Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Bachelor's Project

Energy-aware Coverage of Large Areas with Unmanned Aircraft

Adam Johanides

Supervisor: Ing. Martin Selecký

Study Program: Open Informatics

Study Branch: Software Systems

May 25, 2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Johanides** Jméno: **Adam** Osobní číslo: **434775**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Softwarové systémy**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Pokrývání rozlehlých oblastí pomocí bezpilotních strojů s omezeným doletem

Název bakalářské práce anglicky:

Energy-aware Coverage of Large Areas with Unmanned Aircraft

Pokyny pro vypracování:

1) Study the problems of area coverage and energy-aware planning
2) Design a method for coverage of large areas with UAS with limited flight time
3) Design a method that minimizes the time for area coverage in case of multiple starting locations
4) Implement these methods and compare the required coverage times and number of flights for large area coverage

Seznam doporučené literatury:

[1] Li, Y., Chen, H., Er, M.J. and Wang, X., 2011. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics, 21(5), pp. 876-885.
[2] Di Franco, C. and Buttazzo, G., 2016. Coverage path planning for uavs photogrammetry with energy and resolution constraints. Journal of Intelligent & Robotic Systems, 83(3-4), pp. 445-462.
[3] Barrientos, A., Colorado, J., Cerro, J.D., Martinez, A., Rossi, C., Sanz, D. and Valente, J., 2011. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. Journal of Field Robotics, 28(5), pp. 667-689.
[4] Santamaria, E., Segor, F., Tchouchenkov, I. and Schönbein, R., 2013, May. Rapid aerial mapping with multiple heterogeneous unmanned vehicles. In ISCRAM.
[5] Avellar, G.S., Pereira, G.A., Pimenta, L.C. and Iscold, P., 2015. Multi-uav routing for area coverage and remote sensing with minimum time. Sensors, 15(11), pp. 27783-27803.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Selecký, katedra počítačů FEL

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

Datum zadání bakalářské práce: **15.02.2018** Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Martin Selecký podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

. _____
Datum převzetí zadání

_____
Podpis studenta

Aknowledgements

I would like to thank my supervisor Ing. Martin Selecký for the valuable comments and remarks he has given me during the creation of this work.

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 May 25, 2018 .....

Abstract

This thesis proposes an approach to the mapping of large convex areas using a single Unmanned Aerial Vehicle (UAV). Previous works focus on Coverage Path Planning (CPP) from only one start position, and they do not take into account the maximum UAV flight time. We focus on areas too large to be covered with a single flight. After a battery replacement, it is efficient to start the mapping of a next subarea directly from within it. Sometimes, however, it is impossible for the UAV to launch from optimal positions due to some restrictive conditions such as forests or other natural constraints, or it would take too long for the operator to move to the optimal starting location. The proposed algorithm divides a given area of interest between appropriately selected launch positions and applies a method for solving the Energy Constrained Coverage Path Planning (ECCPP) problem. We compare the proposed method with an approach that uses only a single launch position. Test results show that for large areas where more batteries are needed, it is possible to save time and the number of used batteries with the proposed approach, to the detriment of operator relocation.

Key Words: Unmanned Aerial Vehicles, Coverage Path Planning, Energy-aware Coverage

Abstrakt

Tato práce navrhuje řešení pro mapování rozsáhlých konvexních ploch s použitím jednoho bezpilotního stroje (UAV). Dosavadní práce navrhují plánování tras UAV pouze z jedné startovací pozice a nezohledňují jeho maximální dobu letu. My se zaměřujeme na oblasti, které jsou příliš velké na pokrytí jedním průletem. Po výměně baterie je výhodné začít mapování další části oblasti přímo zevnitř dané oblasti. Někdy je však nemožné, aby UAV startovalo z optimální pozice, a to kvůli omezujícím podmínkám, jakými jsou například lesy nebo jiná přírodní omezení, nebo by operátorovi přesun do dané pozice trval příliš dlouho. Navrhovaný algoritmus rozděluje oblast na dílčí části pokrývané z různých dostupných bází a aplikuje na ně metodu řešící námi definovaný problém plánování pokrývání oblasti s omezeným doletem (ECCPP). Tento přístup jsme porovnali s metodou využívající pouze jedno startovací místo. Výsledky testování ukazují, že pro rozsáhlé oblasti, kde je zapotřebí většího množství baterií, je s touto metodou možné ušetřit čas a počet použitých baterií, na úkor potřeby přemístění operátora.

Klíčová slova: bezpilotní stroje, plánování pokrývání oblastí, plánování s omezeným doletem

Překlad názvu: Pokrývání rozlehlých oblastí pomocí bezpilotních strojů s omezeným doletem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last two decades, Unmanned Aerial Vehicles (UAVs) were found useful in a variety of areas. One of the usages is an area mapping for various reasons. Currently, satellites, planes, and UAVs are the three most common ways of obtaining images of areas. Thanks to lower cost, lower dependency on the weather and ease of use, UAVs have become very practical in this tasks. The Coverage Path Planning (CPP) algorithms are used to generate paths that ensure full coverage of the area of interest. CPP applications of UAVs are already used for terrain mapping, target search, security monitoring, etc. The usage of UAVs also reduces human involvement in potentially hazardous tasks, so they are frequently used in military operations [9].

The coverage path planning problem has been discussed in several papers from energy consumption and geometrical points of view. However, these works do not address the issue of coverage of large areas with the possibility of multiple starting positions.

1.1 Motivation

Obtaining images of a whole area can be very time-consuming. There are situations where we need to cover an area too large to be covered by a single flight. Then it is best to start with the UAV from a position near the subarea to be covered. Time spent on a coverage highly depends on revisiting already photographed locations by the UAV. This revisiting is inevitable if the battery needs to be replaced. If a finite set of starting positions is given, it should be possible to divide the area between multiple starting positions minimising revisited locations and time of the coverage mission. However, sometimes the number of starting positions is limited or it is better to take off from a farther location because getting to the optimal position would take too long on the ground (see Figure 1.1). Current approaches have mostly addressed the subproblems of this issue, and we, therefore, define the Energy Constrained Coverage Path Planning (ECCPP).

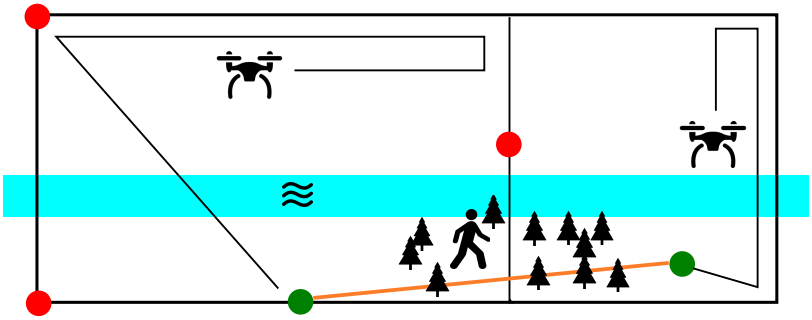


Figure 1.1: An example illustrating the situation where the number of starting positions is limited because of the forest. Unused starting positions are in red and used in green. The launch position closer to the first one is not selected because of the river and high cost of transportation between them. The orange line represents the path on the ground.

The motivation is to provide a coverage path solutions using multiple starting positions, rather than covering the whole area from a single location. This approach involves decomposition of the area between launch positions, effective coverage path planning for every subarea and computation of the time necessary for travelling between starting positions. The main goal of the thesis is to describe the problem of ECCPP, propose a method to address it, experimentally verify the solution in various scenarios and compare it with a simple Lawnmower approach.

1.2 Thesis Structure

This thesis is organised as follows. After this introduction, the related works associated with the aerial mapping and CPP is investigated in Chapter 2. In Chapter 3 we discuss the related subproblems and formulate the ECCPP problem. We propose the goal specification at the end of the chapter, together with the chapter summary. Chapter 4 provides the method used for solving the ECCPP problem. The approach is divided into partial problems, which are solved separately. In Chapter 6 we summary the content of this work and the results achieved. It further discusses the advantages and disadvantages of the proposed ECCPP method and mentions possible further work for improvements.

Chapter 2

Related Work

This chapter briefly introduces problems related to aerial mapping and mentions some of the previous works. Aerial mapping problem with UAVs is closely related to the Coverage Path Planning (CPP). The majority of CPP research has been focused on Unmanned Ground Vehicles (UGVs) in the past [20, 15, 35, 3]. However, several works already exist for UAVs as well as for Autonomous Underwater Vehicles (AUVs) [21]. A standard part of the CPP is avoiding already visited cells and minimisation of the time duration required for the coverage [15]. The approach of CPP differs according to constraints of a type of the vehicle and according to the conditions where the vehicle is used. CPP for the UAVs mainly depends on the manoeuvring capabilities and maximal flight time of the aircraft. Apart from UAV's limitations, weather factors such as possible windy environment or bad lighting conditions should be taken into account as well. However, these factors are usually omitted for the sake of simplicity.

This work addresses optimisation in energy-aware CPP using multiple starting positions. Section 2.1 describes two main approaches for CPP and their related works. Section 2.2 focuses on the problem of area decomposition as the assignment of subregions to the available launch positions is part of the problem at hand. Approaches to the following problem, the search for the time-optimal visiting order of launch positions, are discussed in Section 2.3. The problem of area decomposition has been addressed for scenarios with multiple UAVs. The related approaches are discussed in Section 2.4. Lastly, Section 2.5 mentions works focusing on energy efficiency optimisation of the UAV flight.

2.1 Online and Offline Approaches

There are two main approaches for CPP, offline and online. The online algorithms are sensor-driven and propose plans based on sensor data gathered during the coverage from the field. The benefits of the online approach are that detailed knowledge of the area is not required and the aircraft is able to complete the coverage mission regardless of the unexpected disturbances such as another aircraft on the way. Paull et al. [23] have presented such a sensor-driven algorithm along with real-time optimisation of angles of a camera gimbal to maximise the rate of coverage. These authors also proposed a similar online algorithm for AUVs [21], and they illustrated benefits of the online algorithm over the standard offline

approach called Lawnmower pattern. Sujit and Beard [27] proposed a cooperative CPP technique for multiple UAVs exploring an unknown region.

The offline approach [17, 2, 7] plans the path before the coverage mission and assumes the UAV can follow the planned trajectory precisely. This approach brings better efficiency if there is complete knowledge of the area. It also decreases the on-board computation demands.

2.2 Area Decomposition

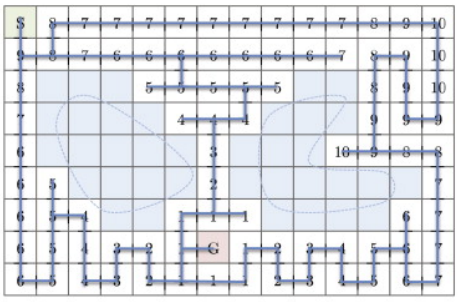
Decomposition of the area is required if multiple flights or multiple drones are used. There are two commonly used decomposition approaches for UAVs, approximate cellular decomposition [29] and exact cellular decomposition [17]. The approximate decomposition uses a grid with regular cells. Algorithms used with the grid approximation usually focus on the cell visiting order. The first grid-based CPP algorithm was proposed by Zelinsky et al. [34]. The goal and starting position must be specified in the proposed algorithm. All grid cells are marked with numbers using the wavefront algorithm as follows:

1. Assign a 0 to the goal cell and a 1 to all adjacent cells.
2. Assign a 2 to all unmarked cells adjacent to cells marked by 1.
3. Repeat the step 2 incrementally until the start cell is reached.

The path planning starts on the start cell. Then in every step during the CPP, next cell is found by selecting a non-visited cell with the highest label adjacent to the last visited cell. Next cell is selected randomly if two or more cells with the same label exist in the neighbourhood. If there is no unvisited cell in the neighbourhood of the current cell, the vehicle returns to the previous position, see Figure 2.1.

S	8	7	7	7	7	7	7	7	7	7	8	9	10
9	8	7	6	6	6	6	6	6	6	7	8	9	10
8				5	5	5	5	5			8	9	10
7					4	4	4				9	9	9
6						3				10	9	8	8
6	5					2							7
6	5	4			1	1	1					6	7
6	5	4	3	2	1	G	1	2	3	4	5	6	7
6	5	4	3	2	1	1	1	2	3	4	5	6	7

(a) Wavefront distance transform for the selection of the start position (S) goal position (G).



(b) Coverage path generated using the wavefront distance transform with the selection of the start position (S).

Figure 2.1: Grid-based coverage using the wavefront algorithm [11]

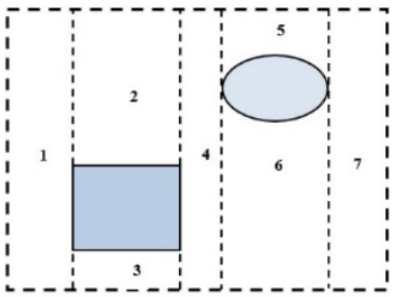
Approximate decomposition does not have to be rectangular. For example, triangular-cell-based map approximation [20] was used for cleaning robot navigation, where the triangular representation enables more navigation directions. However, the rectangular grid is

mostly used for UAVs, because a rectangular cell can represent actual footprint of the camera during aerial mapping [2]. Santamaria et al. [26] propose a fast offline algorithm for multiple UAVs using approximate cellular decomposition, similar to the decomposition used in this thesis. The mapped area is partitioned between UAVs with different cell size according to the footprint of a current UAV. Path planning algorithm used in that work prioritises the selection of long straight segments on the grid, and the authors showed that this approach reduces the number of turns of a vehicle.

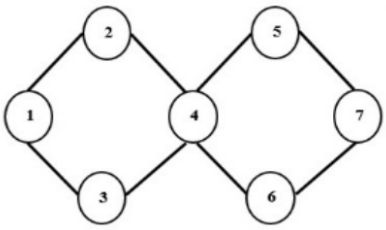
On the contrary, exact cellular decomposition divides the original area into a set of (usually convex) subareas. Maza and Ollero [18] use $n-1$ lines for exact decomposition of a convex area without any obstacles between n vehicles. After decomposition, they compute an optimal sweep direction of the zigzag pattern for every subregion.

2.3 Subregion Visiting Order Estimation

After area decomposition into smaller subregions, backtracking techniques are used to solve the CPP. This issue can be formulated as a Travelling Salesman Problem (TSP), which can usually be found in literature as a Hamiltonian Cycle Problem [15]. The area of interest can be represented as an adjacency graph after decomposition (see Figure 2.2) and then optimal backtracking sequence can be used. A direct solution for the TSP is to try all permutations minimising the cost. However, TSP belongs to the class of NP-complete problems and running time increases rapidly with the number of locations [10].



(a) Boustrophedon decomposition resulting in 7 cells.



(b) Adjacency graph of the environment.

Figure 2.2: Example of adjacency graph representation for backtracking techniques. [15]

TSP is difficult to solve in large problems of aerial mapping because of a high number nodes. Also, node relations would have to be defined because turns of the UAV should be minimised. However, several backtracking sequence approaches for the CPP have been proposed with approximate, practical solutions.

A common approaches for this task are based on greedy algorithms like Depth First Search (DFS) [2], A* [31] or extension of D* called Complete Coverage D* (CCD*) [5]. However, dynamic programming [35] and evolutionary algorithms such as Ant Colony Optimisation (ACO) [3] or Particle Swarm Optimisation (PSO) [22] can be found in the literature to solve the visiting order of the cells during the CPP too.

2.4 Aerial Mapping With Multiple UAVs

A practical approach [2] to a crop field mapping by using a fleet of UAVs was proposed with an offline algorithm decomposing the area into a rectangular grid. In the first phase, robots divide the field among themselves. The criteria for negotiation for each of the drones is to cover as much area as possible in the given task, according to their limitations such as battery capacity, and to minimise the task overlapping with the other robots. Avellar et al. [1] propose a solution to the problem of area mapping with multiple UAVs. In this case, numerous UAVs starts from a single position one after another and maximum flight time and time needed to prepare UAVs are taken into account. A similar approach to the complete coverage path planning [28] proposes two strategies for stationary and persistent long-term coverage with multiple Small Unmanned Vehicles using ground charging stations.

2.5 Energy-aware approach

Franco and G. Buttazzo [6] proposed an energy-aware algorithm reducing the required total distance travelled for area coverage. However, their approach does not reason with the multiple flight requirements in the case of large areas. Karydis et al. [14] centres on increasing energy efficiency of small-scale aerial vehicles. Li et al. [16] model the 3D terrains first and then create an energy consumption map to find energy-optimal paths. Ware and Roy [32] improve flight performance of the aircraft in an urban environment by taking complex urban wind fields into account. Energy consumption minimisation method for six-rotor aircraft was proposed by Vicencio et al. [30] using Hybrid Random-Key Genetic Algorithm.

2.6 Summary

The aerial mapping problem was addressed many times with different approaches to cover the assigned area with a single flight. These approaches do not address the necessary returns of the UAVs to the starting position for battery replacement or the possibility to take-off from a limited set of locations. These mission constraints can significantly affect the overall mission efficiency, and they have not been addressed in any previous work to the best of our knowledge. The approach described in our work addresses these constraints and demonstrate its efficiency in a series of experiments.

Chapter 3

Problem Formulation and Goal Specification

In this chapter, we propose the Energy Constrained Coverage Path Planning (ECCPP) problem as an extension to the common Coverage Path Planning (CPP) problem with a limitation of maximal flight time of the aircraft. This chapter describes the main concepts and individual parts of the ECCPP problem. Section 3.1 proposes a description of different types of aircraft and their properties along with a calculation of their projected area to fully discuss the topic. Following part addresses the CPP problem and its common solution approaches. A common approach for the CPP with a single launch position called Lawnmower pattern is described in Section 3.2. Section 3.3 then focuses on the energy constraints which are taken into account in the minimisation criterion discussed in Section 3.4. The main motivation of this work is modification of the often used Lawnmower pattern approach for large areas with a finite set of starting positions. The area decomposition allows shorter returns necessary for battery replacements. Therefore, Section 3.5 describes the decomposition of the area and assignment of the subareas to the particular launch positions, its possible advantages and limitations. The goal specification is located at the end of this chapter.

3.1 Introducing Main Concepts

3.1.1 UAV

An Unmanned aerial vehicle (UAV) is an aircraft vehicle without a human pilot aboard. Also, the term drone is commonly used by the public. It is controlled remotely by a human or flies with an autonomy. Unmanned Aerial System (UAS) consists of all components needed to operate the system. It includes the ground control system, GPS, camera, software and the UAV itself. There are many types of UAVs. They come in different sizes and shapes, and each of these has their pros and cons. However, there are two main categories of UAVs at this time [33], rotary wing, and fixed wing, see Figure 3.1.

The Fixed Wing UAVs consists of a rigid wing. UAV's forward airspeed generates lift and makes flight capable. A combustion engine or electric motor usually generate the forward thrust. A much simpler structure of a fixed-wing UAV is the main advantage in comparison

with a rotary wing. Maintenance and repair process is less complicated and the simple structure allows longer flight duration and larger survey areas per flight. Rotary wing UAVs come in wide range of setups with one or multiple rotors. Constant aircraft forward movement is not necessary, because the rotors themselves produce airflow required to generate lift. The main advantage is the ability for VTOL (vertical take-off/landing). With rotary UAVs, the operator can operate with no substantial takeoff/landing area required. They are well suited to applications where precision manoeuvring is needed, because of their capacity of hovering and better agile manoeuvring. Longer flight ranges are usually preferred for aerial mapping discussed in this thesis. Therefore, if the user is not limited to a landing/takeoff area or if slower flight with a lower altitude is not preferable, fixed-wing UAVs are more suitable for this task.

(a) Fixed Wing

(b) VTOL

Figure 3.1: UAV type comparison

3.1.2 Projected Area

A portion of the area, which corresponds with a picture of the camera pointing down, is called projected area or the camera footprint. This footprint is usually rectangular or nearly rectangular, depending on the camera pitch angle. The width of the projected area depends on the altitude and the angle of view of the camera, see Figure 3.3. The camera footprint is shown in Figure 3.2.

The bottom width w_2 of the projected area is usually used as the width of images during the sweep motion, to avoid blurry images. [17]

Let h be the flight altitude, $fov1$ the vertical angle of FOV (Field of View), $fov2$ the horizontal angle of FOV, af the front-mounted angle and θ the pitch angle as is shown in Figure 3.2. Then the width w_2 can be computed as follows [17]:

$$w_2 = \frac{2h\tan(fov2/2)}{\sin(af - \theta + fov1/2)} \tag{3.1}$$

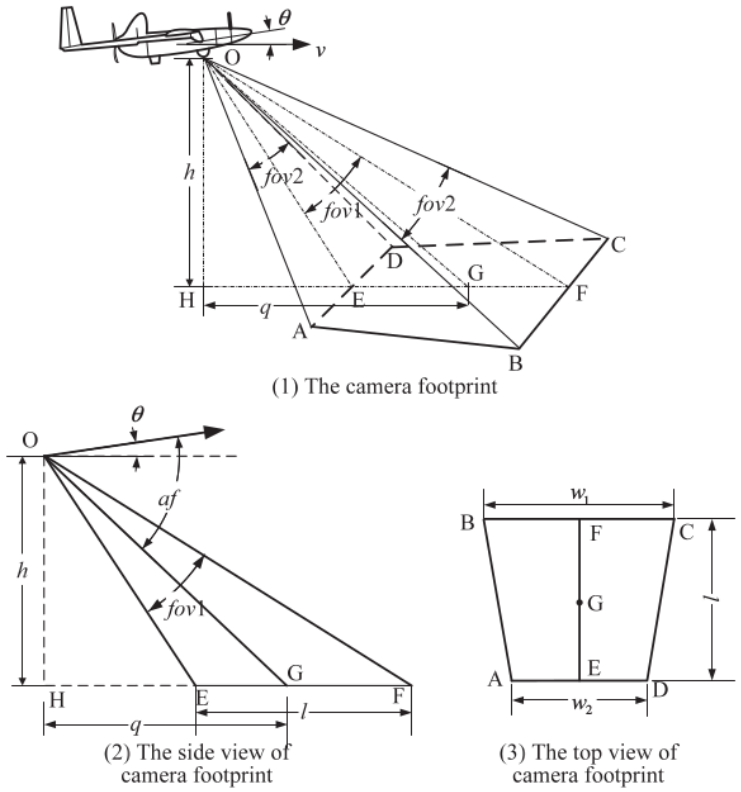


Figure 3.2: The camera footprint of an UAV [17]

3.1.3 Coverage Path Planning

In aerial mapping, coverage path planner determines a path that passes over all points of the area of interest. CPP where the whole area is to be covered is also often called as Complete Coverage Path Planning (CCPP) in literature. The result of a CPP algorithm should be a finite set of way-points for a given UAV/UAVs. There are three main criterion in literature. Area decomposition, sweep direction selection and backtracking mechanism.

Since the number of turns of the UAV significantly affects energy consumption and time of coverage, optimal sweep direction minimising the number of turns is desirable [17]. Finding the optimal sweep direction is necessary especially with usage of exact cellular decomposition methods where the given area is divided into simple non-overlapping subregions. Then the subregions with no obstacles can be covered with a simple motion like in Lawnmower pattern. Standard methods for decomposition of an area with obstacles are trapezoidal decomposition, boustrophedon decomposition or morse-based cellular decomposition.

Other coverage methods use a grid-based approximation. This approximation approach provides a straightforward representation of the area of interest. However, memory requirements grow exponentially with the increase of grid size and regular grid cell structure leads to imprecise representation [15].

The optimal solution visits each location once and provides the shortest path possible. This issue can be formulated as a variation of Travelling Salesman Problem (TSP). Related work for these approaches is discussed in Section 2.3.

3.2 Lawnmower Pattern for Convex Polygon

3.2.1 Problem Definition

A convex polygon is a basic region which can be covered. A non-convex area can be decomposed into several subregions of a complex polygon, and then a backtracking algorithm can be applied for generating a sequence of subregions to visit. The back and forth motion, also known as Lawnmower or zig-zag pattern, is often selected to cover the convex polygon and it was also used in this work. The authors of [17] and [13] prove that the number of turns is the main factor that affects the energy consumption during the coverage. They also show that the optimal line-sweep motion direction for convex polygons is in a direction parallel to one of the boundary edges and corresponds to the parameters of the polygon such as its span and width.

Definition 3.2.1. The span (D) of a convex polygon (P) is the distance between a pair of support parallel lines (l_1, l_2). A line of support l is a line intersecting P and such that the interior of P lies to one side of l [25].

Definition 3.2.2. The width (W) of a convex polygon is the minimum span [17].

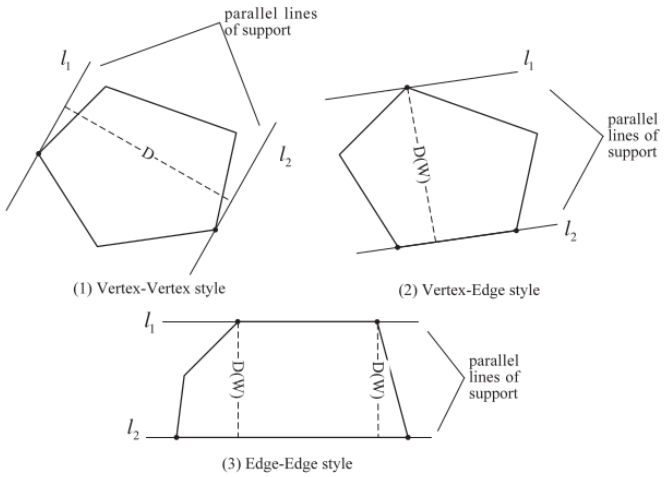


Figure 3.3: The span and width of a convex polygon [17]

Figure 3.3 shows three cases for the span of the polygon. According to [17], the width of the polygon can occur only in Vertex-Edge style span. Note that Edge-Edge style span can be considered as a special case of Vertex-Edge style span.

Let w be a width of a footprint of a camera. Then number of turns of the UAV is defined by

$$n_{turn} = \lceil D/w \rceil \tag{3.2}$$

Thus, we can find a minimum number of turns through calculation the width of the polygon W. Optimal coverage is obtained if the UAV flies along the direction of the line of support used for the minimum span [17].

3.3 Energy Constraints

Multiple flights with a single UAV are expected in this work. Therefore, battery replacement planning is desirable. This section describes the computation of information whether the UAV can cover the whole area with a single flight and energy needed for a safe return for planning the position of the UAV when the battery should be changed.

Definition 3.3.1. Let $E_{climb}(0,h)$ be the energy needed to ascend to the optimal flight altitude h, $E_{desc}(h,0)$ be the energy needed to land on the ground from the flight altitude h, n_t be the total number of turns, E_{turn} be the energy required for the turning motion, $E_{acc}(0,v_i^*)$ and $E_{dec}(0,v_i^*)$ be the energy needed to accelerate to/decelerate from the optimal speed v_i^* and $E_v(d_i,v_i^*)$ be the energy needed to travel between the extreme way-point of the i_{th} segment. Then the energy for the entire path E_{path} can be computed as follows:

$$E_{path} = E_{climb}(0,h) + E_{desc}(h,0) + n_t E_{turn} + \sum_i (E_{acc}(0,v_i^*) + E_{dec}(0,v_i^*) + E_v(d_i,v_i^*)) \tag{3.3}$$

Today's UAVs are usually powered by LiPo batteries. Each LiPo battery provides specification about its nominal voltage V_n, maximal capacity, the discharge rate and number of cells connected in series or in parallel. The battery capacity is expressed in mAh and specify the theoretical current I_H under which the battery would be discharged in one hour. However, discharging of LiPo batteries is not linear during flights [6]. Figure 3.4 shows how a discharging voltage can look.

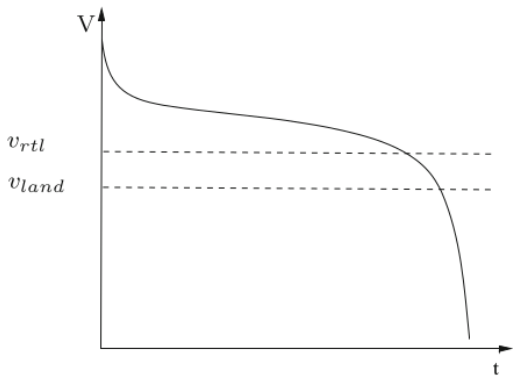


Figure 3.4: Example of discharging voltage curve of a 3S LiPo battery during a flight [6]

Therefore, only part of the total energy P (usually around 0.7) can be used to avoid damaging the UAV. Overusing of the battery can also completely damage the battery. Because of that reason, available energy is computed as [6]:

$$E_{tot} = V_n \cdot I_H \cdot t \cdot P \tag{3.4}$$

where t is a constant equal to 3600 s. The UAV is able to cover the whole area with a single flight if $E_{path} < E_{tot}$. If there is remaining energy (E_{tot} is much greater than E_{path}), the altitude of the flight can be decreased to obtain a better spacial image resolution. However, this leads to a longer path and longer time.

Definition 3.3.2. Let E^D be energy required for a safe return. This energy is equal to sum of energy required to fly a euclidean distance from a current place to the starting position and energy required to land on the ground [6]:

$$E^D(v^*(D(t)), D(t)) = E_a(v^*(D(t)), D(t)) + E_{desc}(h) \tag{3.5}$$

where $D(t)$ is the euclidean distance between current and launch position in time t and E_a is energy required to fly a distance $D(t)$ with the optimal speed $v^*(D(t))$.

3.4 Energy-aware Coverage with the Lawnmower Pattern

If the given area is too large, energy constraints from Section 3.3 have to be taken into account. This section describes the ECCPP problem with a single starting position with the back and forth motion described before, where the UAV is not able to cover the whole region with only single flight. The condition $E^D(v^*(D(t)), D(t)) \leq E(t)$ has to be always fulfilled during the coverage mission, where $E(t)$ is current energy at time t. If the condition $E_{path} \leq E_{total}$ is also fulfilled, the ECCPP approach is not necessary, because the area can be covered with a single flight.

Definition 3.4.1. Let $A \subset \mathbb{R}^2$ be an area decomposed in a finite set of regular cells $C = \{c_1, ..., c_n\}$ such that $A \approx \bigcup_{i=1}^{n} c_i$. Let us consider coverage path P using the back and forth motion for the area A. P can be described as a set of sorted cells $P = \{p_1, ..., p_n\}$, where $p_i \in C$, $C = \bigcup_{i=1}^{n} p_i$, $c_i \cap c_j = \emptyset$ for integers $1 \leq i, j \leq n$ with $i \neq j$ and where p_i is the i_{th} visited cell. Let p_1 be the starting position of a drone. Thus, P fully represents the planned path for a UAV with *unlimited* battery capacity, where no cell is visited more than once.

According to [2], P is optimal iff number of turns is minimal.

Definition 3.4.2. Let P^* be the optimal coverage path for the back and forth motion starting in cell p_1.

3.4.1 Parameters

The following parameters must be considered subject in the coverage mission with battery limits and the coverage path P^*:

t Coverage time to completion of the path P^*

t' Flight time of UAV over locations not included in the path P^*

k Number of used batteries

α Time needed for a landing, changing a battery and next take off of a vehicle

μ Importance of full battery usage

T Coverage time to completion of the area A

3.4.2 Problem Statement

Definition 3.4.3. Let D_i be a distance travelled on the i_{th} change of a battery from a cell p_j to the starting position p_1. Let D'_i be a distance travelled after the i_{th} change of the battery from the starting position to the cell p_{j+1}, so that $1 \leq i \leq k-1$ and $1 \leq j < n$. UAVs can fly directly to the starting position for a new battery. Assuming simplified problem where a UAV has a constant speed v, we can write

$$t' = v \sum_{i=1}^{k-1} (D_i + D'_i)$$

For the simplified problem, where the acceleration is omitted, duration of the mission can be described as

$$T = t + v \sum_{i=1}^{k-1} (D_i + D'_i) + (k-1)\alpha$$

Because t is time for the shortest P^*, t is considered as the best possible. Therefore minimisation criterion for the shortest time T^* is

$$min\{v \sum_{i=1}^{k-1} (D_i + D'_i) + (k-1)\alpha\}$$

Thus,

$$T^* = t + min\{v \sum_{i=1}^{k-1} (D_i + D'_i) + (k-1)\alpha\} \tag{3.6}$$

(a) Battery replacement

(b) More effective battery replacement

Figure 3.5: Comparison of battery replacement approaches.

Figure 3.5 shows an example of the battery replacement optimisation. This criterion can lead to wasting of batteries in some cases. If we prefer a lower number of used batteries at the expense of better time, we can use the μ parameter in (3.6). Then the optimal cost c^* of the mission is

$$c^* = t + min\{v \sum_{i=1}^{k-1} (D_i + D'_i) + \mu(k-1)\alpha\}, \ \mu \geq 1 \tag{3.7}$$

(a) Flight where 3 batteries are used.

(b) Flight where two batteries are used.

Figure 3.6: Different battery replacement approach comparison. Figure 3.6a provides shorter flight time while case 3.6b uses lower number of batteries

Figures 3.6a and 3.6b show such an example of different battery replacement approaches. We can assume that Figure 3.6a shows a path plan with a lower time completion. Note that this example is more probable if the time required for the battery replacement is low. μ in the criterion has to be used if a path plan with a lower number of used batteries is preferred.

3.5 Multiple Starting Positions and Area Decomposition

Sometimes a drone operator is forced to use multiple starting positions. A good example is an area where a UAV is not even able to get to the other side of the region with its battery capacity. Another example can be prohibited airspace in the middle of the area, so a UAV cannot fly from the one side to the other.

However, using more starting positions for a CPP as in Figure 3.7 can be useful even if it is not necessary. It can be more efficient to cover part of a region from a single location, pack the UAV, move to another launch position and cover the other part of the area. On the other hand, repeated starts from a single location are better if the cost of travel between

possible starting positions is high. Since we minimise total time of the coverage mission in the criterion, the cost matrix should be represented by the time required to travel from one position to another. We suppose the cost of travel is the same in opposite directions. Therefore, cost matrix is symmetrical.

Definition 3.5.1. Let $S = \{s_1, ..., s_r\}$ be a finite set of subareas, $L = \{l_1, ..., l_t\}$ be a finite set of line segments that divide A. Let r be the number of used starting positions, each subarea is assigned to exactly one launch position and an UAV always starts and lands on the same position. Thus, $A = S \cup L$ [2] where

$$S = \bigcup_{i=1}^{r} s_i \; and \; L = \bigcup_{i=1}^{t} l_i$$

Let ${h_i}^*$ be the optimal time of coverage of s_i, m_i be the time required to move from the launch position of a region s_i to the launch position of a region s_{i+1}. Finding the optimal sequence of visited subregions can be modelled as the TSP. Assuming the starting positions order is optimal, the optimal time of coverage H^* of S with the current selected launch position is

$$H^* = \sum_{i=1}^{r-1} ({h_i}^* + m_i) + {h_r}^* \tag{3.8}$$

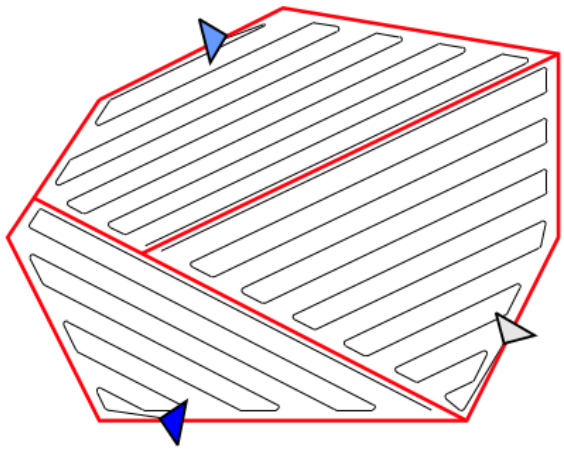
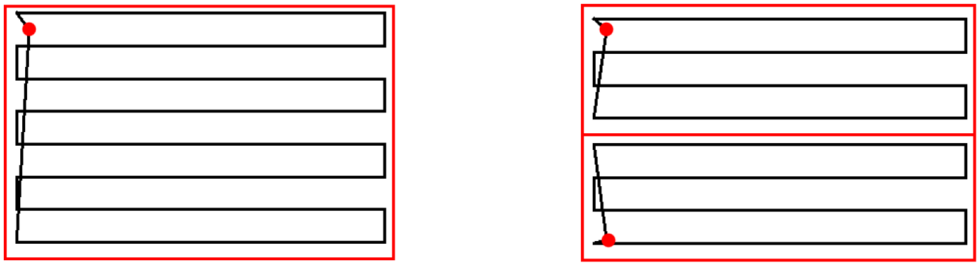


Figure 3.7: Area decomposition with two line segments and optimal sweep motion direction for every subarea. Starting positions are represented by triangles. [18]

Dividing the area leads to a higher number of turns and additional image overlapping, see Figure 3.8. Thus, if a UAV can cover a given convex area with a single flight with the Lawnmower pattern, dividing the area and using multiple starting positions will always provide path plan with a higher time of coverage.



(a) CPP without decomposition.

(b) CPP with decomposition.

Figure 3.8: Decomposition of a convex area always leads to a higher number of turns and usually to a higher image overlapping.

Therefore, decomposition of the area can be beneficial only if multiple flights are predicted. Decomposition of the area can provide a solution with shorter flights needed for battery replacements. Thus, it may decrease the number of needed batteries and total time for the coverage mission.

3.6 Summary and Goal Specification

This thesis addresses a problem of minimisation of the time of coverage required to cover the area of interest using a UAV. The UAVs flight time limitations and possible multiple starting positions are taken into account.

The problem consists of optimal decomposition of the area of interest by the principles discussed in Sections 3.5 and 3.5, finding the optimal visiting order of the starting positions by solving a small TSP problem, and proposing an optimal coverage path plans for every subarea according to Sections 3.2, 3.3 and 3.4. The goal of the work is to provide such an algorithm for a convex area without obstacles and compare the results with a method where only a single launch position is used.

Chapter 4

The Proposed Approach

In this chapter, we propose a solution to the ECCPP problem defined in Chapter 3. The proposed approach decomposes the problem to individual subproblems. Figure 4.1 shows the sequence of subproblems running for all combinations of available launch positions. Cutting lines are used inside of the "Find the Optimal Area Decomposition" step to decompose the area. Figure 4.2 divides this step to additional subproblems. The steps in Figure 4.2 are done repeatedly for different positions of the cutting lines.

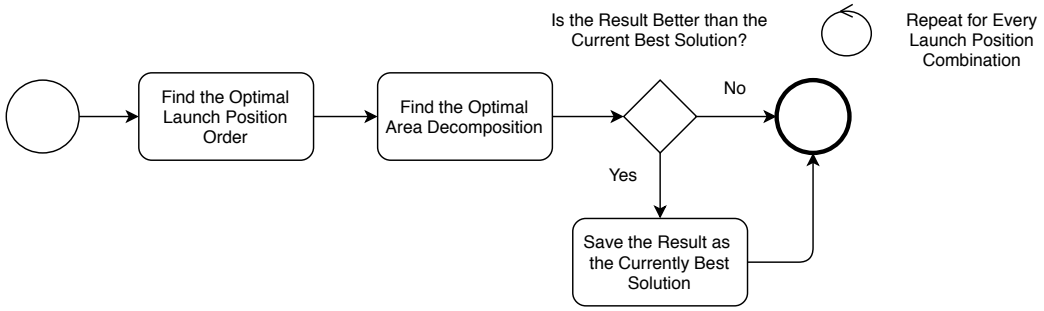


Figure 4.1: Sequence of subproblems illustration. A polygon and chosen launch positions are on the input.

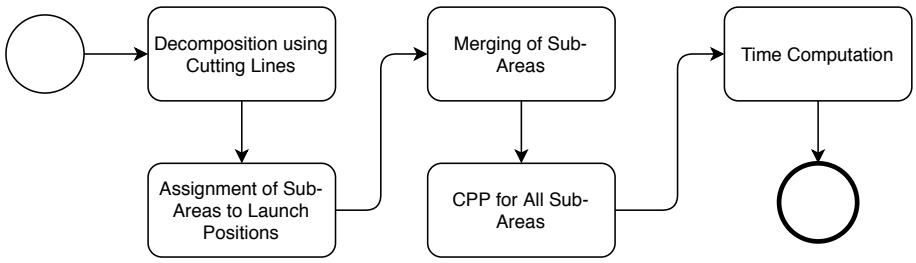


Figure 4.2: Individual subproblems during the search of the optimal area decomposition. A polygon, selected launch positions and tested cutting lines are on the input.

Approaches to these subproblems are discussed in the following sections. First, we discuss the algorithm addressing the ECCPP with a single launch position. Before applying the energy-aware Lawnmower pattern, the area of interest is decomposed between a finite set of launch positions using line segments. The proposed algorithm for the ECCPP problem is then applied on every subarea. The coverage times are then summed up along with the time required for transport between the launch positions. Optimal visiting order of used launch positions is found with a TSP solver described in Section 4.3. Section 4.4 covers computation of the time estimation for the coverage mission. The coverage time of the mission is used as the evaluation of a solution during the search for the optimal decomposition in Section 4.5.

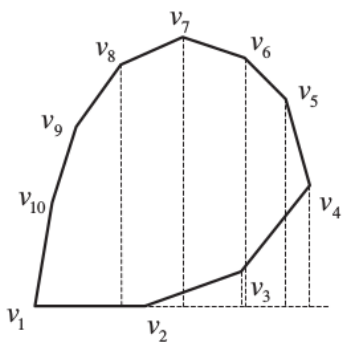
4.1 Energy-aware CPP for a Convex Polygon

A modified Lawnmower pattern approach was used in this work along with a rectangular grid approximation of the area for a simplification. According to Section 3.2, the width of a convex polygon P is computed to find the optimal sweep motion direction. The rectangular grid then represents images taken by a camera during scanning. Unlike in the common zig-zag pattern algorithms, battery replacements are expected. Therefore, constraints discussed in Section 3.3 are taken into account.

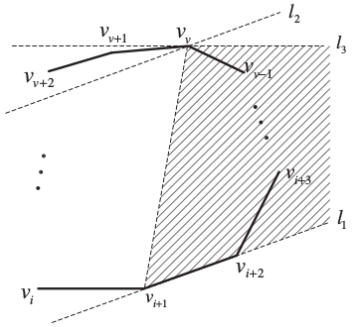
4.1.1 Width of a Convex Polygon

The width appears only in Vertex-Edge style span (Edge-Edge style is considered as a special case of Vertex-Edge style span, as is shown in Figure 3.3).

Theorem 4.1.1. Let $V = \{v_1, v_2, ..., v_n\}$ be a set of vertices of P ordered in a counter-clockwise order according to Cartesian coordinates. Let the vertex v_v be the antipodal vertex for the edge $\overline{v_i v_{i+1}}$. Then the antipodal vertex of the subsequent edge $\overline{v_{i+1} v_{i+2}}$ is either v_v or the following vertex (vertices, if collinear vertices are possible). [17]



(a) Span calculation where the antipodal vertex for edge $\overline{v_1 v_2}$ is the vertex v_7
.



(b) The sketch map where v_v is the antipodal vertex of $\overline{v_i v_{i+1}}$ and where the hatched region shows vertices which can not be the antipodal vertex of $\overline{v_{i+1} v_{i+2}}$.

Figure 4.3: Span calculation [17]

Definition 4.1.1. Let x_v and y_v be the coordinates of a vertex v_v in Cartesian system and $Ax + By + C = 0$ be the linear equation in the general form of a line corresponding to edge $\overline{v_i v_{i+1}}$. Than the distance d between the vertex v_v and edge $\overline{v_i v_{i+1}}$ is computed in the work as follows:

$$d = \frac{|Ax_v + By_v + C|}{\sqrt{x_v^2 + y_v^2}} \tag{4.1}$$

Based on Theorem 4.1.1 the width of the polygon is calculated using Algorithm 1:

Algorithm 1 Polygon Width Calculation

Input: Polygon P
Output: Width of the polygon P

1: $n \leftarrow$ number of edges in P
2: $v_v \leftarrow$ antipodal vertex of edge $\overline{v_1 v_2}$
3: $width \leftarrow$ distance between v_v and $\overline{v_1 v_2}$
4: **for** `i = 2 to n` **do**
5: $v_v \leftarrow$ antipodal vertex from subsequent vertices of edge $\overline{v_i v_{(i+1)\%(n+1)}}$
6: $span \leftarrow$ distance between v_v and $\overline{v_i v_{(i+1)\%(n+1)}}$
7: **if** $span < width$ **then**
8: $width = span$
9: **end if**
10: **end for**
11: **return** $width$

This algorithm uses two pointers. The first points to the edges of the polygon, and traverses them exactly once. The second is a vertex pointer and traverses them not more than twice. Therefore, this algorithm has the time complexity $O(n)$.

4.1.2 Grid Approximation

A polygon representing an area can be split through an approximate cellular decomposition where the polygon is sampled as a regular grid [2]. Centres of the cells are considered as waypoints, and the cell size matches the size of the camera footprint 3.2. The best approximation of a line segment is obtained using the procedure called rasterization [24].

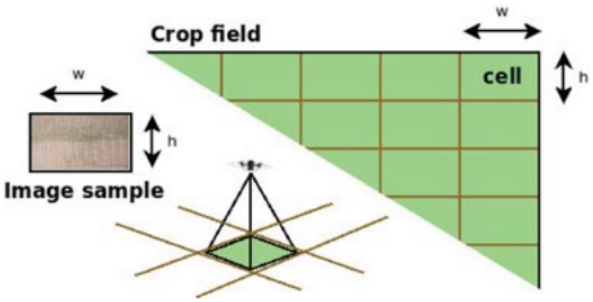


Figure 4.4: Area sampling schematic. Cells of the grid represents footprint of the camera. [2]

Grid approximation was used for a more straightforward usage of the energy-aware Lawnmower pattern. First, the polygon is rotated with a rotation matrix to make the sweep motion vertical. Then the grid approximation is used, so the grid is turned according to the sweep motion of the UAV as in Figure 4.5.

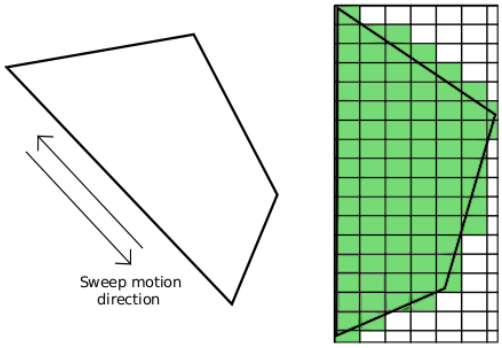


Figure 4.5: A demonstration of the grid approximation after the rotation.

4.1.3 Energy-aware Lawnmower Pattern

Since the sweep motion is used, two possible paths are covering the convex area, see Figure (4.6). Each of them can be entered from two sides, giving us four possible path plans in total. The path always starts and ends in the same launch position. Therefore, the shortest of the paths can be selected.

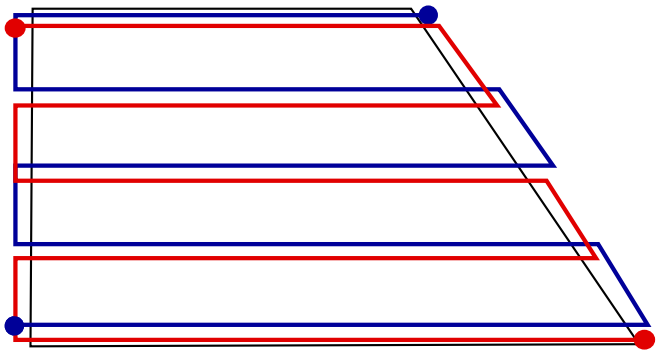


Figure 4.6: Possible paths for the Lawnmower pattern with the first path in red and the second in blue.

The proposed algorithm for the CPP of convex areas works according to principles discussed in Section 3.2, 3.3 and 3.4. The area of interest is approximated with a rectangular grid where the centre of each cell represents a way-point for the UAV. The UAV continues flying only if the UAV has enough energy to fly to the subsequent way-point and for a safe return to the launch position (see Figure 4.7). If the remaining energy is lower, UAV flies to the starting position immediately, the operator replaces the battery, and the UAV continues to the following way-point.

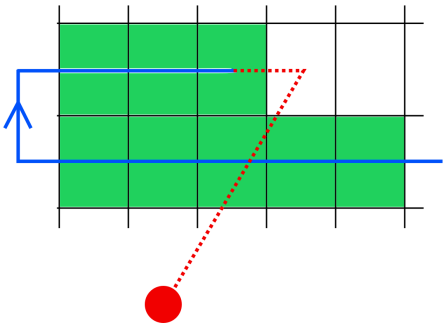


Figure 4.7: Computation of necessary energy for the next way-point. Already travelled path in blue along with the visited cells from the grid in green. The UAV has to be able to fly the distance expressed with the dotted line in red to get permission to go to the next way-point.

After the coverage path plan is found on the grid described in Section 4.1.2, the path way-points are rotated back to the original polygon position with the same rotation matrix.

4.2 Area Decomposition

4.2.1 Decomposition with Cutting Lines

The area decomposition is performed by cutting lines between each two launch positions. Similar approach for the area decomposition was proposed by Maze and Ollero [18] with an algorithm based on divide-and-conquer approach using $n-1$ cutting lines, where n is the number of launch positions.

(a) Divide-and-conquer style decomposition with $n-1$ line segments

.

(b) The implemented decomposition with cutting lines between each two launch positions.

Figure 4.8: An example comparison between Divide-and-conquer style and proposed decomposition.

The advantage of the approach shown in Figure 4.8a is in its simplicity. The area decomposition algorithm produces exactly n convex subareas already assigned to the launch positions. The area decomposition method proposed in this work (see Figure 4.8b) provides more options for dividing the area, and thus, better coverage efficiency optimisation. However, the number of produced subareas with these cutting lines is always equal to or greater than $2\binom{n}{2}$. Each of these convex subareas then needs to be assigned to some launch position and merged with others to a single subarea, ideally convex shaped. The quality of the found solution was preferred over the fast computation.

4.2.2 Assignment of Subareas to Launch Positions

After the area is decomposed into smaller subareas, the subareas need to be assigned to the launch positions. Let n be the number of launch positions and d the number of created polygons after cutting the area. Assume that every used launch position lies inside one of these polygons and the polygon is assigned to this position. Then the number of possible combinations C of assignment of the remaining polygons is:

$$C = n^{(d-n)} \tag{4.2}$$

Computation requirements for trying all the combinations and computing the coverage time would be extremely high. Therefore, just a single simple assignment is used with Algorithm 2. Every used cutting line belongs to some two launch positions A and B and divides subareas between them (see Figure 4.9).

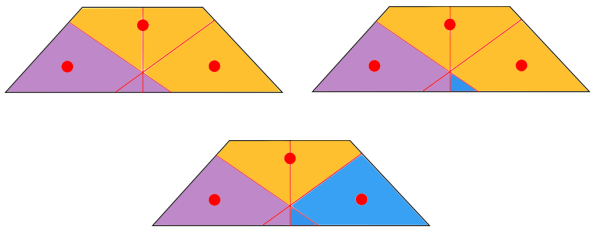


Figure 4.9: Demonstration of the assignment of subareas. First, subareas are divided between purple and orange, then between purple and blue and at the end, between orange and blue.

Algorithm 2 Launch Position Assignment

Input: Set of polygons P sorted by x coordinates, set of cutting lines L

for all $l \in L$ **do**
  for all $p \in P$ **do**
    $launchPosition \leftarrow$ current launch position of p
    if (centroid of p is on the same side of l as launch position $l.getA()$) and ($launchPosition$ is $null$ or $launchPosition$ is $l.getB()$) **then**
      $launchPosition \leftarrow l.getA()$
    else if centroid of p is **not** on the same side of l as launch position **then**
      if centroid of p is on the same side of l as launch position $l.getB()$ **then**
        if $launchPosition$ is $null$ or $launchPosition$ is $l.getA()$ **then**
          $launchPosition \leftarrow l.getB()$
        end if
      else if $launchPosition$ is null **then**
        $launchPosition \leftarrow$ the closer launch position to the centroid of P
      end if
    end if
  end for
end for

4.2.3 Merging of Polygons

It is desirable to create only one convex subarea for every launch position so the energy-aware Lawnmower pattern method described in Section 4.1.3 can be used. The connection of smaller subareas into one subarea for a launch position is done using Graham's Scan method [4] which finds the convex hull of the finite set of vertices forming the subarea (see Figure 4.10). If some concave subareas are created during the decomposition, they are reshaped into convex areas. Therefore, the merge of the subareas is independent of the methods used for the area decomposition and the usage of the sweep motion method will always be possible.

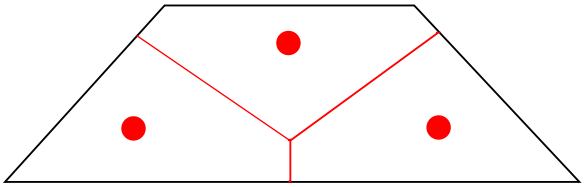


Figure 4.10: Subareas after the merge. Every launch position has exactly one assigned subarea.

Graham's Scan provides convex hulls in $O(nlog(n))$ [4]. The procedure takes as an input a set of vertices V, where $|V| > 3$ and maintains a stack S with candidate vertices using "rotational sweep" technique. It pops every vertex from the stack which is not in the convex hull of V. The final state of S contains exactly the vertices of the convex hull of V in counterclockwise order. Let *head* is the top of the stack, *middle* the entry below the top of the stack. Graham's scan is described with Algorithm 3.

Algorithm 3 Graham's Scan

Input: Set of vertices V
Output: Convex Hull of V

if |V| $<$ 3 or all vertices in V are collinear **then**
    return "Convex Hull is not possible."
end if
$S \leftarrow$ an empty stack
lowest $\leftarrow$ vertex from V with the minimum y-coordinate
sorted $= v_1 \dots v_n \leftarrow$ remaining sorted vertices by polar angle in counterclockwise order around *lowest*
push(*lowest*, S)
push(v_1, S)
push(v_2, S)
for $i = 3$ to n **do**
    while the angle formed by *head*, *middle* and v_i makes a non-left turn **do**
        pop(S)
    end while
    push(v_i, S)
end for
return S

4.3 Launch Position Visiting Order Estimation

The problem of finding the most time-efficient visiting order of the selected launch positions can be formulated as an instance of TSP. Each of the used starting locations is to be visited exactly once. A round trip is usually considered. However, the first and the last visited launch position is different in this case, see Figure 4.11. Symmetric TSP is considered,

so the time needed to get from position p_1 to position p_2 is the same as to get from a position p_2 to p_1. We utilised the implementation of the algorithm provided by the Java Metaheuristics Search Framework (JAMES) [12]. The parallel tempering algorithm [8] was applied to optimise the sequence of launch positions using a neighbourhood that reverses subseries of the launch positions in given solutions. Although the expected number of used launch position is not high, this algorithm is applicable even to bigger problems.

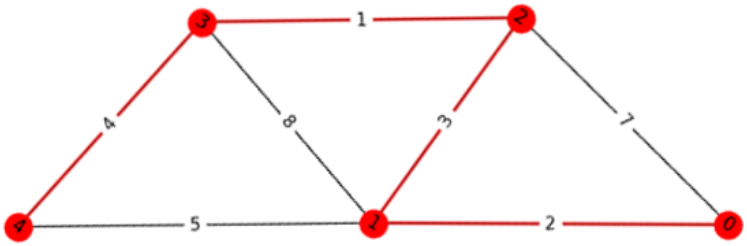


Figure 4.11: Travelling Salesman Problem example without the round trip. The red line shows the chosen path.

4.4 Time Computation

Complete time of a coverage mission consists of the UAV flight time and time required for UAV preparation. First, flight times are computed for every subregion. The proposed algorithm for the ECCPP problem returns way-points for the UAV, so total flight distance and the flight time is known. After that, the ground travel time t_{travel} is added along with time t_{pack} and t_{check} needed for UAV preparation and $t_{battery}$ for battery replacements. Because the flight time is computed separately, the ground travel time is precomputed for all combinations of the possible used launch positions.

Let n be the number of launch positions, t_i the flight time needed to cover area i, $i = 1, \ldots, n$, k be the number of used batteries and all the other parameters be as in the description in Section 5.1. Then complete time of the coverage mission is computed as:

$$t = \sum_{i=1}^{n} t_i + t_{travel} + n(2t_{pack}) + k(t_{battery} + t_{check}) \tag{4.3}$$

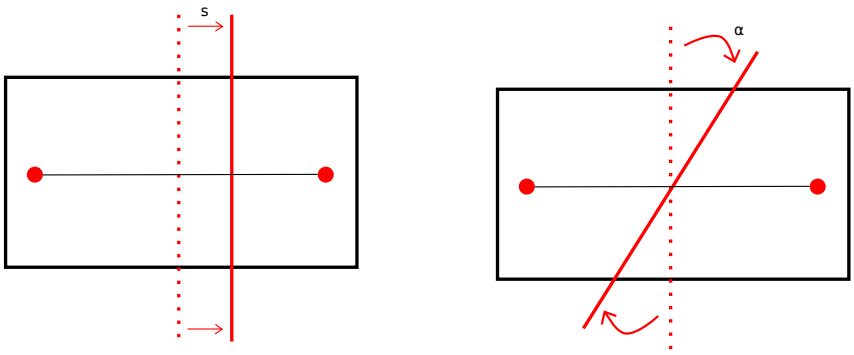
4.5 Optimisation

The criterion that is being minimised during the optimisation process is the overall time of the coverage mission. Every combination of available launch positions is tested, whether it provides a better time of the mission than the other combinations. Therefore, if n is the number of launch positions, then the number of tested combinations is:

$$\sum_{i=1}^{n} \binom{n}{i} \tag{4.4}$$

If only one launch position is used, energy-aware Lawnmower pattern described in Section 4.1.3 is applied. Otherwise, the area decomposition is made by principles in Section 4.2.1 using Variable Neighbourhood Search (VNS) [19] with two neighbourhoods defined by possible moves. The possible moves are:

- Shift of a cutting line by a value s, see in Figure 4.12a
- Rotation of a cutting line by a value α around the centre between intersections with the polygon, see in Figure 4.12b



(a) Shift of a cutting line

(b) Rotation of a cutting line.

Figure 4.12: Possible moves of cutting lines.

Steepest descent algorithm (also referred to as hill climbing) minimising the time of the coverage mission is used within these neighbourhoods. Again, the JAMES framework was utilised for the implementation of the search algorithms. The search starts in default initial solution - the area where each cutting line is the perpendicular bisector of a line connecting two launch positions as in Figure 4.8b. Then the search tries every solution from the neighbourhood and always accepts the best solution as the new current solution. The neighbourhood is defined by possible moves of bisectors used for the area decomposition.

Le S be the maximal possible shift of a cutting line and l be the number of cutting lines. Then since the cutting line can be shifted to both sides, the total number of possible moves w in this neighbourhood is:

$$w = 2 \cdot l \cdot S/s \tag{4.5}$$

As w increases, the precision of the search improves, because the probability of ending in a local minimum is lower. However, computation requirements then increase too. The total number of possible moves in the neighbourhood with rotations can be computed in the same way. A cutting line moved outside of a polygon is considered as unused and has no impact on the final solution.

4.6 Summary

We have described the proposed approach for the solution of the problem formulated in Chapter 3. First, we have introduced the method for the energy-aware CPP problem with a

single starting position. The maximal flight time of the UAV has been taken into account for the general Lawnmower pattern in this method. Then we have focused on the decomposition of large areas between multiple launch positions. This decomposition consists of three parts. First, the polygon is divided by a set of cutting lines. Next, the subareas are assigned to launch positions. Lastly, merging the subareas using the Graham's Scan algorithm to find their convex hull to be covered from a single location. Then we described the way how the total time of the coverage mission is computed. Finally, optimisation using JAMES's VNS with steepest descent local search to complete the coverage mission was described. Another usage of JAMES's local search metaheuristics was for solving the problem of launch positions visiting order formulated as an instance of the TSP.

Chapter 5

Experimental Results

In this chapter, we present the experimental results that validate the proposed solution for addressing the ECCPP problem. Experiments have been performed with realistic parameters values constant throughout all the scenarios. Only the shape and size of the areas are different in the scenarios together with the launch positions layout. Section 5.1 describes used parameters and their influence on the CPP. Section 5.2 includes experiments which verify the correctness of the solution with examples where the solution is easily predictable. Section 5.3 then includes experiments where the solution is not so obvious and compares the proposed solution with the solution where only a single launch position is used.

5.1 Parameters

Table 5.1 shows parameters and their default values used for the experiments.

Parameter	**Unit**	**Value**	**Symbol**
SPEED_DURING_SCANNING	$m.s^{-1}$	3	v_{travel}
UAV_SPEED	$m.s^{-1}$	6	v_{flight}
TRAVELLING_SPEED	$m.s^{-1}$	9	v_{scan}
BATTERY_REPLACEMENT_TIME	s	300	$t_{battery}$
CHECK_UP_TIME	s	300	t_{check}
PACK_UAV_TIME	s	180	t_{pack}
BATTERY_CAPACITY	mAh	3000	$E_{capacity}$
ENERGY_COST_PER_METER	$mAh.m^{-1}$	0.72	e_{cost}
ALTITUDE	m	100	h
HOR_FOV	°	19.0	fov_2
VER_FOV	°	12.7	fov_1
PITCH_ANGLE	°	0	θ
AF	°	90	af

Table 5.1: Parameters

- **SPEED_DURING_SCANNING** is the average speed of the UAV during scanning. It depends on altitude and actual visibility.
- **UAV_SPEED** is the average speed of the UAV while the UAV is not scanning the given area.
- **TRAVELLING_SPEED** is the average travelling speed on the ground.
- **BATTERY_REPLACEMENT_TIME** is the time required for a battery replacement. This parameter does not include the possibly necessary system check-up of the UAV.
- **CHECK_UP_TIME** is the time of the UAV's system check-up required after replacing a battery.
- **PACK_UAV_TIME** is the time required to pack the UAV and prepare it for transportation.
- **BATTERY_CAPACITY** is the capacity of used batteries. We assume that all batteries for a coverage mission are the same.
- **ENERGY_COST_PER_METER** is the energy loss per meter. The parameter depends on the used aircraft and weather conditions. The default value was computed to be able to cover an area of approximately 10 ha with a single 20-minute flight.
- **ALTITUDE** is the flight altitude. Better visibility allows higher altitude, which leads to a shorter time of coverage completion.
- **HOR_FOV** is the horizontal field of view described in Section 3.1.2. The used value corresponds to lens focal length 70 mm.
- **VER_FOV** is the vertical field of view described in Section 3.1.2. The used value corresponds to lens focal length 70 mm.
- **PITCH_ANGLE** is the angle shown in Figure 3.1 in range $[0, \pi/2]$.
- **AF** is the front-mounted angle which is the included angle between the longitudinal axis of the UAV and the bisector of vertical field of view. θ equal to $0°$ with af equal to $90°$ approximately corresponds to the rotary wing (VTOL) type UAVs.

With a few exceptions, which are always mentioned, all used variable parameters are as in Table 5.1 and the distance between two launch positions is their Euclidean distance. The UAV in the experiments is able to cover an area a little over 10 ha with a single battery.

The further figures use the following notation:

- Black line - Area/Subarea borders
- Purple line - Coverage path plan for the first subarea
- Blue line - Coverage path plan for the second subarea

- Yellow line - Coverage path plan for the third subarea
- Orange line - Connecting line between used launch positions
- Green point - Used starting position
- Red point - Unused starting position

5.2 Basic Verification

This section addresses experiments performed to verify the correctness of the proposed approach in simple scenarios.

5.2.1 Single Flight Area

According to Figure 3.5, a given area should not be decomposed between multiple launch positions if it can be covered with a single flight. Several tests have been done with different shapes of the area sized around 10 ha and the coverage path plan was always planned with a single flight as predicted.

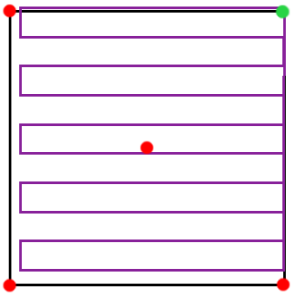


Figure 5.1: A small area example demonstrating multiple unused launch positions in red with the single flight line in purple

We can see an example area of size 10 ha in Figure 5.1. Only one out of the five launch positions was used for the coverage as predicted because the UAV can cover the area with a single flight. The UAV flight time was estimated to 19 min 21 sec and time of the whole coverage mission with considering the time to set-up, take-off, land, and pack the UAV, was estimated to 32 min 21 sec.

5.2.2 Area with a High Cost of Transportation

Multiple starting positions should not be chosen if the cost of transportation between them is too high. More precisely, the sum of the saved time on manipulation and the flight of the UAV must be higher than the time required for moving between launch positions.

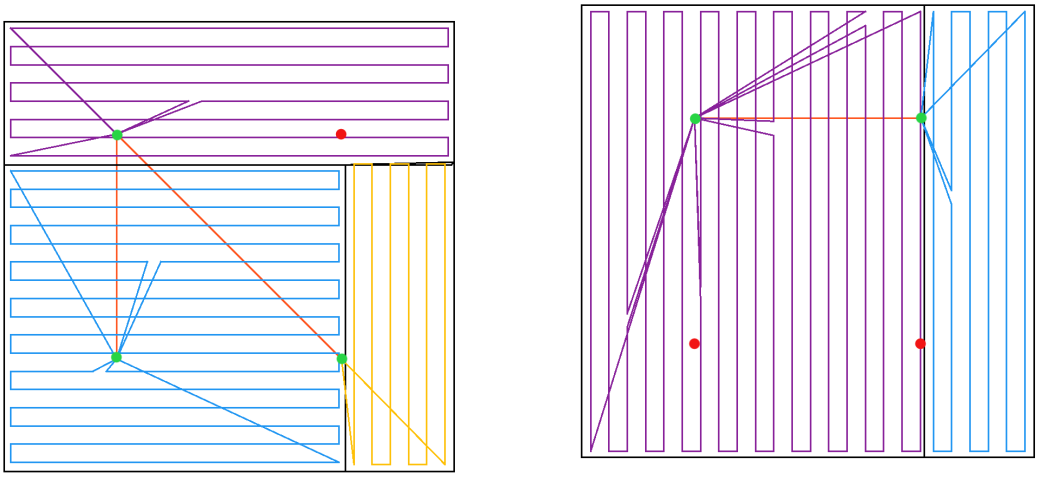
The same area and launch position layout as in Section 5.3.5 was used to verify the correctness of the starting position selection. Figure 5.3 shows an example where the decomposition provides shorter time for the mission, with the total distance travelled on the ground being 832 m. Figure 5.3a shows the solution where $v_{travel} > 3.8$ km.h^{-1}. Figure 5.3b then shows the experiment where $v_{travel} < 3{,}8$ km.h^{-1}. A solution with a lower number of starting positions is preferable over those with the same coverage time and higher number of utilised starting positions. These results have been expected.

5.2.3 Launch Positions Selection

The travel time between the ground locations influences the final solution. A high cost of travelling to a launch position should lead to changing the visiting order of the launch positions or to omitting the launch position completely.

In the scenario in Section 5.3.1, the time of transporting the UAV between launch positions depends linearly on their Euclidean distance. The launch positions are placed in the corners of a square of an edge length 416 m. Figure 5.2a shows the example where the time of the transport along diagonals between launch positions is two times lower than along the edges of the square. The proposed method changes the visiting order as expected.

Figure 5.2b depicts an example where the cost of moving to the bottom positions is ten times higher than their Euclidean distance. The proposed method than correctly omits the bottom launch positions.



(a) Area with lower cost of move on diagonals

(b) Area with high cost of move to the bottom launch positions

Figure 5.2: A square shaped area example demonstrating the launch position selection and their visiting order depending on the cost matrix.

5.3 Comparison with Simple Solution

The following experiments show a comparison of the proposed method with the traditional Lawnmower pattern approach that utilises a single launch position. The launch position,

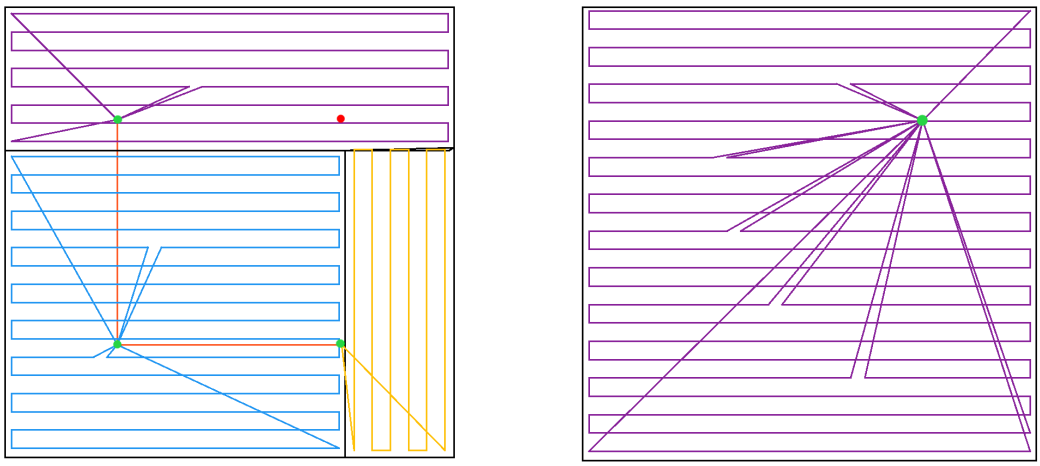
where the estimated coverage time was the shortest, is selected for the traditional Lawnmower pattern. The following tables show a comparison of various mission parameters. These parameters are:

- Used Launch Pos. - The number of used launch positions for the mission
- Used Batteries - The number of batteries used for the mission
- Manipulation - Time required for the battery replacement, system check-up, and packing of the UAV
- Transport - The necessary time to travel between the selected launch positions
- Flight - The Flight time of the UAV
- Total Time - Total time of the mission

The values in the tables are in the following format: **[**Result with multiple launch position/Result with a single launch position **(**Difference between the solutions**)]**. Tests have been performed on a machine Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz.

5.3.1 Square Area Scenario

Square shaped area of size 69,22 ha with four available starting positions, see Figure 5.3. The launch positions are placed in corners of a square of an edge length 416 m. The computation time was 84,8 s.



(a) Usage of multiple launch positions (b) CPP with a single launch position

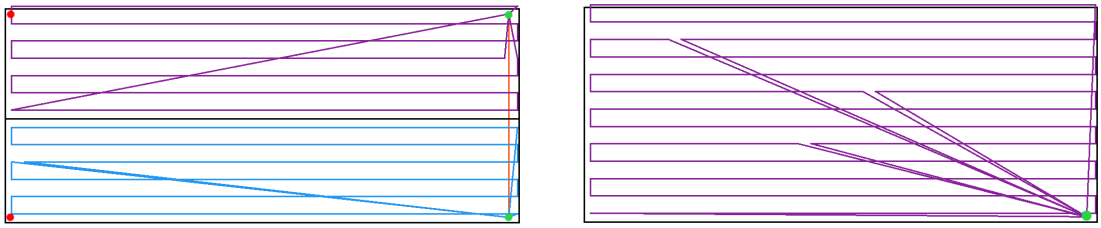
Figure 5.3: Square shaped area CPP comparison

Used Launch Pos.	3/1
Used Batteries	6/7 (-1)
Manipulation	1:09:00/1:13:00 (-0:04:00)
Transport	0:01:32/- (+0:01:32)
Flight	2:03:38/02:12:48 (-0:09:10)
Total Time	3:14:10/03:25:48 (-0:11:38)

Table 5.2: Square area example results

5.3.2 Rectangle Area Scenario

Rectangle shaped area of size 41,60 ha with four available launch positions, see Figure 5.4. The launch positions are situated in corners of the rectangle. The computation time was 47,6 s.



(a) Usage of multiple launch positions (b) CPP with a single launch position

Figure 5.4: Rectangle shaped area CPP comparison

Used Launch Pos.	2/1
Used Batteries	4/5 (-1)
Manipulation	0:46:00/0:53:00 (-0:07:00)
Transport	0:00:44/- (+0:00:44)
Flight	1:22:03/1:27:29 (-0:05:26)
Total Time	2:08:47/2:20:29 (-0:11:42)

Table 5.3: Rectangle area example results

5.3.3 Triangle Area Scenario

Triangle shaped area of size 64,0 ha with three launch positions available, see Figure 5.5. The computation time was 24,5 s.

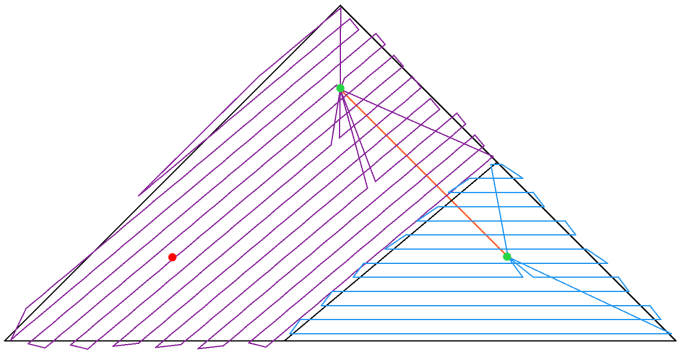


Figure 5.5: Triangle shaped area example

Used Launch Pos.	2/1
Used Batteries	6/7 (-1)
Manipulation	1:06:00/1:13:00 (-0:07:00)
Transport	0:01:02/- (+0:01:02)
Flight	2:07:17/2:14:09 (-0:06:52)
Total Time	3:14:19/3:27:09 (-0:12:50)

Table 5.4: Triangle area example results

5.3.4 Hexagon Area Scenario

Hexagon shaped area of size 64,0 ha with two launch positions available, see Figure 5.6. The computation time was 7,8 s.

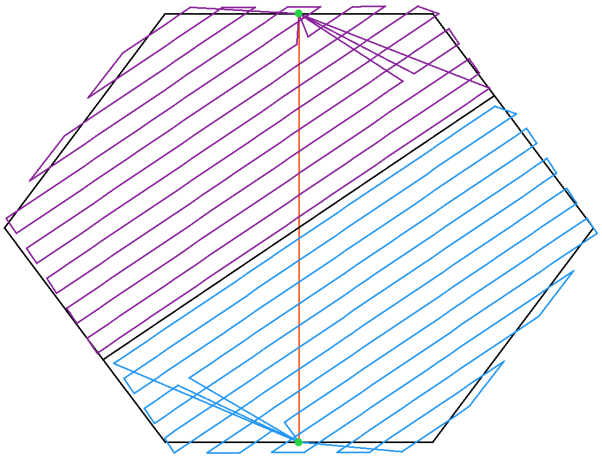


Figure 5.6: Hexagon shaped area example

Used Launch Pos.	2/1
Used Batteries	6/7 (-1)
Manipulation	1:06:00/1:13:00 (-0:07:00)
Transport	0:01:28/- (+0:01:28)
Flight	2:03:28/2:14:08 (-0:10:40)
Total Time	3:10:56/3:27:08 (-0:16:12)

Table 5.5: Hexagon area example results

5.3.5 Large Area Scenario

Large rectangle shaped area example of size 200,0 ha with four launch positions available, see Figure 5.7. The launch positions are situated in the corners of the rectangle. The UAV is unable to cover the given area with a single flight in this example. Therefore, there is no comparison, and only a solution with multiple launch positions is proposed. The computation time was 250 s.

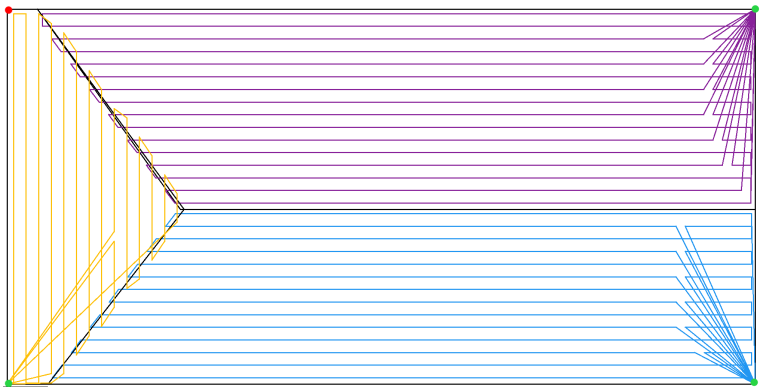


Figure 5.7: A large area example where the UAV is unable to cover the area from a single launch position.

Used Launch Pos.	3	**Used Batteries**	18
Manipulation	3:09:00	**Transport**	0:05:33
Flight	6:13:41	**Total Time**	9:28:14

Table 5.6: Large area results

5.4 Summary

In this chapter, we have verified the proposed approach to solving the ECCPP problem. First, variable parameters influencing the CPP have been described in 5.1.

We have verified the behaviour of the proposed method in several scenarios, and we compared the method with existing simple solution. The method selects the launch positions and their visiting order depending on the cost of the moves between them. Solutions with no area decomposition are correctly provided if an area can be covered with a single flight. We have then compared the proposed method with the common Lawnmower pattern approach. For simplicity, the flight dynamics and precise energy consumption modelling were not taken into account. The experiments have shown that the proposed solution method can save both time and used batteries in comparison to method using the classical Lawnmower pattern.

It can be seen that the predicted UAV returns are minimised thanks to the optimal area decomposition in Figure 5.7. This behaviour was the goal of this approach. Therefore, the system behaves as expected.

Chapter 6

Conclusion

This thesis investigates the problems of area coverage and energy-aware planning. Current works do not address the issue of covering large areas with flight time limitations and the possibility of starting from multiple locations. Therefore, we propose an Energy Constraint Coverage Path Planning (ECCPP) problem, together with a method for its solution. This method plans the coverage path in a convex polygon area with possible multiple starting positions for a UAV. Firstly, we propose the optimisation criterion for the ECCPP which minimises the time required for covering the whole area of interest. Next, we describe the ECCPP method for covering an area from a single starting position. This method uses modified Lawnmower pattern method where the maximal flight time of the UAV is taken into account and uses the strategy of flying along the vertical direction of the width of the polygon. After that, we address an area decomposition method, which divides the area between multiple starting positions. The optimal area decomposition is found using Variable Neighbourhood Search (VNS) with the steepest descent local search using a JAMES framework. The JAMES framework was also utilised for the additional small Travelling Salesman Problem (TSP) which refers to searching for the optimal launch position visiting order.

Finally, several experiments have been done to test the system behaviour and to compare the proposed solution within the simple method for Coverage Path Planning (CPP). In the four scenarios in Section 5.3, the proposed method reduced the total time required for the coverage mission by 7% on average. Also, at least one battery was always saved. Apart from that, the proposed method provides a feasible and effective solution for large areas that the UAV cannot cover from one starting position. Experiments have shown that covering an area from a single starting position is always preferable if the UAV can cover the area of interest with one flight. The decomposition of smaller areas where the number of required batteries is low is also not beneficial, because dividing the area causes additional coverage overlapping and increases the number of turns of the UAV.

6.1 Future Work

As a future work, we plan to modify the proposed method for non-convex areas with possible no-flight zones inside. Although the thesis aims at the usage of a single UAV, the proposed

area decomposition can be applied as a decomposition between multiple UAVs. Comparison of the area decomposition between numerous vehicles with other known approaches is another possible further work. Additional further work concerns the improvement of the ECCPP subproblems such as the grid approximation and launch position assignment to subareas.

Bibliography

[1] G. S. Avellar, G. A. Pereira, L. C. Pimenta, and P. Iscold. Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, 2015.

[2] 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*, 28(5):667–689, 2011.

[3] I. Châari, A. Koubâa, S. Trigui, H. Bennaceur, A. Ammar, and K. Al-Shalfan. Smartpath: An efficient hybrid aco-ga algorithm for solving the global path planning problem of mobile robots. *International Journal of Advanced Robotic Systems*, 11(7):94, 2014.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 33.3: Finding the convex hull. *Introduction to Algorithms*, pages 955–956, 1990.

[5] M. Dakulović, S. Horvatić, and I. Petrović. Complete coverage d* algorithm for path planning of a floor-cleaning mobile robot. *IFAC Proceedings Volumes*, 44(1):5950–5955, 2011.

[6] C. Di Franco and G. Buttazzo. Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *Journal of Intelligent & Robotic Systems*, 83(3-4):445–462, 2016.

[7] T. M. Driscoll. Complete coverage path planning in an agricultural environment. 2011.

[8] D. J. Earl and M. W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[9] J. Everaerts et al. The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(2008):1187–1192, 2008.

[10] M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[11] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.

[12] D. B. Herman, D. G. F., D. M. Geert, and F. Veerle. James: An object-oriented java framework for discrete optimization using local search metaheuristics. *Software: Practice and Experience*, 47(6):921–938.

[13] W. H. Huang. Optimal line-sweep-based decompositions for coverage algorithms. In *ICRA*, 2001.

[14] K. Karydis and V. Kumar. Energetics in robotic flight at small scales. *Interface focus*, 7(1):20160088, 2017.

[15] A. Khan, I. Noreen, and Z. Habib. rage path planning algorithms for non-holonomic mobile robots: Survey and challenges. *J. Inf. Sci. Eng.*, 33(1):101–121, 2017.

[16] D. Li, X. Wang, and T. Sun. Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. *Electronics Letters*, 52(9):699–701, 2016.

[17] Y. Li, H. Chen, M. J. Er, and X. Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, 2011.

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

[19] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.

[20] 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*, 51(3):718–726, 2004.

[21] L. Paull, S. Saeedi, M. Seto, and H. Li. Sensor-driven online coverage planning for autonomous underwater vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(6):1827–1838, 2013.

[22] L. Paull, M. Seto, J. J. Leonard, and H. Li. Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping. *The International Journal of Robotics Research*, 37(1):21–45, 2018.

[23] L. Paull, C. Thibault, A. Nagaty, M. Seto, and H. Li. Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle. *IEEE Transactions on Cybernetics*, 44(9):1605–1618, 2014.

[24] J. Pineda. A parallel algorithm for polygon rasterization. *SIGGRAPH Comput. Graph.*, 22(4):17–20, June 1988.

[25] H. Pirzadeh. *Computational geometry with the rotating calipers.* PhD thesis, McGill University, 1999.

[26] E. Santamaria, F. Segor, and I. Tchouchenkov. Rapid aerial mapping with multiple heterogeneous unmanned vehicles. In *ISCRAM*. Citeseer, 2013.

[27] P. B. Sujit and R. Beard. Multiple uav exploration of an unknown region. *Annals of Mathematics and Artificial Intelligence*, 52(2):335–366, Apr 2008.

[28] A. Trotta, M. Di Felice, K. R. Chowdhury, and L. Bononi. Fly and recharge: Achieving persistent coverage using small unmanned aerial vehicles (suavs). In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[29] J. Valente, A. Barrientos, J. del Cerro, C. Rossi, J. Colorado, D. Sanz, and M. Garzón. Multi-robot visual coverage path planning: Geometrical metamorphosis of the workspace through raster graphics based approaches. In B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, editors, *Computational Science and Its Applications - ICCSA 2011*, pages 58–73, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[30] K. Vicencio, T. Korras, K. A. Bordignon, and I. Gentilini. Energy-optimal path planning for six-rotors on multi-target missions. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2481–2487. IEEE, 2015.

[31] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. Chung. Ba*: an online complete coverage algorithm for cleaning robots. *Applied Intelligence*, 39(2):217–235, Sep 2013.

[32] J. Ware and N. Roy. An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1507–1514. IEEE, 2016.

[33] Fixed wing versus rotary wing for UAV mapping applications, 2016. `https://www.questuav.com/media/case-study`/fixed-wing-versus-rotary-wing-for-uav-mapping-applications/, Accessed: 2018-04-16.

[34] 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*, volume 13, pages 533–538, 1993.

[35] P. Zhou, Z.-m. Wang, Z.-n. Li, and Y. Li. Complete coverage path planning of mobile robot based on dynamic programming algorithm. In *2nd international conference on electronic and mechanical engineering and information technology*, pages 1837–1841, 2012.

Appendix A

List of Abbreviations

ACO Ant Colony Optimisation

AUV Autonomous Underwater Vehicle

CCD* Complete Coverage D*

CCPP Complete Coverage Path Planning

CPP Coverage Path Planning

DFS Depth First Search

ECCPP Energy Constrained Coverage Path Planning

FOV Field of View

JAMES Java Metaheuristics Search Framework

PSO Particle Swarm Optimisation

TSP Travelling Salesman Problem

UAS Unmanned Aerial System

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

VNS Variable Neighbourhood Search

VTOL Vertical Take-Off/Landing

Appendix B

User Guide

The program is implemented in Java. The source code is proposed, so it can be opened with any IDE for Java. The program takes these arguments:

1. path to the file with specified launch positions along with their cost matrix.
2. path to the file with vertices of a convex polygon.
3. (optional) -t parameter symbolising that the cost matrix represents time in seconds. The cost matrix represents distance in meters if this parameter is omitted.

The program runs with a default example if no argument is given.
The file with launch positions is in the following format:

- number of positions
- x and y coordinates of the launch positions
- cost matrix

The file with vertices of the polygon is in the similar format:

- number of vertices
- x and y coordinates

The examples used in scenarios in the thesis are located in "examples" folder.

Appendix C

CD Content

```
.
|– BP_Johanides.pdf – Bachelor's Thesis
|– examples – examples used for test scenarios
|– latex – latex source code
| |– chapters
| | |– conclusion.tex
| | |– experiments.tex
| | |– introduction.tex
| | |– problem_formulation.tex
| | |– related_work.tex
| | |– selected_approach.tex
| |– figures
| |– hyphentex
| |– k336_thesis_macros.sty
| |– main.tex
| |– reference.bib
| |– zav_prace.pdf
|– pom.xml
|– README.txt
|– src – program source code
|– tree.txt

    42 directories, 158 files
```