finally, the last but not least example of regions in proximity of a few hundreds of meters to a few kilometers is from agriculture. Consider the crop spraying or conventional aerial imaging of the fields with a multi-spectral camera for precision agriculture. There is more time to determine the optimal path in such scenarios because, unlike natural disasters and wildfires, the regions to be covered do not rapidly change. Besides, the multi-region coverage can be found in forestry monitoring for greenhouse gas emissions [53] or biomass estimation [54], to name a few others.

The multi-region coverage task is to completely cover all regions and gather the data in a cost-efficient manner and avoid all the obstacles. Moreover, since the task consists of large-scale scenarios, it should allow to clearly define the starting and ending location of the covering UAV. In the rest of this chapter, we provide an overview of the existing multi-region coverage problem methods, and we point out their differences.

2.9.1 Boustrophedon Cell Coverage in Cooperation with UGV

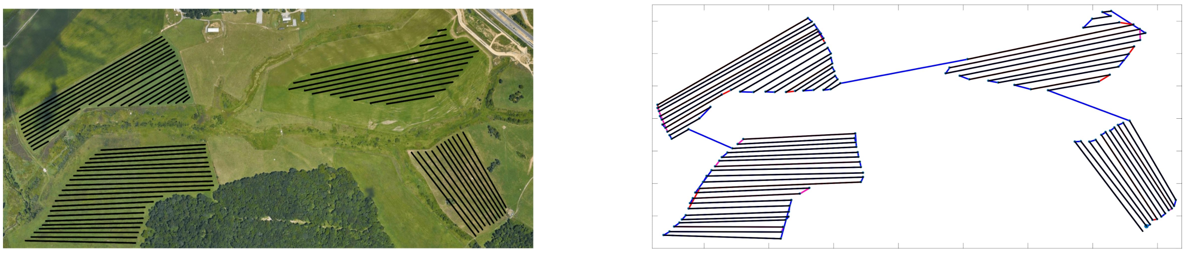
The cooperation of UAVs with an *Unmanned Ground Vehicles* (UGV) in a multi-region coverage problem is proposed in [11]. The method is focused on the multi-rotor UAV coverage of multiple boustrophedon cells that can be disconnected and distant from each other. The UGV is working as a battery recharging mobile station, but it can also be utilized to transport the UAV during its recharging, and thus increases efficiency. The authors intentionally defined the size and shape of boustrophedon cells to exactly match the width of the UAVs footprint. Therefore, it takes just a single pass in a given direction to completely cover a cell. Each cell can be approached from one of the two entry points, and thus it can be covered from one of the two allowed directions. The length of the cell is arbitrary, and the user defines it. The task is to find the optimal coverage path that completely covers all the given boustrophedon cells and ensures the UAV never runs out of the battery. Unlike the classical area coverage problem, a path for the UGV has to be also determined.

Regarding the addressed multi-region coverage, the interesting part of the method [11] is the definition of the individual boustrophedon cells. In the previous methods (such as those described in Section 2.7.2 and Section 2.8.3), the boustrophedon decomposition is employed to produce cells as large as possible while maintaining the convexity of individual cells, which support to cover the decomposed area optimally. Here, in multi-region coverage, however, it is problematic if the non-holonomic vehicle is used for the actual coverage because it is usually unable to follow the regular sweep-motion consisting of straight lines and orthogonal connecting lines. In such a situation, it is advantageous to use a minimalistic approach for the definition of the boustrophedon cell because it fits better with the actual coverage with the non-holonomic vehicle. A more detailed explanation is provided in Chapter 3.

The method [11] addresses the task as a variant of the GTSP extended to determine the path also for the UGV. The results have been verified in outdoor experiments, and an example

[2] https://www.google.com/maps/d/u/0/viewer?ie=UTF8&hl=en&msa=0&ll=38.757898120227956%2C140.18004956875916&spn=22.232593%2C53.569336&t=h&z=5&mid=1Z2jVOS3e-hDFNQEJ3zHpkg2APdA

of the therein reported final coverage path is depicted in Figure 2.17. As can be seen, the boustrophedon cells are not far from each other, partially caused by a small velocity of the UAV and the UGV.



(a) Input boustrophedon cells into the multi-region coverage algorithm, taken from [11].

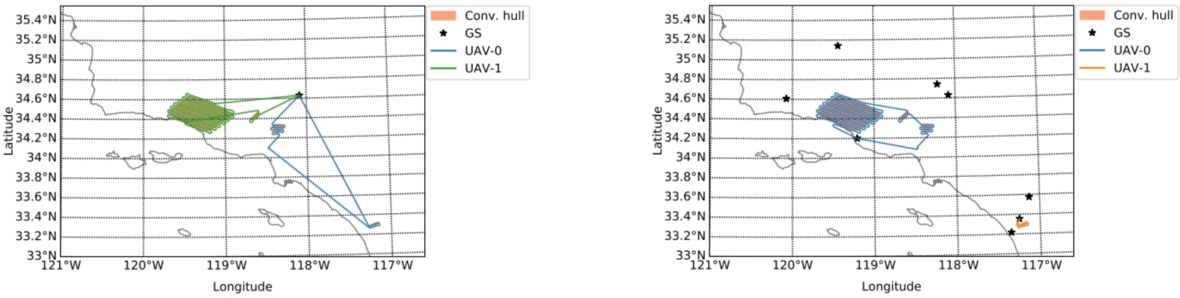
(b) Output coverage tour of the same instance of boustrophedon cells, taken from [11].

Figure 2.17: An example of coverage path determined by the boustrophedon cell based multi-region coverage methods [11].

2.9.2 Multiple UAVs based Large Scale Coverage and Distribution

A different situation is encountered in [5], where the authors are focused on wildfire monitoring in California by a team of fixed-wing UAVs. The wildfires are usually far from each other, reaching distances up to hundreds of kilometers, which disqualifies multi-rotor UAVs because of short operational time. The method presented in [5] is approaching the problem in realistic scenarios involving multiple UAV stations and multiple UAVs. Each station can contain a different number of UAVs, and the task is to distribute the coverage with respect to the proximity of stationed UAVs in an efficient manner.

Once any region with the wildfire is entered, it has to be covered entirely by the UAV. Each region is transformed into a convex region using the convex envelope to simplify the coverage of individual regions. The actual coverage path is then determined in two steps. In the first step, the optimal coverage pattern for each region is determined. The UAV is modeled as Dubins vehicle, and for simplicity, the vehicles are considered identical. In the second step, the multi-depot vehicle routing problem with all UAVs and ground stations is solved. The final solution is achieved as a combination of both stages. The algorithm was verified using the numerical simulations and selected reported results [5] are depicted in Figure 2.18.



(a) Large scale wildfire management coverage path, scenario 1, taken from [5].

(b) Large scale wildfire management coverage path, scenario 2, taken from [5].

Figure 2.18: Examples of large scale multi-region coverage in wildfire management scenario [5].

Table 2.1: Area coverage problem methods summarization table (AC - approximate cellular, SA - semi approximate, EC - exact cellular, MR - multi-region, N/A - not classified, N - No, Y - Yes).

Method	Online	Type	Multi	Src.
Spiral Spanning Tree Coverage	Y	AC	N	[38]
Wavefront algorithm	N	AC	N	[39]
Backtracking Spiral Algorithm	N	AC	N	[40]
Triangular-Cell-Based	Y	AC	N	[18]
Hexagonal-Cell-Based	N	AC	Y	[34]
Neural networks online coverage algorithm	Y	AC	Y	[42]
Online terrain coverage algorithm	Y	SA	N	[20]
Trapezoidal decomposition	N/A	EC	N/A	[43]
Boustrophedon decomposition	N/A	EC	N/A	[29]
Morse-based cellular decomposition	N/A	EC	N/A	[44]
Minimal sum of altitudes	N	EC	N	[47]
Polygonal area decomposition	N	EC	Y	[48]
Optimal sweep direction and Zamboni pattern	N	EC	Y	[49]
Boustrophedon decomposition-based efficient coverage	N	EC	Y	[50]
Multi robot coverage and game theory	N	AC	Y	[1]
Boustrophedon cell coverage in cooperation with UGV	N	MR	N	[11]
Multiple UAV large scale coverage and distribution	N	MR	Y	[5]

The interesting part of the solution is that more regions to be covered than the number of available UAVs can occur. Therefore, one UAV has to cover multiple regions in a single flight. In an extreme scenario, exactly one UAV can be available to cover multiple regions. Such a scenario is close to the idea of the herein addressed problem formulated in the following chapter.

2.10 Summary of the Existing Coverage Methods

All presented methods solving the single region and the multi-region area coverage problem are for the convenience summarized in the following summarization Table 2.1.

CHAPTER 3

Problem Statement

The following Chapter provides the formulation of the area coverage problem divided into two-phases. The first phase, called the *scanning phase*, can be approached as a single region coverage problem. The single region coverage problem is considered to be solved by the existing literature, and because this work is extending the previous one, the scanning phase is defined the same as in [55]. The taxonomy used is respecting previously established notation. The *collection phase* involves multiple disconnected regions that must be completely covered. The motivation to split the area coverage problem into two phases is to increase the speed and efficiency. We consider that regions to be covered fulfills the following assumptions:

- A1: regions to be covered are convex,
- A2: there is no intersection between obstacles and the regions to be covered.

Many regions in realistic scenarios meet these requirements without any modification, considering the agriculture, many fields with the crops are of roughly rectangular shape. If any region does not meet the requirements, it can always be modified, so it does fulfill them, for example, by omitting the obstacles if the coverage is performed in a sufficient altitude or by extending the shape of the region, so it becomes convex, or using one of the several decomposition methods described in Chapter 2.

The problem of the coverage of multiple regions can be simplified by discretization of each region, denoted $\mathcal{R} = \{r_i, \ldots, r_n\}$, in a way that it can be efficiently covered with respect to the motion constraints of the fixed-wing UAV. Because the UAV is fast and unable to make sharp turns, the number of turns must be minimized. The number of the turns can be minimized by discretization of the regions into a parallel segments S, producing a coverage pattern above the regions similar to the boustrophedon pattern 2.1d,

$$S = \left(\left(s_1^1, s_1^2, \ldots, s_1^{k_1}\right), \ldots, \left(s_n^1, s_n^2, ..., s_n^{k_n}\right)\right), \tag{3.1}$$

where the lower index determines to which region the specific segment belongs and upper index determines the specific segment inside a region. There is a bound k_i for each region r_i, which is equal to the number of segments needed to cover the same specific region r_i. Each such a segment is straight and starts at the border of the region and ends at the opposite border. The segment is defined by its two intersection points with the border of the region, denoted as p_1 and p_2 that are specific for every segment. The segment is considered to be covered if it is entered in one of its intersection points and left in the other one, while the trajectory between the points is a straight line. The segments can be further transformed into coverage cells. For each segment, there are two cells, one for each possible coverage direction determined by the intersection point of the segment that is visited first

$$C = \left(c_1^1, c_1^2, c_2^1, c_2^2, \ldots, c_\phi^1, c_\phi^2\right), \tag{3.2}$$

where the number ϕ represents total number of the segments needed to cover all the regions $\mathcal{R}$ and can be computed as

$$\phi = \sum_1^n k_i = |S| = \frac{|C|}{2}, \tag{3.3}$$

$$k_i = w(r_i, \rho_i), \tag{3.4}$$

where the function w returns the number of segments needed to cover the region r_i in a coverage heading ρ_i, further explained below. The exact number of the segments is influenced by the perception model of the UAV, the altitude of the UAV, the shape of the region, and the selected coverage heading. The camera is positioned perpendicularly to the frame of the UAV (see Figure 3.1), hence, the space between the segments is determined by the size of the footprint f that is equal to the width of the projection of the viewing frustrum on the ground

$$f = 2z \tan\left(\frac{\alpha}{2}\right), \tag{3.5}$$

where z refers to the altitude of the flight of the UAV and α represents the field of the view of the attached camera.

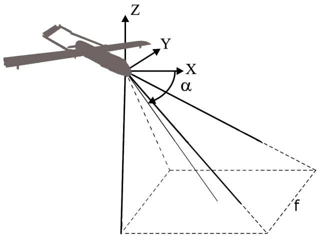


Figure 3.1: Fixed-wing UAVs rectangular on-board camera footprint.

Since there are two cells for each segment, to distinguish which cell is selected in the final solution, the direction of coverage of individual segments must be defined

$$D = (d_1, \ldots, d_\phi), \tag{3.6}$$

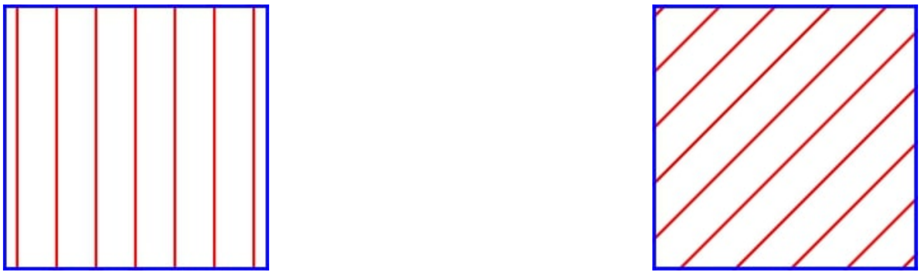
$$d_i \in \{1, 2\}, \qquad i = \{1, \ldots, \phi\}, \tag{3.7}$$

if d_i is equal to 1, then c_i^1 is used, and if d_i is equal to 2, then c_i^2 is used, thus the selected cell can be denoted $c_i^{d_i}$ in general.

Each region r_i can be approached from the arbitrary direction, called coverage heading, and further denoted ρ_i. If a region is covered from different coverage heading, different segments are determined, and the number and size of the individual segments can change, and therefore the final coverage path can be different, see Figure 3.2. Because different coverage heading can be selected for each individual region separately, it is important to define a set P of the selected coverage headings

$$\mathrm{P} = (\rho_1, \dots, \rho_n), \qquad \forall i \in \{1, \dots, n\}, \rho_i \in \langle 0, \pi) , \tag{3.8}$$

where ρ_i represents the selected coverage heading of the region r_i. Because the segments can be visited in two directions, the solutions in the interval $\langle \pi, 2\pi)$ are redundant with those in the interval $\langle 0, \pi)$ and this is the reason that the optimization is bounded by the proposed interval in Equation (3.8). Because the coverage cells are influenced by the selected coverage headings P of the regions $\mathcal{R}$, all the cells can be determined using the function h



(a) An original coverage heading requiring 7 segments.

(b) An example of different coverage heading requiring 10 segments.

Figure 3.2: Example of the two different coverage headings and its influence on the number, position and length of the segments.

$$C = \left(c_1^1, c_1^2, c_2^1, c_2^2, \dots, c_\phi^1, c_\phi^2\right) = h(\mathcal{R}, P). \tag{3.9}$$

Once the coverage cells are determined, it is needed to connect them with turns respecting the motion constraints of the fixed-wing UAV. The used model is a Dubins vehicle model, which determines the shortest trajectory between two points with given initial and final angles of the vehicle. Because it is assumed in this work that the fixed-wing UAV is able to stay in a constant altitude above the ground the Dubins car model is used

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta \\ \sin\theta \\ u_\theta \rho^{-1} \end{bmatrix}, \tag{3.10}$$

$$q \in SE(2). \tag{3.11}$$

The variables x and y represents the location in R^2, θ is representing the heading angle of the vehicle, v is constant speed and $u_\theta \in [-1, 1]$ represents the control input. The use of the Dubins car model produces an unintended problem. The boustrophedon pattern is connecting neighboring cells with each other, but it may not be further possible. If the minimal turning radius of the vehicle is greater than the space between two neighboring cells, the connecting Dubins maneuver contains additional turns, and its length is increased. The task is to find the coverage trajectory such that the length of the trajectory T is minimum, and therefore, the minimum length sequence of visits of the segments Σ must be found

$$\Sigma = (\sigma_1, ..., \sigma_\phi) , \qquad 1 \le \sigma_i \le \phi, i \ne j \implies \sigma_i \ne \sigma_j \tag{3.12}$$

where the σ_i determines next coverage cell c_{σ_i} that will be visited. Therefore, the cells can be visited in an arbitrary order, unlike in a scenario implementing the boustrophedon pattern. The value of a specific σ_i and d_i precisely defines which cell c_i will be visited and its direction of coverage.

The coverage path of the UAV of the multiple regions consists of the visits of the cells and the turns. Both, the flight through the cells and turns can be approached as a maneuver, denoted m. Because the order of cells and turns is consistently switching, the even maneuvers m_{2i} represent the turns determined by the Dubins maneuvers, and the odd maneuvers represent the flight through the cells. The Dubins maneuvers connecting the cells are precisely defined by the sequence of the visits of the cells Σ and the selected directions of the coverage of the cells D.

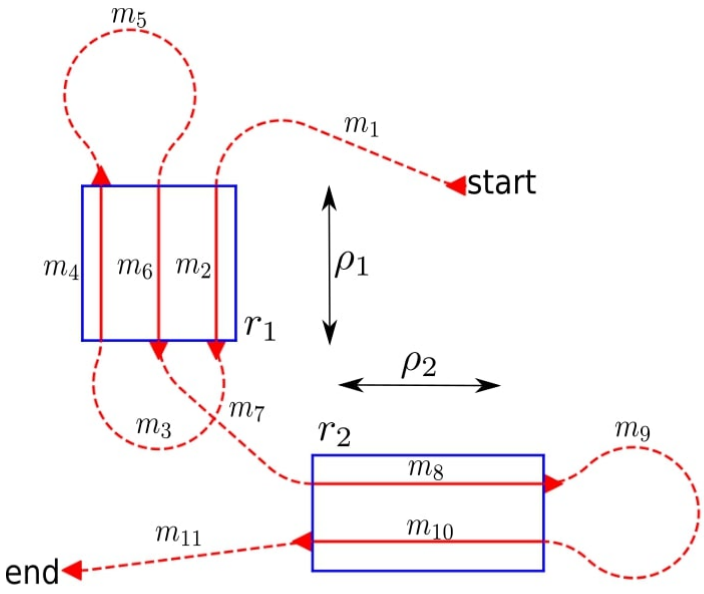


Figure 3.3: An illustration of the problem formulation. Two regions are discretized into 5 segments and 10 coverage cells determined by the coverage headings. The coverage path consists of 5 coverage cells and 11 maneuvers from starting point and the ending point.

Lastly, it is required to define the starting point and the ending point of the multi-region coverage problem. The reason is that the regions to be covered are determined after the scanning phase of the coverage is performed, and therefore, the collection phase is entered from some point and exited at a different point. This issue can be solved by treating the starting and ending point as a part of the first and the last maneuvers. There is only one direction allowed for the starting and the ending point. Therefore, the starting point is equal to the initial configuration q_s and the ending point to the final configuration q_e, which are added to the beginning and to the end of the maneuvers. The problem is transformed into the variant of the TSP by adding a maneuver of the zero cost between the end and the start of the collection phase. Such a maneuver is not considered in the final solution.

Given all mentioned variables, the multi-region coverage problem is treated as a variant of TSP over the possible coverage headings and possible order of visits of individual cells as:

Problem 1 *Multi-region area coverage:*

$$\text{minimize}_{\mathrm{P},\Sigma,D}\ T = \sum_{i=0}^{2\phi+1} \mathcal{L}\left(m_i\right)\ , \tag{3.13}$$

$$\textit{s.t.:}$$

$$C = h(\mathcal{R}, P)\,, \tag{3.14}$$

$$\mathrm{P} = (\rho_1, \ldots, \rho_n)\,, \qquad \forall i \in \{1, \ldots, n\}\,, \rho_i \in \langle 0, \pi)\ , \tag{3.15}$$

$$\Sigma = (\sigma_1, \ldots, \sigma_\phi)\,, \qquad 1 \leq \sigma_i \leq n\,, \tag{3.16}$$

$$D = (d_1, \ldots, d_\phi)\,, \qquad d_i \in \{1, 2\}\,, \tag{3.17}$$

$$m_i(1) = m_{i+1}(0)\,, \qquad \forall i \in \{1, \ldots, \phi\}\ , \tag{3.18}$$

$$m_{2i-1} = c_{\sigma_i}^{d_{\sigma_i}}\ , \qquad \forall i \in \{1, \ldots, \phi\}\ , \tag{3.19}$$

$$q_s = m_1(0)\,, \tag{3.20}$$

$$q_e = m_{2\phi+1}(1). \tag{3.21}$$

The first phase of the area coverage can be viewed as a special case of the multi-region coverage problem. In this case, the starting and the ending point, and corresponding indexes can be omitted. An illustration of the multi-coverage scenario and the problem formulation is depicted in Figure 3.3.

CHAPTER 4

Proposed Solution and Used Methods

The proposed solution to the introduced two-phase area coverage problem is described in this chapter. The scanning phase of the area coverage problem, including the single region only, is considered to be well explored by the existing literature, and various methods are introduced in Section 2.2. Therefore, we are focused on proposing a solution to the collection phase, composing of the multi-region area coverage defined as Problem 1. Moreover, the scanning phase is a special case of the collection phase, and the proposed method can be used to find its solution.

The proposed method uses several existing approaches for solving the TSP. The given problem formulation is treated as a variant of the TSP and uses a modified version of the Noon-Bean transformation [56]. The solution of the actual instance of the TSP is computed with the existing heuristic solver [24]. The individual maneuvers between the coverage cells are modeled as Dubins vehicle maneuvers [15]. The proposed method is considered to be offline in its current state. In addition to the heuristic solution of the TSP-based formulation, we also consider the exact method used for the comparison. Its description is provided in the last part of the chapter.

4.1 Proposed Heuristic Method

4.1.1 Decomposition of the Problem

The multi-region coverage problem defined in the previous chapter is a continuous optimization problem with infinite possible coverage trajectories. The problem can be transformed into a discrete space using fixed-sized steps regarding the optimization of the coverage headings and selecting the individual combinations of the coverage headings for the regions. The number of the possible coverage trajectories is determined by the position and properties of the regions, the starting and ending point of the collection phase, the perception model of the UAV, the maneuver capabilities of the UAV, and the selected coverage headings for the regions. The problem can be modeled as the Generalized TSP (GTSP), which is known to be NP-hard.

The existing exact solvers can find a solution to instances that contain dozens of thousands of vertices in a reasonable time. The approximating solvers can solve instances with a few million vertices with a reasonable threshold of around $1 - 2\%$ above the optimum [57].

The TSP complexity grows with the number of vertices, and in the multi-region problem, each vertex represents one direction of the segment. The number of segments to be covered is large for even small multi-region coverage scenarios. The number of the needed segments to cover one region k_i can be calculated with the use of the diameter function $d(\rho_i, r_i)$, which returns the width of a specific region r_i for a given coverage heading ρ_i and the size of the footprint of the UAV f

$$k_i = w(r_i, \rho_i) = \frac{d\left(\rho_i, r_i\right)}{f}. \tag{4.1}$$

Each segment can be visited from one of two opposite directions. Each direction is modeled as a cell, and therefore, the number of cells is doubled compared to the number of the segments. Each such a cell is equal to a vertex in a graph representing the corresponding TSP. There are two options to approach the multi-region coverage problem. The first option is to precompute all the cells for all the coverage headings and then find the solution at once. This option has two disadvantages. The first one is that more constraints must be added to prohibit all the cells from the different coverage headings of a specific region once the cell is selected into a solution. It must be carefully checked; otherwise, the final solution could contain the coverage trajectory, which would use multiple coverage headings to cover an individual region. Therefore, it would violate the condition to avoid unnecessary overlaps. The second disadvantage is that such a state space is huge. The total number of cells $|C|$ considered in a multi-region coverage scenario can be computed as

$$|C| = 2 \sum_{i=1}^{\lceil B/\Delta\rho \rceil} \sum_{j=1}^{|\mathcal{R}|} \left(\frac{d((i-1)\Delta\rho, r_j)}{f} \right) + 2, \tag{4.2}$$

where $\Delta\rho$ represents the step of the coverage heading, $|\mathcal{R}|$ the total number of the regions, and B is the bound till which the coverage headings are evaluated. If the regions of general shapes are present, B is 180° because the solutions are symmetric, and $\Delta\rho$ is selected to be reasonably small. To give the idea, if only three small regions are considered with only 30 passes needed for each to be covered and $\Delta\rho$ is 1°, the number of vertices to consider is $32,402$. In this case, the growth of the vertices is bounded by $\mathcal{O}\left(\left\lceil \frac{B}{\Delta\rho} \right\rceil |\mathcal{R}|\ k\right)$, where k represents an average number of segments needed to cover all the regions with respect to the selected combination of the coverage headings.

The second option to approach the multi-region coverage problem is to decompose it into the individual combinations composed of the specific coverage headings selected for each region. Therefore, the solution to every separate combination can be found and compared to the best one so far. The advantage is that smaller instances will be solved. In the specific example above, each instance would contain only about 182 vertices. The drawback of this approach is that for each region, the specific coverage headings will repeat many times because this method evaluates overall combinations given by the coverage step and the bound separately. Therefore, to find the same solution as in the example above, it takes $5,823,000$ steps.

The presented method selects the second option - the decomposition into individual combinations. In this case, the number of the solutions to be compared can be bounded by

$\mathcal{O}\left(\left(\left\lceil \frac{B}{\Delta\rho} \right\rceil\right)^{|\mathcal{R}|}\right)$. The reason for selecting this approach is that it can be improved a lot, and the number of combinations can be drastically decreased. Usually, it is not necessary to compare the solution whose selected coverage headings differ by just one degree. Therefore, a more considerable step can be selected to decrease the number of iterations. Moreover, the second approach allows the distribution of the problem among the CPUs, and the real computation time can be further decreased. Finally, the last reason for this choice is that the approach allows adding heuristic optimization, e.g., based on the results provided in the last chapter of this work.

4.1.2 Determination of Coverage Cells

The previous section was devoted to the problem's size and the decomposition of the problem into the individual combinations determined by the specific coverage headings. The illustrative example given in the previous section assumed the constant number of coverage cells for each region and the coverage heading. In a real scenario, this does not hold, and the number of coverage cells does not stay constant for every combination of the headings. The number of coverage cells for the given region is determined by the size of the footprint f, and the value of diameter function d for specific coverage heading, as defined by equation (4.1.1).

The number of the coverage cells can differ with respect to the selected coverage headings because in some cases, the coverage cells on the boundary of the region must be added to avoid some of its parts would not be covered, even though part of the footprint of the fixed-wing UAV will capture the area outside of the region, see Figure 4.1. Moreover, as can be seen, the coverage cells must be prolonged outside of the region to avoid skipping coverage of its parts. This is required due to the perception model introduced in Section 2.1. It is assumed that the camera is attached perpendicularly to the UAV, and therefore, the footprint of the UAV is rectangular. It allows reducing the model to the width of the footprint (3.5) with respect to the direction of the flight. This assumption enforces the prolongation of the cells and simplifies future image processing and region reconstruction. The length of the individual cells is given by the intersection of the boundary of the area and the UAV's footprint. The exact position of the cells can be determined by placing the cells perpendicularly to the selected coverage heading from the center of the region towards its sides. The cells representing the same segment but different direction are overlapping. The total number of the cells for one specific

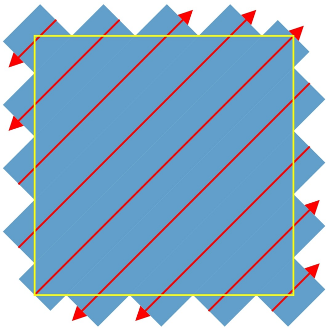


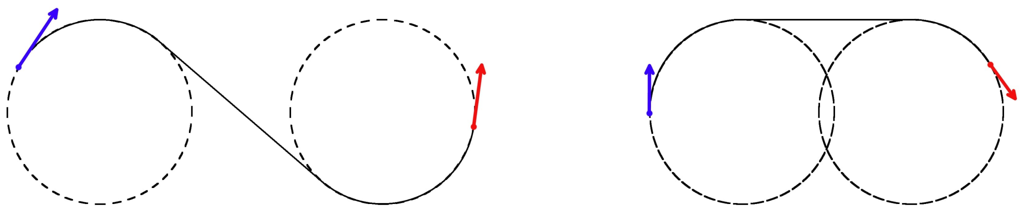
Figure 4.1: An example of the configuration of the region which requires to add the boundary cells. The cells are depicted as the rectangular areas colored in blue, the trajectory of the fixed-wing UAV inside the cell is colored in red and the maneuvers outside the cells are colored in darker blue. The boundary of the region is colored in yellow.

combination of the coverage headings is determined by

$$|C| = 2\sum_{1}^{n} k_i = 2\sum_{i=1}^{n} \frac{d\left(\rho_i, r_i\right)}{f}. \tag{4.3}$$

4.1.3 Connection of the Cells

Once the coverage cells are determined, the next step is to find the connecting maneuvers between the cells. The lengths of the individual connecting maneuvers form the distance matrix **D** that is used to determine the coverage trajectory. As introduced in Chapter 3, the connections of the individual cells are modeled as Dubins maneuvers (3.11). The Dubins maneuvers model feasible trajectory for the fixed-wing UAV because the motion of the fixed-wing UAV in a plane can be determined from its velocity and the minimum turning radius. Except for the minimal turning radius and the UAV's velocity, it is required to know the position of the two points to be connected and the initial and final heading at these points. In the case of the proposed method, the initial and final points and the related headings are determined by the position of the cells. It is proven that the shortest path following the Dubins model consists of the arcs of maximal curvature, i.e., the arcs matches the minimum turning radius of the vehicle and straight-line segments [15]. The arcs are denoted according to its orientation with respect to the motion, i.e., left-turning arc is denoted L and right turning arc R. The straight-line segments connecting the arcs are denoted S. The optimal path between two points with given headings is always one of six possible types: RSR, RSL, LSR, LSL, RLR, LRL. Examples are depicted in Figure 4.2.



(a) An example of the Dubins maneuver of type RSL. **(b)** An example of the Dubins maneuver of type RSR.

Figure 4.2: Examples of the Dubins maneuvers.

Table 4.1: An example of the distance matrix **D**. The values in the table represent the distance from the row vertex to the column vertex. The symbol "-" represents non existing edges.

	c_1^1	c_1^2	c_2^1	c_2^2	s	e
c_1^1	-	-	$\mathcal{L}(m_1)$	$\mathcal{L}(m_3)$	-	$\mathcal{L}(m_{12})$
c_1^2	-	-	$\mathcal{L}(m_6)$	$\mathcal{L}(m_8)$	-	$\mathcal{L}(m_{11})$
c_2^1	$\mathcal{L}(m_2)$	$\mathcal{L}(m_5)$	-	-	-	$\mathcal{L}(m_{15})$
c_2^2	$\mathcal{L}(m_4)$	$\mathcal{L}(m_7)$	-	-	-	$\mathcal{L}(m_{16})$
s	$\mathcal{L}(m_{10})$	$\mathcal{L}(m_9)$	$\mathcal{L}(m_{13})$	$\mathcal{L}(m_{14})$	-	-
e	-	-	-	-	0	-

The distance matrix **D** is computed from the Dubins maneuvers between all the cells and the starting and the ending point of the collection phase. The starting point s and the ending point e can be represented by the zero-length cell and can be added at the end of the distance matrix. The distance matrix for the example situation in Figure 4.3, is depicted in Table 4.1. As can be seen in the example, an additional edge connecting the starting and the ending point must be added into the matrix **D** to achieve an instance of the TSP. The feasible solution should leave the starting point, visit all the coverage cells, and return to the starting point through a virtual edge with zero-length from end to start. Moreover, each coverage cell can be visited from one of its two coverage headings that are modeled as two separate cells (vertices). It can be seen in the matrix **D**, where one direction is denoted c_i^1, and the opposite direction in the same coverage cell is denoted c_i^2.

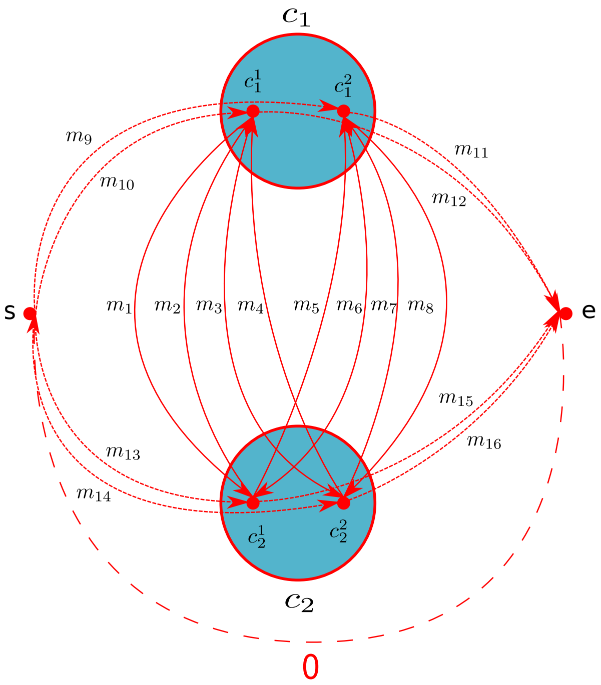


Figure 4.3: An example of the possible connections in a scenario with only two coverage cells c_1, c_2 and the start s and end e.

4.1.4 Individual Solution

Let the coverage cells be determined, and the distance matrix **D** pre-computed, the multi-region problem can be modeled as the GTSP, which is also known as the set TSP. The GTSP is a variant of the TSP in which the final path must visit all clusters – subsets of disjoint vertices.

In the multi-region problem, individual clusters are composed of the coverage cells representing the two possible directions. If one of the two cells is selected in a solution, the other cell is considered to be visited. The vehicle starting and ending points can be visited only in a single direction. Both points are added at the end of the distance matrix, and it is required to visit both of them. Therefore, the instance comprises the coverage cells organized into clusters and two extra clusters of the size one containing only the starting and the ending point.

The proposed method leverages the existing solver to the TSP, the Helsgaun's implementation of the Lin-Kernighan heuristic (LKH) [24]. The LKH solver [58] can find the solution to the TSP or Asymmetric TSP (ATSP), which is optimal or close to the optimum. The distance matrix is symmetric for the TSP; thus, the distance between vertices a and b is the same as the distance from b to a, while in the ATSP, these two distances can be different. An instance of the GTSP must be transformed into the TSP or ATSP to utilized the LKH solver that can be achieved by the Noon-Bean Transformation [56], which transforms the GTSP into the ATSP.

The main idea of the Noon-Bean transformation is to add zero-length edges between the vertices belonging into the same cluster and adding a substantial constant M to the edges leading between the vertices that do not belong to the same cluster. Correspondingly to these changes, the edges between the clusters must be edited as well. The transformation of the example depicted in Figure 4.3 is shown in Figure 4.4 and the corresponding matrix **N** is displayed in Table 4.4.

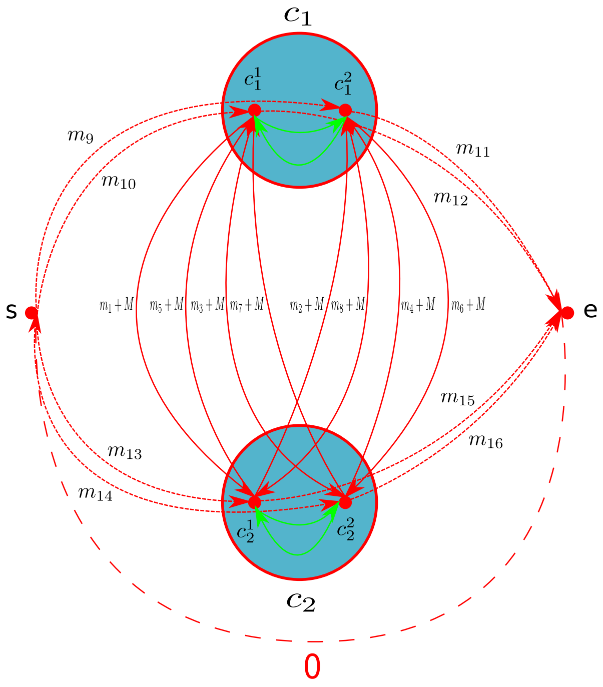


Figure 4.4: An example of the Noon-Bean transformation of the instance shown in Figure 4.3.

Table 4.2: An example of the Noon Bean transformation matrix **N** of the distance matrix **D** 4.1.

	c_1^1	c_1^2	c_2^1	c_2^2	s	e
c_1^1	-	0	$\mathcal{L}(m_3)+M$	$\mathcal{L}(m_1)+M$	-	$\mathcal{L}(m_{12})$
c_1^2	0	-	$\mathcal{L}(m_8)+M$	$\mathcal{L}(m_6)+M$	-	$\mathcal{L}(m_{11})$
c_2^1	$\mathcal{L}(m_5)+M$	$\mathcal{L}(m_2)+M$	-	0	-	$\mathcal{L}(m_{15})$
c_2^2	$\mathcal{L}(m_7)+M$	$\mathcal{L}(m_4)+M$	0	-	-	$\mathcal{L}(m_{16})$
s	$\mathcal{L}(m_{10})$	$\mathcal{L}(m_9)$	$\mathcal{L}(m_{13})$	$\mathcal{L}(m_{14})$	-	-
e	-	-	-	-	0	-

As can be seen in the matrix **N**, the transformation needs to be applied to the coverage cells without the start and end points. Moreover, the values inside the individual clusters of the matrix **N** are circle shifted to the right. The new distance matrix **N** is used to compute the final solution of the single instance of the multi-region problem with the LKH solver.

4.1.5 Algorithm for Multi-region Area Coverage Problem

Based on the presented methods, the multi-region area coverage algorithm is overviewed in Algorithm 1.

Algorithm 1: Multi-region coverage algorithm

```
Data: Regions $\mathcal{R}$, coverage step $\Delta\rho$, coverage bound $B$, minimum turning radius $t$,
      footprint width $f$
Result: Shortest tour $T_s$, cost of the tour $C_s$
Initialization;
for $p \in$ P do
    Compute all the coverage cells $C$ from $p$;
    Compute $\mathbf{D}$;
    $\mathbf{N} \leftarrow$ noon-bean transformation of $\mathbf{D}$ ;
    Solve LKH for $\mathbf{N}$;
    Reconstruct the tour $T$ and its cost $c$;
    if $c \leq C_s$ then
        $T_s \leftarrow T$;
        $C_s \leftarrow c$;
```

The proposed algorithm can be further extended to prohibit the re-entries between the regions. However, such improvements are considered out of the scope and are dedicated to future work.

4.2 Exact Method

In addition to the LKH-based solution, which is known to be non-optimal, a reference solution is desirable to compare the quality of the proposed method. Therefore, we consider the reference solution on top of the same structure as the proposed method, due to the lack of any other existing method in the literature. Instead of a heuristic solution, we solve the related ATP using Integer Linear Programming (ILP) solved by the CPLEX solver [59].

The ILP formulation of the TSP was proposed by Dantzig et al. [60] that has been later shown to solve the ATSP [61], and the formulation is known as Dantzig-Fulkerson-Jefferson formulation (DFJ). The DFJ formulation introduces a binary variable $x_{ij}, i, j \in \{1, \ldots, 2\phi + 2\}$ that is set to 1 only if the edge between the cells i and j is a part of the solution; otherwise, it is set to 0. The input to the solver is the cost matrix; in our case, the matrix **N**. The objective function minimizes the value of the selected edges with respect to the provided cost matrix; therefore, there must exist one variable for every edge. The original formulation had the following structure:

$$\min \sum_{i=1}^{2\phi+2} \sum_{j=1}^{2\phi+2} \mathbf{N}_{ij} x_{ij}, \qquad x_{ij} \in \{0, 1\}\,, \tag{4.4}$$

$$\sum_{i=1}^{2\phi+2} x_{ij} = 1, \qquad j = 1, \ldots, 2\phi+2, \tag{4.5}$$

$$\sum_{i=1}^{2\phi+2} x_{ij} = 1, \qquad i = 1, \ldots, 2\phi+2, \tag{4.6}$$

$$\sum_{i=1}^{2\phi+2} x_{ii} = 0, \tag{4.7}$$

$$\sum_{i\in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1, \qquad \forall Q \subset \{1, \ldots, 2\phi+2\}\,, |Q| \geq 2 \tag{4.8}$$

The first two constraints (4.5) and (4.6) ensure that there is exactly one incoming and one outgoing edge to every cell in the valid solution. The constraint (4.7) prohibits to visit the same cell, i.e., the valid solution cannot contain an edge from the cell and back to itself. The last constraint (4.8) prohibits all the sub-tours in the final solution. The last constraint is added lazily to decrease the number of constraints and speed up the solver. Thus, if the solution is found, it is checked whether it contains any sub-tours, and if it does, all specific values representing them are prohibited by adding the respective constraints. The solver is then called again to provide a new solution with respect to the new constraints. The implementation of the original ILP formulation was too slow, and therefore the following constraint was added to improve its performance.

$$x_{ij} + x_{ji} \leq 1 \qquad \forall i, j \in \{1, \ldots, 2\phi+2\} \tag{4.9}$$

Its purpose is to prohibit all the sub-tours of the length equal to 2, and thus speed up the solver at the beginning by avoiding many of these invalid solutions.

4.3 Implementation

The proposed method and the reference approach have been implemented in the Julia language, version 1.5.0. [62]. The formulated instance of the GTSP is transformed into the ATSP that is solved by the LKH solver [58]. The third-party solver CPLEX [59] with an academic license has been used to compute the solution of the ILP formulation. The complexity of the individual solutions computed by the LKH can be approximate by $\mathcal{O}(n^{2.2})$ [24].

CHAPTER 5

Results

In the presented results, we aim to show the situations that benefit from the allowed re-entries between the regions and minimize the final coverage trajectory length. The first section is devoted to studying regions' distribution influence on the coverage path length and the re-entries. The second section is devoted to showing the turning radius on the re-entries between two regions of fixed size and position. The results achieved by the proposed method are evaluated with the use of the ILP-based solution, as described in Section 4.2. All the theoretical results were achieved in the computational environment with the 2,5 GHz Quad-Core Intel Core i7 and 16 GB DDR3 RAM. The used operating system was macOS Catalina.

5.1 Influence of the Distribution of the Demanding Areas

The advantage of the trajectories with the re-entries is shown in a scenario with two regions with increasing distance. The motion constraints used for all the computations are the same as those of the proposed testing platform depicted in Table 6.1. The regions have a rectangular shape with the size 500 m×500 m. The initial distance between the regions is 50 m and is extending with a step of 5 m up to the final distance 400 m. The starting and the ending points of the collection phase are always in the same distance from the regions; thus, its influence is neutralized. The first path of the evaluation is devoted to determining the best combination of the coverage headings for both methods. The second part of the evaluation provides a comparison of the solutions provided by different solvers. Finally, the last part compares the trajectories with the re-entries and the trajectories prohibiting them.

5.1.1 Determination of Coverage Headings

The influence of the distribution of the regions is compared for the best combination of the coverage headings determined by the proposed method and the method prohibiting the re-entries. The coverage headings are evaluated with the step size 2° for the initial setup of the regions. The rest of the examination compares trajectories with a preserved combination of the coverage headings, and only the distance of the regions is changing with the position of the starting and the ending points. Because only two regions are included in the scenario, the results are displayed in the form of the heading contours. The four contours are provided because the heuristic and exact solutions are found. The results are depicted in Figure 5.1.

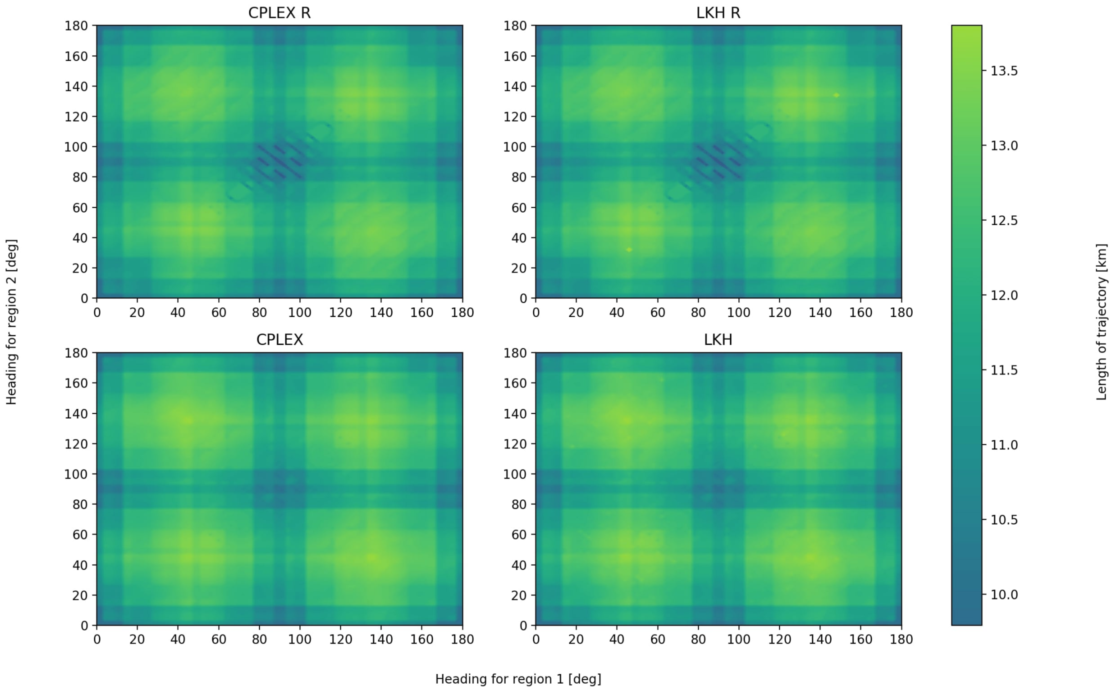


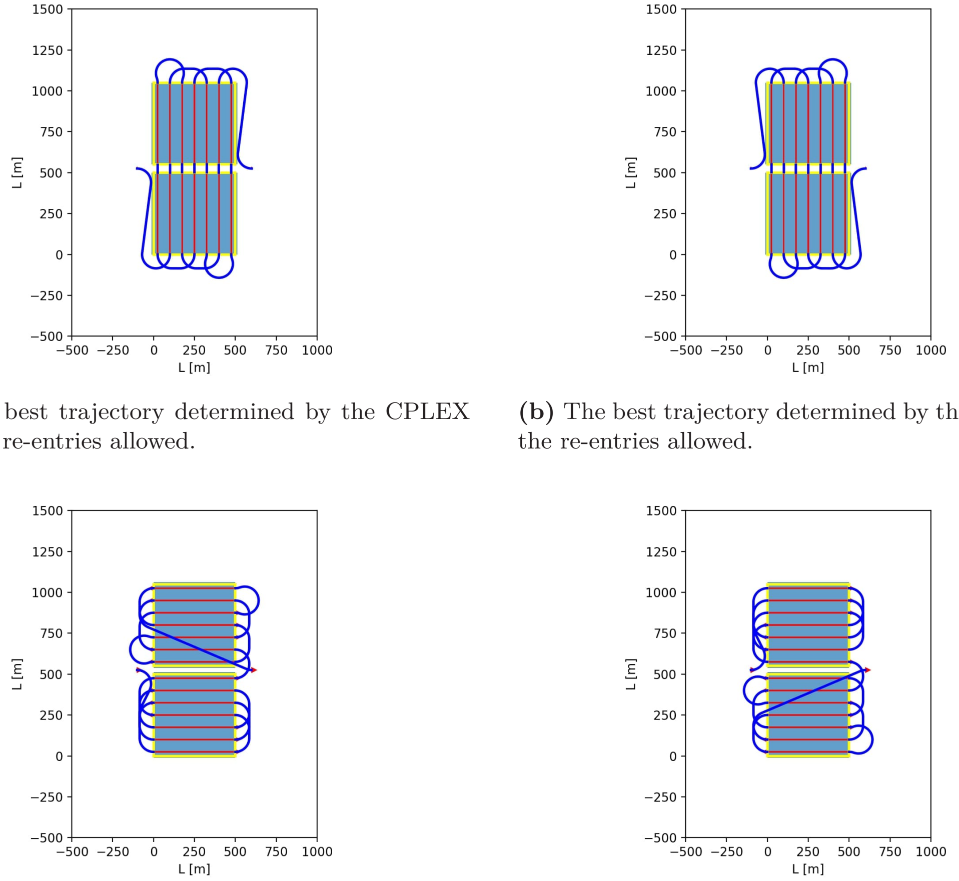
Figure 5.1: The heading contours displaying the lengths of the final coverage trajectories of two regions. The results of the trajectories with the re-entries are labeled as R.

It can be seen from the presented results that all the heading contours are similar and divided into bright areas containing long coverage trajectories and dark areas marking the best combination of the coverage headings. The darkest areas in the contour occur every time the coverage headings are parallel or close to being parallel to the sides of the regions. Such a result corresponds with the expectation that the shortest trajectory will contain the lowest number of the segments. So, the coverage heading of the region should be parallel to one of its sides. It is evident in an environment containing the rectangular regions, but if the regions of more complicated shapes are present, the final coverage trajectory can select different coverage heading to achieve the shortest length, which is reported later in the next chapter. The best solutions found are depicted in Figure 5.2.

The method containing the re-entries selected a different combination of the coverage headings, and it contains headings $(90, 90)$. On the other hand, a method with prohibited re-entries composes of headings $(0, 0)$. In case of the allowed re-entries, it is worth to go around the region from the starting point, because the two regions can be treated as one connected region. Both combinations of the headings are selected for further evaluation.

5.1.2 Comparison of the Solvers

The results shown in the previous section are examined from the utilized solver point of view. The solutions provided by the CPLEX solver should be optimal with respect to the given



(a) The best trajectory determined by the CPLEX with the re-entries allowed.

(b) The best trajectory determined by the LKH with the re-entries allowed.

(c) The best trajectory determined by the CPLEX with the re-entries prohibited.

(d) The best trajectory determined by the LKH with the re-entries prohibited.

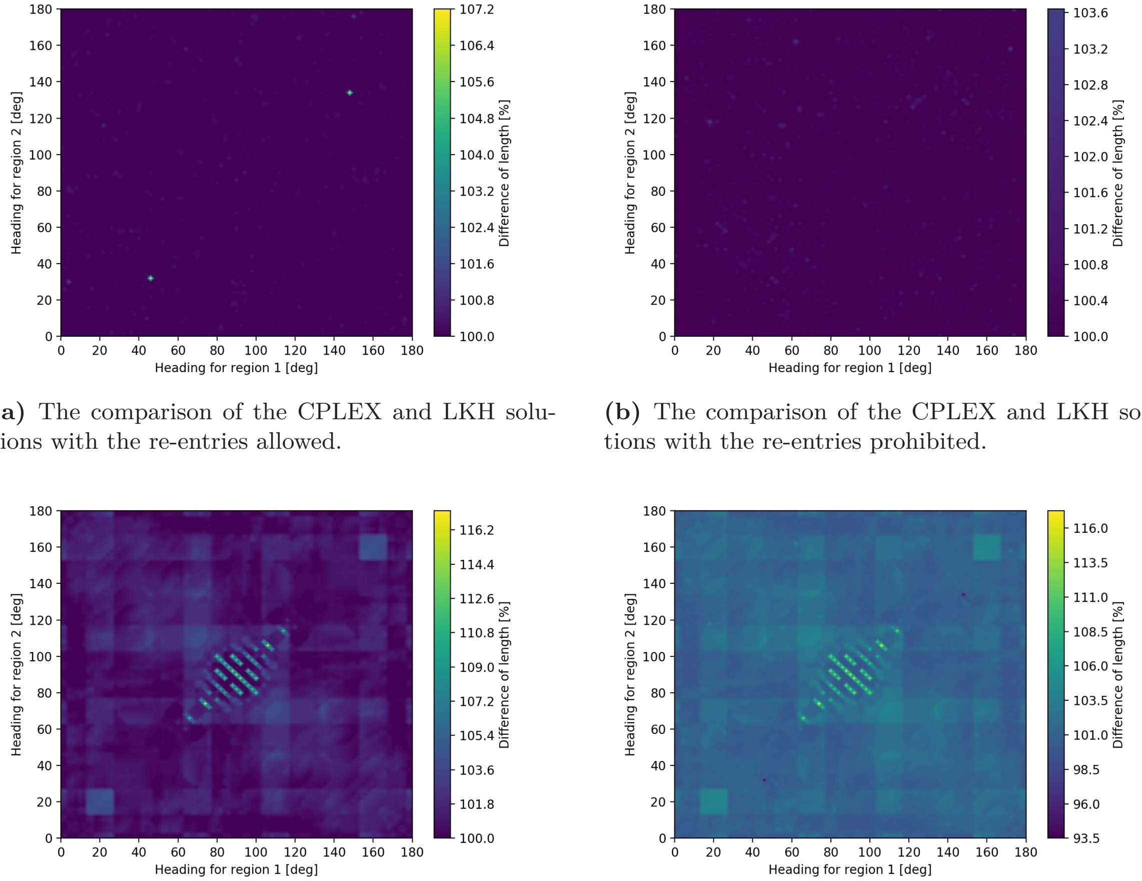
Figure 5.2: The best solutions determined by the used methods and solvers.

ILP formulation. Therefore it should be better or the same as the solutions provided by the LKH solver. Differences between the solutions found by the individual solvers are shown by comparing the heading contours depicted in Figure 5.1 with each other again. Firstly, the solutions with the re-entries are compared in Figure 5.3a, where the computed solutions mostly have the same length except for two very bright areas the LKH solver provides significantly worse solution than ILP.

The results for the prohibited re-entries are shown in Figure 5.3b. As before, the solutions provided by the LKH are sometimes worse than the optimal ILP-based solutions found by the CPLEX. However, in this case, there are no bright spikes. The solutions provided by the LKH are close to optimum provided by the CPLEX.

The results for the ILP formulation with re-entries prohibited and allowed are depicted in Figure 5.3c, where we can see solutions with the prohibited re-entries are much worse than solutions with allowed re-entries. The improvements by the re-entries are about 16 percentage points for the specific combinations of the headings close to $(90, 90)$.

A similar result is achieved if the solutions computed by the LKH solvers are compared, as it can be seen in Figure 5.3d. In the case of some combinations of coverage headings, the solutions with re-entries are worse than those without them. It happens only when the LKH solver is used because it is known to be not optimal. In most of the cases, the solutions with the re-entries are better than those without the re-entries. However, few solutions are worse because of the properties of the utilized LKH solver.



(a) The comparison of the CPLEX and LKH solutions with the re-entries allowed.

(b) The comparison of the CPLEX and LKH solutions with the re-entries prohibited.

(c) The comparison of the CPLEX with the re-entries prohibited and CPLEX with re-entries allowed.

(d) The comparison of the LKH with the re-entries prohibited and LKH with re-entries allowed.

Figure 5.3: The comparison of the solvers and their solutions.

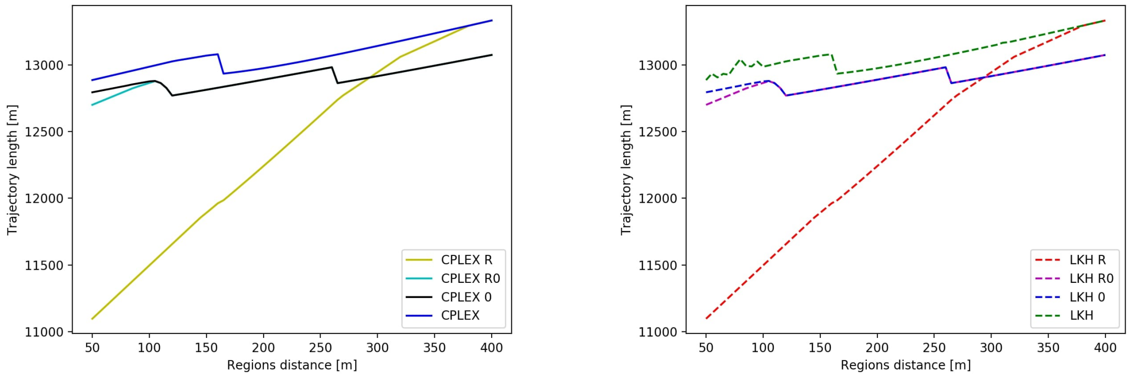
The main advantage of the LKH solver is in the smaller computing requirements. The best, worse, and average lengths of the solutions provided by the individual algorithms are presented in Table 5.1 together with the computational time.

5.1.3 Influence of the Distance between the Regions

We examine the influence of the distance between the regions for methods with and without re-entries. Both determined combinations of coverage headings are evaluated, and all the results are displayed in Figure 5.4. For clarity, the results produced by the CPLEX and LKH solvers are separated into individual figures. The R0 is denoted to the combination of the coverage headings consisting of $(0, 0)$ with re-entries, and just 0 for the case the re-entries are prohibited.

Table 5.1: Performance of the solvers and computed lengths of the solutions.

	Best [m]	Worst [m]	Average [m]	Time [s]
CPLEX R	11096	16566	14932	4689
LKH R	11096	17270	14935	1838
CPLEX	12793	16790	15096	9147
LKH	12793	16797	15103	3420



(a) Evaluation of the distribution of the regions and it's influence on the trajectory length in case the CPLEX solver is used.

(b) Evaluation of the distribution of the regions and it's influence on the trajectory length in case the LKH solver is used.

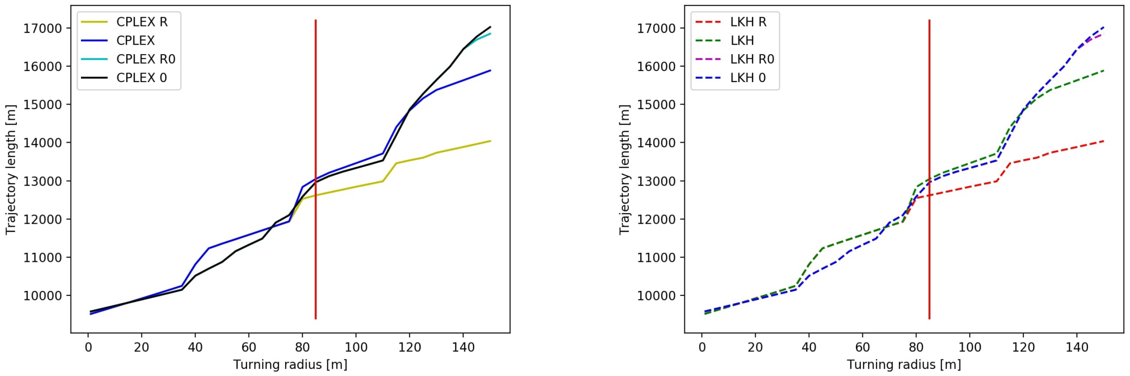
Figure 5.4: Evaluation of the distribution of the regions.

For the allowed re-entries, the best combination of the coverage headings shortens the trajectory length. The effect is slowly diminishing with the increasing distance between the regions. For sufficiently distant regions, re-entries are not exploited, and the effect diminishes around the distance between the regions equal to approximately 280 m. In our case, it is more than three times of the turning radius of the used fixed-wing UAV model. The results also indicate that solvers provide very similar solutions, and thus, lower computational requirements can be preferred.

5.2 Influence of the Turning Radius on the Multiple Visits

The proximity of the regions is not the only factor influencing the length of the final coverage trajectory. The number and advantage of the re-entries of the regions are also influenced by the maneuvering capabilities of the used fixed-wing UAV. We consider a similar testing scenario as in the previous case to study the evaluation of the influenced UAV on the multi-region coverage problem. A similar scenario, as in the above example, is utilized; however, instead of changing the distance of the regions, a turning radius of the UAV is changing in the range 1 m to 150 m. The distance between the regions is 250 m. The evaluation results are depicted in Figure 5.5 for both LKH and CPLEX solvers and with and without re-entries.

The results indicate that in the case of a small turning radius, the lengths of the solutions



(a) The influence of the turning radius of the fixed-wing UAV on the length of the coverage trajectory computed by the CPLEX.

(b) The influence of the turning radius of the fixed-wing UAV on the length of the coverage trajectory computed by the LKH.

Figure 5.5: The influence of the turning radius of the fixed-wing UAV on the length of the coverage trajectory.

provided by the particular methods and solvers overlap. The important point in our case is that the turning radius equal to 85 m, which is the recommended turning radius of the proposed fixed-wing UAV, depicted in Figure 5.5 as the red vertical line. Slightly before this value is achieved and since then, the trajectories produced by the proposed method are shorter than the trajectories without the re-entries. It can be explained that if the turning radius is small, the UAV can turn and connect to the neighboring cell without the need to use any re-entries between the regions that are simply too far. Once the turning radius is increased enough, the UAV cannot connect to the neighboring cell, and the trajectory search is forced to find some sequence of visits skipping the cells and visiting them later. From some point, it is more efficient to move to the different region and return to finish the coverage later than staying in the same region, because to finish the coverage of the current region forces the UAV to add additional turns to maneuver back to the not visited cells. Another advantage of the proposed method is that since it minimizes the number of turns, the fixed-wing UAV can maintain its heading and speed more often and perform the coverage with the increased average speed.

CHAPTER 6

Real Deployment

The proposed method for solving the multi-region coverage problem has been further considered for an experimental deployment using real UAV. In this chapter, we provide a description of the proposed testing platform and the real deployment results. The proposed testing platform consists of a fixed-wing UAV and corresponding ground station. The real deployment is evaluated in three different scenarios, each with increased complexity with respect to the safety. The results of the real deployment are compared with the pre-computed results described in Section 6.2.

6.1 Testing Platform

The fixed-wing UAV to be used for the experimenting is the Maja UAV, depicted in Figure 6.1. The Maja UAV provides sufficient features for the research and experimenting, mainly because the operation time is up to 30 minutes. Besides, it has an ample storage space inside its hull for the equipment and sensors, and it is supported officially by its manufacturer Bormatec company to be controlled with the Pixhawk controller and PX4 autopilot. The practical advantage is also in easy dismantle and transportation to any site.

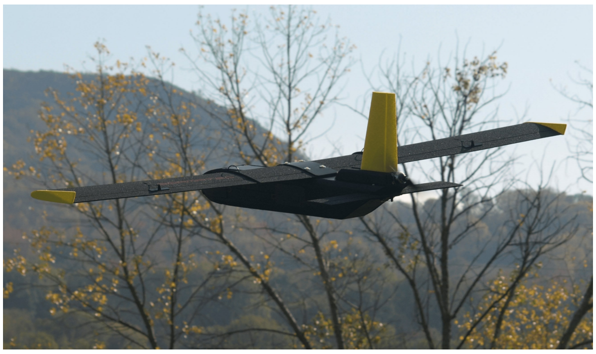


Figure 6.1: A proposed testing platform - Maja.

Few test flights need to be performed before the evaluation in the testing scenarios to identify the limitations of the Maja that are important for autonomous control. The performed

Table 6.1: Summary table of the properties of the Maja UAV.

Equipment	Recommended
Wingspan	220 cm
Length	120 cm
Empty weight	2.5 kg
Max. weight	3.0 kg
Max. cargo weight	0.5 kg
Cargo space	35 cm × 10 cm
Max. flight time	0.3 h
Min. flight speed	$10\,\mathrm{m\,s^{-1}}$
Cruise flight speed	$15\,\mathrm{m\,s^{-1}}$
Min. turning radius	50 m
Rec. turning radius	85 m
Max. pitch	25°
Min. pitch	25°
Max. roll	30°
Max. flight distance	16.2 km

tests were mainly concerned by the safety of autonomous deployments, and therefore, the flight properties were identified to get the idea about its controllability and overall performance. However, the actual testing flights were controlled by a human pilot on the ground. The identified properties are depicted in Table 6.1.

Minimal turning radius – The first important measured property is the minimum turning radius of the UAV because it is needed for the computations of the Dubins maneuvers and the actual coverage trajectory. The minimal turning radius was measured as an absolute minimum. However, during the actual deployment, it should be intentionally increased at least by a factor of 1.5 to prevent the vehicle from sharp turns during which it has problems to keep the altitude. The Maja vehicle is equipped with large wings of the total wingspan 2.2 m, and the size of the hull and a rudder in comparison with the wingspan prevent the drone from keeping the same altitude if the maneuver is too tight. If the minimum radius is increased, it gives the space to a controller to increase the curvature of the turn if needed, for example, due to the weather conditions like the changing wind or vertical movement of the air caused by the changes of the temperature.

Pitch angle limitations – The maximum and minimum pitch was fixed on the values presented in Table 6.1, and these values do not represent the limits of the drone. The selected values allow the UAV to stay agile enough to stay in the fixed altitude above the ground in the testing scenarios and simultaneously prevent the controller from making too violent adjustments that could lead to the destruction of the wing of the UAV.

Fligt time – The real measured maximum flight time 0.3 h is lower than the expected 0.5 h. It is because of the additional onboard equipment that consumed additional power. The flight was also performed at a lower altitude than the nominal altitude. Nevertheless, mostly the reason was the human pilot with a relatively high number of altitude and heading corrections, which further increase power consumption.

Flight distance and cruise speed – The maximum flight distance was computed based on the cruise speed in horizontal flight and the measured time flight but can be influenced by the weather conditions. During the experimental realizations, the flight distance should be intentionally decreased to prevent any problems caused by the loss of energy. The cruise speed was oscillating between $15\,\mathrm{m\,s^{-1}}$ and $20\,\mathrm{m\,s^{-1}}$, and for safety reasons, it was decided to use the first one for the computations and realization of the experiments.

Flight control – The Maja UAV is controlled by the Pixhawk 4 board with the PX4 autopilot stack. The PX4 autopilot was selected because it is officially supported by the manufacturer of the Maja, and therefore, its model can be easily uploaded into the board. Furthermore, a recent version of the PX4 contains built-in functions to precisely plan and navigate the fixed-wing UAVs through simple waypoints that can be uploaded into the autopilot via the ground station installed on the computer. Therefore, the coverage trajectory can be transferred into the waypoints and uploaded into the autopilot, and the PX4 autopilot can handle the flight and navigation itself. The recent version of the PX4 plans the trajectory with respect to the uploaded waypoints. It can predict and determine the necessary adjustments of the flight to keep the smooth trajectory through all the waypoints. It is a big step from the previous versions, which navigated the UAV to the current waypoint, and if the waypoint was a way of the current direction of the flight, the UAV mostly missed it, and the autopilot has been then forced to return to it. Thus, such an autopilot is not suitable for the trajectory following missions like the area coverage; however, the used version of the PX4 provides satisfactory performance.

Flight safety – The safety of the flight in the autonomous mode is maintained with the geofence and all-the-time radio contact, allowing to regain the manual control by the pilot on the ground. The geofence system allows creating a boundary around the expected area of the autonomous flights. If the UAV intersects it, it is forced to cancel the actual mission and return to the home position. All the systems and flight modes were carefully tested.

6.2 Real Deployment in Experimental Scenarios

In this section, we present three experimental scenarios to test the proposed platform in a real deployment. The proposed scenarios have different complexity and number of the regions to be covered. All the scenarios were designed in such a way that the safety of the autonomous flights is ensured, and all-the-time visual contact of the pilot on the ground could be maintained. Each scenario is defined by the GPS coordinates and the corresponding coordinates in the XY plane with respect to the reference point used for the actual computations. All the necessary coordinates are contained in the tables summarized in the appendices of this work, see Appendix A. The coverage trajectory for each scenario was firstly computed with the proposed method. The computed trajectory was tested in the software in the loop to ensure the feasibility of the trajectory for the testing platform. After that, each trajectory was transformed into the waypoints for the UAV and tested on a real flight. The altitude of the flights was determined to be 100 m above the ground in all scenarios. Because of the determined altitude, the distance of the coverage segments from each other was 75 m.

6.2.1 Experimental Scenario 1

The first proposed experimental scenario, depicted in Figure 6.2 contains one region. The position of the real region and a real start and end of the coverage was designed to enable the all-the-time visual contact with the UAV and to maintain safety while testing the flight control system. It was decided to contain a single region coverage scenario to test the coverage in an autonomous flight and to see the difference between the real and computed trajectories before more complex scenarios were deployed. The length of the computed coverage path is 5609 m, and the solutions computed by the LKH and CPLEX solvers are identical. The computed coverage trajectory is depicted in Figure 6.3a, and the comparison of the captured trajectory from a real flight and the computed trajectory is depicted in Figure 6.3b.

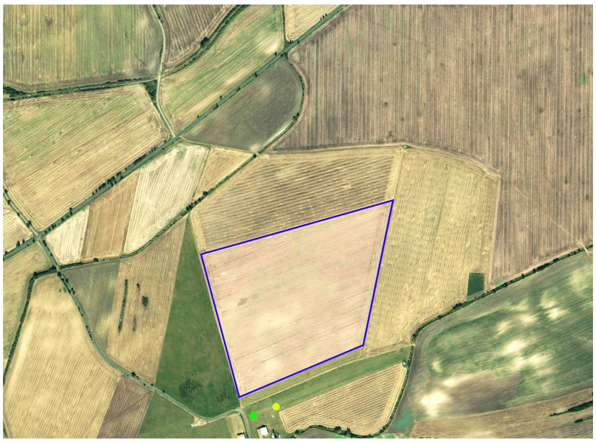
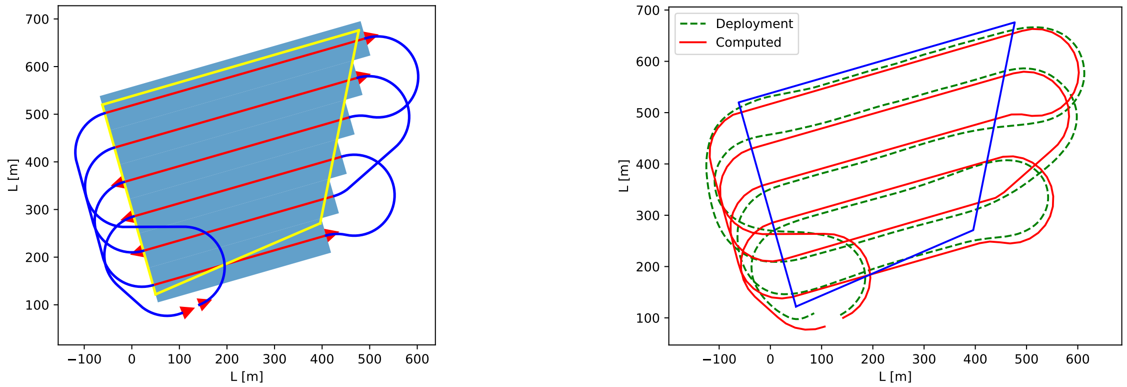


Figure 6.2: The first testing scenario. The region to be covered (blue polygon) represents one field that is close to the starting point (yellow dot) and the ending point (green dot).



(a) Computed coverage trajectory of the first proposed scenario.

(b) The comparison of the computed trajectory and the trajectory of the real deployment in the first scenario.

Figure 6.3: Results of the first proposed experimental scenario.

Based on the captured trajectories, the computed trajectory and the actual trajectory are very similar. The differences were caused by the wind, and the measured wind speed was oscillating between 3 m s^{-1} and 6 m s^{-1} during the tests.

6.2.2 Experimental Scenario 2

The second experiment, depicted in Figure 6.4, contains two regions to be covered. The computed coverage trajectory intentionally does not contain any re-entries between the regions. The regions are in such a distance that the fixed-wing UAV covers firstly one region and then the other one before landing. The computed coverage trajectory is depicted in Figure 6.5a. The length of the final coverage path is 5238 m, and the solutions of the LKH and CPLEX are the same.

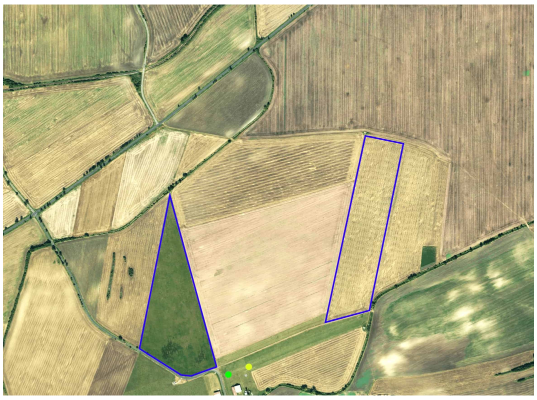
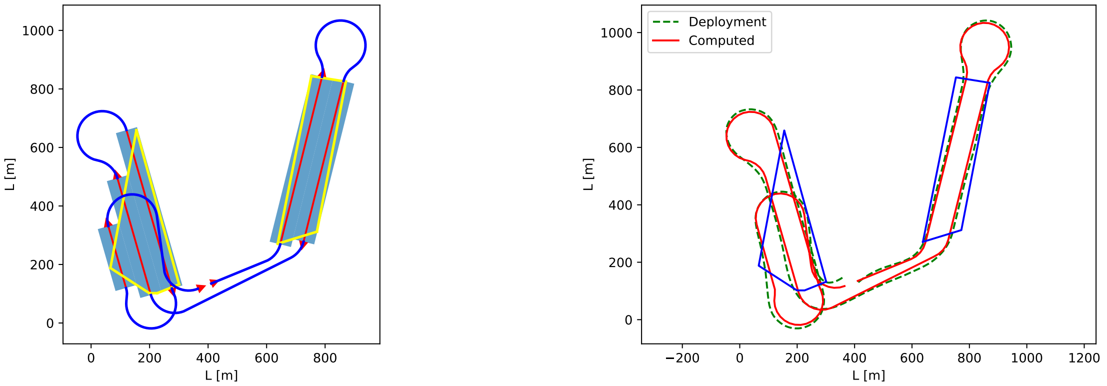


Figure 6.4: The second proposed testing scenario. The regions to be covered (blue polygon) represents two fields that are close to the starting point (yellow dot) and the ending point (green dot).



(a) Final coverage trajectory of the second proposed scenario.

(b) The comparison of the computed trajectory and the trajectory of the real deployment in the second scenario.

Figure 6.5: Results of the second proposed experimental scenario.

A notable feature of the final coverage trajectory is that the region on the right is covered in a non-dominant coverage heading. The dominant coverage heading is a heading for which the width of the region is minimal. In a multi-region scenario, it can happen as it did here, that some of the regions are not covered in a dominant coverage heading because the selected headings for each region are influencing each other. In more complicated scenarios, the selected coverage headings can consist of many coverage headings that are not dominant for the given regions. The position of the starting and ending points is also influencing the final selected headings for the coverage.

6.2.3 Experimental Scenario 3

The last proposed experimental scenario, depicted in Figure 6.6, contains three regions. The final coverage trajectory has been computed by the LKH solver only due to the time complexity of the optimal solution. The scenario contains mutually adjacent regions that can make re-entries beneficial.

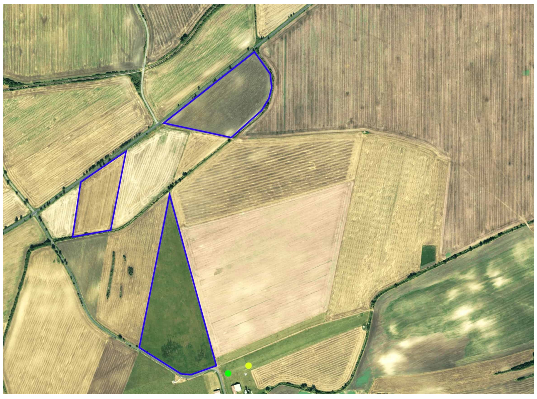
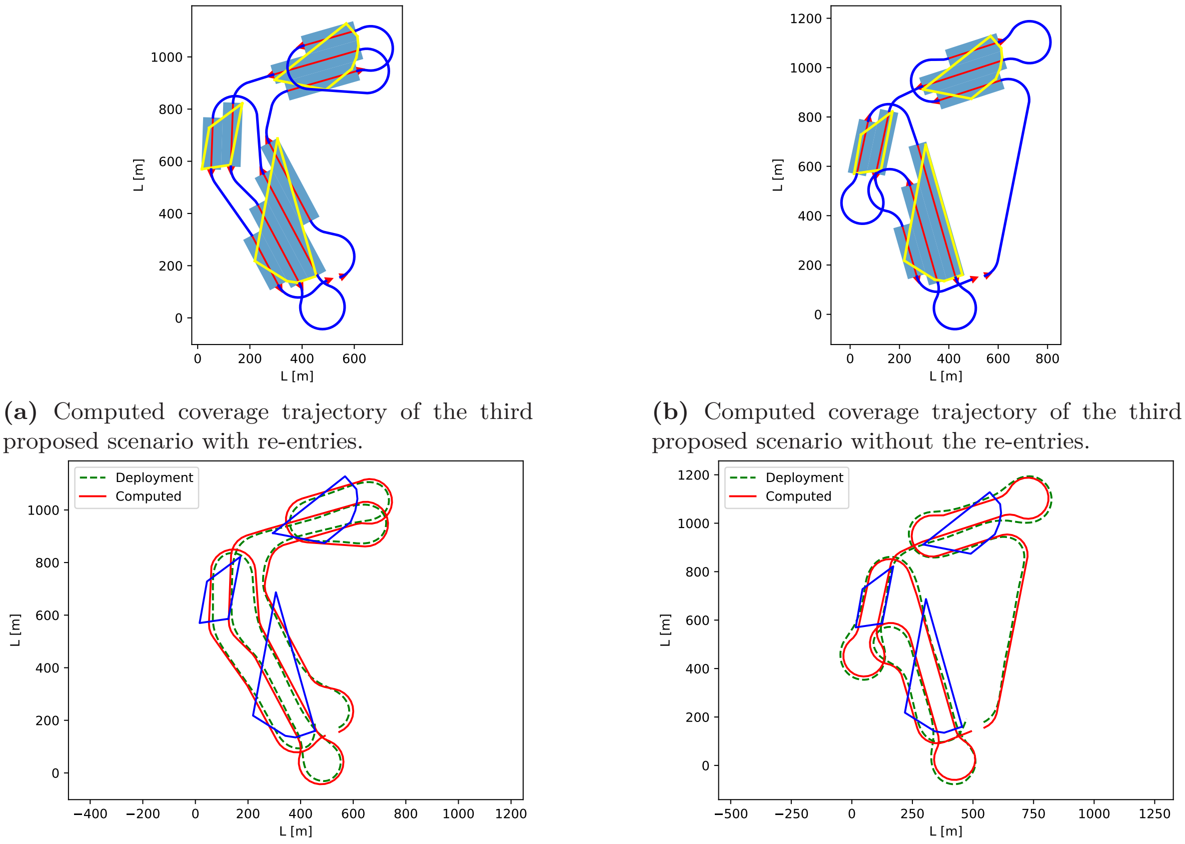


Figure 6.6: The third experimental scenario. The regions to be covered (blue polygon) represents two fields that are close to the starting point (yellow dot) and the ending point (green dot.)

The coverage trajectory containing the re-entries, depicted in Figure 6.7a is 6235 m long and as before, some of the regions are covered in a non-dominant coverage heading. The length of the trajectory without the re-entries, depicted in Figure 6.7b is 6583 m. Therefore, the found solution further demonstrates the benefits of re-entries about 5 % more efficient than without the re-entries. The comparison of the real trajectories and the computed trajectories of both variants of this scenario is depicted in Figure 6.7.

Because of the safety concerns, more complex testing scenarios weren't used in a real deployment due to the problems with finding enough large areas without the inhabitants and infrastructure. To increase the safety in a third proposed scenario, the actual regions to be covered were shifted by 500 m to the East in order to maintain a greater distance from the roads displayed on the map, depicted in Figure 6.6.



(a) Computed coverage trajectory of the third proposed scenario with re-entries.

(b) Computed coverage trajectory of the third proposed scenario without the re-entries.

(c) Computed and real trajectories in the third scenario with the re-entries.

(d) Computed and real trajectories in the third scenario without the re-entries.

Figure 6.7: Comparison of the coverage trajectories of the third proposed scenario.

CHAPTER 7

Conclusion

In this thesis, the multi-region area coverage problem with fixed-wing UAV is studied. Existing works are focused on the single-region area coverage problem, and only a few methods address the multi-region area coverage problem. We evaluate a basic method solving the multi-region area coverage problem and show that it deserves to be further studied. The results show that the multi-region area coverage problem can be treated differently for cases with nearby regions to be covered and distant regions. The proposed method allows the fixed-wing UAV to cover the regions with the trajectory containing the re-entries, minimizing its length and allowing the UAV to maintain the higher velocity. Trajectories with re-entries are shorter in comparison with the traditional approach that covers each region separately. Furthermore, the results indicate that the traditional approach to the area coverage problem can lead to longer coverage trajectories.

In the proposed approach, coverage headings are examined to select heading for each region. The final path can approach the regions in non-dominant coverage headings, unlike the existing methods. The benefit of re-entries is influenced by the distance between regions and the maneuvering capabilities of the deployed fixed-wing UAV. Both factors, the changing distance of the regions and the changing turning radius of the fixed-wing UAV, have been evaluated and compared with the traditional approach.

The multi-region area coverage problem is a recent challenging topic of the general area coverage problem known in the literature; hence, the presented results are for relatively small scenarios. The proposed method focuses on finding the dominant combination of the coverage headings and shows the influence of the distance between the regions, and capabilities of the fixed-wing UAVs on the selected headings. The problem is formally defined as a variant of the Generalized Traveling Salesman Problem (GTSPN) that can be solved optimally with a high computational cost. Therefore, the optimal solution is utilized as the baseline reference approach, and a more pragmatically heuristic-based solution is employed in the proposed method. Based on the reported results, the proposed heuristic solution provides solutions of similar quality to the optimal solution. Thus, the proposed method is a suitable choice for practical deployments, further supported by the reported results on experimental field deployment with a real vehicle.

Bibliography

[1] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[2] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of mavs," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2708–2714, IEEE, 2011.

[3] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar, "Devices, systems, and methods for automated monitoring enabling precision agriculture," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 462–469, IEEE, 2015.

[4] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[5] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Multi-uas path-planning for a large-scale disjoint disaster management," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 799–807, IEEE, 2019.

[6] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

[7] J.-D. Nicoud and M. K. Habib, "The pemex-b autonomous demining robot: perception and navigation strategies," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, pp. 419–424, IEEE, 1995.

[8] 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.

[9] P. Liu, A. Y. Chen, Y.-N. Huang, J.-Y. Han, J.-S. Lai, S.-C. Kang, T.-H. Wu, M.-C. Wen, M.-H. Tsai, *et al.*, "A review of rotorcraft unmanned aerial vehicle (uav) developments and applications in civil engineering," *Smart Struct. Syst*, vol. 13, no. 6, pp. 1065–1094, 2014.

[10] A. Goodchild and J. Toy, "Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing co2 emissions in the delivery service industry," *Transportation Research Part D: Transport and Environment*, vol. 61, pp. 58–67, 2018.

[11] K. Yu, J. M. O'Kane, and P. Tokekar, "Coverage of an environment using energy-constrained unmanned aerial vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3259–3265, IEEE, 2019.

[12] L. Liu and N. Michael, "Energy-aware aerial vehicle deployment via bipartite graph matching," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 189–194, IEEE, 2014.

[13] K. C. Magoteaux, B. Sanders, and H. A. Sodano, "Investigation of an energy harvesting small unmanned air vehicle," in *Active and Passive Smart Structures and Integrated Systems 2008*, vol. 6928, p. 692823, International Society for Optics and Photonics, 2008.

[14] 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.

[15] 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.

[16] R. Samar and A. Rehman, "Autonomous terrain-following for unmanned air vehicles," *Mechatronics*, vol. 21, no. 5, pp. 844–860, 2011.

[17] O. Kazemifar, A.-R. Babaei, and M. Mortazavi, "Online aircraft velocity and normal acceleration planning for rough terrain following," *The Aeronautical Journal*, vol. 121, no. 1244, pp. 1561–1577, 2017.

[18] 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.

[19] U. Peless, "Robomow by friendly machines the first commercial fully automatic and systematic lawn mower," *May 19th and 20th, 1998 Technion—Israel Institute of Technology Haifa, Israel*, p. 570, 1998.

[20] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an auv," in *Underwater Robots*, pp. 17–45, Springer, 1996.

[21] H. Wang, H. Li, C. Zhang, S. He, and J. Liu, "A 3d coverage path planning approach for flying cameras in nature environment under photogrammetric constraints," in *2017 36th Chinese Control Conference (CCC)*, pp. 6761–6766, IEEE, 2017.

[22] J. Jin and L. Tang, "Coverage path planning on three-dimensional terrain for arable farming," *Journal of field robotics*, vol. 28, no. 3, pp. 424–440, 2011.

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

[24] 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.

[25] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, "Concorde TSP Solver," 2003. [cited 9 Aug 2020].

[26] J. S. Lewis, W. Edwards, K. Benson, I. Rekleitis, and J. M. O'Kane, "Semi-boustrophedon coverage with a dubins vehicle," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5630–5637, IEEE, 2017.

[27] L. Nam, L. Huang, X. J. Li, and J. Xu, "An approach for coverage path planning for uavs," in *2016 IEEE 14th international workshop on advanced motion control (AMC)*, pp. 411–416, IEEE, 2016.

[28] X. Yu, T. A. Roppel, and J. Y. Hung, "An optimization approach for planning robotic field coverage," in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 004032–004039, IEEE, 2015.

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

[30] J. I. Vasquez-Gomez, J. C. Herrera-Lozada, and M. Olguin-Carbajal, "Spatial resolution optimization for terrain coverage with uavs," in *2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pp. 37–42, IEEE, 2017.

[31] K. N. Tahar and A. Ahmad, "An evaluation on fixed wing and multi-rotor uav images using photogrammetric image processing," in *Proceedings of World Academy of Science, Engineering and Technology*, no. 73, p. 391, World Academy of Science, Engineering and Technology (WASET), 2013.

[32] K. Kučerová, P. Váň, and J. Faigl, "On finding time-efficient trajectories for fixed-wing aircraft using dubins paths with multiple radii," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 829–831, 2020.

[33] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *2007 46th IEEE conference on decision and control*, pp. 2379–2384, IEEE, 2007.

[34] 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 *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1374–1379, IEEE, 2016.

[35] K. Yu, A. K. Budhiraja, S. Buebel, and P. Tokekar, "Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations," *Journal of Field Robotics*, vol. 36, no. 3, pp. 602–616, 2019.

[36] C. Icking and R. Klein, "Competitive strategies for autonomous systems," in *Modelling and Planning for Sensor Based Intelligent Robot Systems*, pp. 23–40, World Scientific, 1995.

[37] D. W. Gage, “Randomized search strategies with imperfect sensors,” in *Mobile Robots VIII*, vol. 2058, pp. 270–279, International Society for Optics and Photonics, 1994.

[38] Y. Gabriely and E. Rimon, “Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 954–960, IEEE, 2002.

[39] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, “Planning paths of complete coverage of an unstructured environment by a mobile robot,” in *Proceedings of international conference on advanced robotics*, vol. 13, pp. 533–538, 1993.

[40] E. González, P. Aristizábal, and M. Alarcón, “Backtracking spiral algorithm: a mobile robot region filling strategy,” in *Proceeding of the 2002 international symposium on robotics and automation, Toluca, Mexico*, pp. 261–266, 2002.

[41] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, “Bsa: a complete coverage algorithm,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2040–2044, IEEE, 2005.

[42] 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.

[43] J.-C. Latombe, *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.

[44] 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.

[45] J. W. Milnor, M. Spivak, and R. Wells, *Morse theory*, vol. 1. Princeton university press Princeton, 1969.

[46] E. U. Acar and H. Choset, “Critical point sensing in unknown environments,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 4, pp. 3803–3810, IEEE, 2000.

[47] W. H. Huang, “Optimal line-sweep-based decompositions for coverage algorithms,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1, pp. 27–32, IEEE, 2001.

[48] 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.

[49] J. Araujo, P. Sujit, and J. B. Sousa, “Multiple uav area decomposition and coverage,” in *2013 IEEE symposium on computational intelligence for security and defense applications (CISDA)*, pp. 30–37, IEEE, 2013.

[50] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, “Efficient multi-robot coverage of a known environment,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1846–1852, IEEE, 2017.

[51] 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.

[52] B. Oskin, "Japan earthquake and tsunami of 2011: Facts and information," Sep 2017.

[53] R. Mlambo, I. H. Woodhouse, F. Gerard, and K. Anderson, "Structure from motion (sfm) photogrammetry with drone data: A low cost method for monitoring greenhouse gas emissions from forests in developing countries," *Forests*, vol. 8, no. 3, p. 68, 2017.

[54] G. V. Laurin, Q. Chen, J. A. Lindsell, D. A. Coomes, F. Del Frate, L. Guerriero, F. Pirotti, and R. Valentini, "Above ground biomass estimation in an african tropical forest with lidar and hyperspectral data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 89, pp. 49–58, 2014.

[55] J. Marek, "Úloha pokrỳvání oblastí vzdušnỳmi robotickỳmi prostředky," B.S. thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2018.

[56] C. E. Noon and J. C. Bean, "An efficient transformation of the generalized traveling salesman problem," *INFOR: Information Systems and Operational Research*, vol. 31, no. 1, pp. 39–44, 1993.

[57] K. Helsgaun, *An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic*. PhD thesis, Roskilde University. Department of Computer Science, 2006.

[58] K. Helsgaun, "LKH solver 2.0.9," 2018. [cited 9 Aug 2020].

[59] "CPLEX." [cited on 2020-Aug-11].

[60] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the operations research society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[61] R. Roberti and P. Toth, "Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison," *EURO Journal on Transportation and Logistics*, vol. 1, no. 1-2, pp. 113–133, 2012.

[62] "Julia 1.5 Documentation." [cited on 2020-Aug-11].

APPENDIX A

Properties of the Experimental Scenarios

A.1 Scenario 1

Table A.1: Properties of the first experimental scenario.

	Real world coord.	**Computational coord.**
Reference point:	50.4320N/13.8070E	
Start coordinates:	50.4329N/13.8093E	[142,100]
End coordinates:	50.4326N/13.8082E	[106,83]
Initial heading:	68°	
Final heading:	68°	
Polygon coordinates:		
	50.4331N/13.8078E	[50,122]
	50.4344N/13.8126E	[396,271]
	50.4380N/13.8137E	[477,676]
	50.4367N/13.8062E	[-62,520]

A.2 Scenario 2

Table A.2: Properties of the second experimental scenario.

	Real world coord.	**Computational coord.**
Reference point:	50.4317N/13.8035E	
Start coordinates:	50.4329N/13.8093E	[411, 135]
End coordinates:	50.4326N/13.8082E	[365, 117]
Initial heading:	68°	
Final heading:	68°	
Polygon 1 coordinates:		
	50.4334N/13.8045E	[301, 132]
	50.4326N/13.8063E	[155, 659]
	50.4326N/13.8067E	[66, 188]
	50.4329N/13.8078E	[199, 102]
	50.4376N/13.8057E	[226, 102]
Polygon 2 coordinates:		
	50.4341N/13.8125E	[638, 271]
	50.4345N/13.8144E	[772, 312]
	50.4392N/13.8156E	[871, 825]
	50.4393N/13.8142E	[753, 844]

A.3 Scenario 3

Table A.3: Properties of the third experimental scenario.

	Real world coord.	**Computational coord.**
Reference point:	50.4314N/13.8013E	
Start coordinates:	50.4329N/13.8093E	[547, 156]
End coordinates:	50.4326N/13.8082E	[492, 141]
Initial heading:	68°	
Final heading:	68°	
Polygon 1 coordinates:		
	50.4334N/13.8045E	[455, 160]
	50.4326N/13.8063E	[306, 687]
	50.4326N/13.8067E	[219, 218]
	50.4329N/13.8078E	[342, 141]
	50.4376N/13.8057E	[381, 135]
Polygon 2 coordinates:		
	50.4365N/13.8015E	[125, 586]
	50.4366N/13.8032E	[171, 822]
	50.4388N/13.8038E	[44, 728]
	50.4379N/13.8019E	[16, 570]
Polygon 3 coordinates:		
	50.4396N/13.8055E	[295, 912]
	50.4392N/13.8084E	[491, 874]
	50.4400N/13.8097E	[590, 954]
	50.4402N/13.8100E	[608, 998]
	50.4406N/13.8101E	[616, 1042]
	50.4410N/13.8101E	[612, 1079]
	50.4416N/13.8094E	[569, 1128]