FEDERAL UNIVERSITY OF PELOTAS
Technology Development Center
Postgraduate Program in Computing

PhD Thesis

Energy-aware Coverage Path Planning for Unmanned Aerial Vehicles

Tauã Milech Cabreira

Pelotas, 2019

Tauã Milech Cabreira

Energy-aware Coverage Path Planning for Unmanned Aerial Vehicles

Thesis submitted to the Postgraduate Program in Computing of the Federal University of Pelotas as a partial requirement to obtain the PhD degree in Computer Science.

Advisor: Prof. Paulo Roberto Ferreira Junior
Coadvisor: Prof. Lisane Brisolara de Brisolara

Pelotas, 2019

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

C117e Cabreira, Tauã Milech

Energy-aware coverage path planning for unmanned aerial vehicles / Tauã Milech Cabreira ; Paulo Roberto Ferreira Junior, orientador ; Lisane Brisolara de Brisolara, coorientador. — Pelotas, 2019.

146 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2019.

1. Coverage path planning. 2. Energy-aware. 3. UAV. 4. Flight pattern. 5. Complete algorithm. I. Ferreira Junior, Paulo Roberto, orient. II. Brisolara, Lisane Brisolara de, coorient. III. Título.

CDD : 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Tauã Milech Cabreira

Energy-aware Coverage Path Planning for Unmanned Aerial Vehicles

Thesis approved as a partial requirement to obtain the PhD degree in Computer Science, Postgraduate Program in Computing, Technology Development Center, Federal University of Pelotas.

Presentation Date: October 25th, 2019

Examination Board:
Prof. Paulo Roberto Ferreira Jr. (advisor)
PhD in Computer Science - Federal University of Rio Grande do Sul.

Prof. Lisane Brisolara de Brisolara. (coadvisor)
PhD in Computer Science - Federal University of Rio Grande do Sul.

Prof. Marco Aurelio Wehrmeister
PhD in Computer Science - Federal University of Rio Grande do Sul / Paderborn University (Germany)

Prof. Alisson Vasconcelos De Brito
PhD in Electrical Engineering - Federal University of Campina Grande / Karlsruhe Institute of Technology (Germany)

Prof. Felipe de Souza Marques
PhD in Computer Science - Federal University of Rio Grande do Sul.

ABSTRACT

CABREIRA, Tauã Milech. **Energy-aware Coverage Path Planning for Unmanned Aerial Vehicles**. Advisor: Paulo Roberto Ferreira Junior. 2019. 146 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2019.

The Coverage Path Planning (CPP) problem is a motion planning subtopic in robotics, where it is necessary to build a path for a robot to explore every location in a given scenario. Unmanned Aerial Vehicles (UAV) have been employed in several application domains related to the CPP problem. Despite the technological progress related to control systems and energy monitoring, one of the most significant limitations of UAVs is endurance, especially in multirotors. These vehicles have a limited payload, which consequently also restricts their batteries. In this way, minimizing the energy consumption of these vehicles is pivotal to prolong and guarantee the coverage mission. The energy consumption is usually associated with the number of turning maneuvers in a given trajectory. Most studies in literature seek to minimize this issue to save energy and, consequently, enhance endurance. However, important factors are usually ignored, such as the acceleration and deceleration phases, optimal speed to travel a straight distance, the entrance speed and turning angle when performing a maneuver. Thus, this work aims to propose energy-aware coverage path planning solutions based on flight patterns, complete algorithms, and pheromone-based methods for regular and irregular-shaped areas containing full and partial information considering the impact of different aspects in the UAV energy consumption, such as traveled distance, mission execution time, turning maneuvers, optimal speed, and external conditions. First, we present an energy-aware spiral coverage path planning algorithm called E-Spiral. The flight pattern performs missions over regular-shaped areas consisting of concave and convex polygons. The proposed approach considers specific requirement applications, such as overlapping and image resolution, to guarantee a complete area mapping in photogrammetric sensing applications. Furthermore, the algorithm explores an improved energy model to adopt optimal speeds in straight segments of the path to minimize the total energy consumption. Next, we present an energy-aware grid-based coverage path planning algorithm called EG-CPP. The complete algorithm generates coverage paths for mapping missions over irregular-shaped areas containing no-fly zones. Our solution improves an existing grid-based method by replacing its original cost function based on the sum of angles to an energy-aware cost function. As a further contribution, two pruning techniques improve the performance of the algorithm. This improvement speeds up the computational time of the algorithm up to 99%, allowing to explore all the different starting positions in the workspace and saving even more energy. Then, we present an energy-aware pheromone-based solution for patrolling missions called NC-Drone. The approach extends an existing real-time search method aiming at minimizing the number of turning maneuvers while keeping the unpredictable behavior during coverage. We propose two types of NC-Drone, a centralized algorithm and a decentralized one with a few variations. These variations explore a matrix-representation to store the visited cells in the vehicle's memory and adopt synchronization schemes to share the information between the UAVs. We also propose cooperative strategies to improve the algorithm and further explore the problem considering relevant aspects, such as time, uncertainty, and communication. We combine all NC-Drone strategies into a single solution for the patrolling problem. Therefore, three novel approaches are proposed able to successfully addressing several problems related to different coverage path planning scenarios, advancing the state-of-the-art in this area.

Keywords: Coverage Path Planning. Energy-aware. UAV. Flight Pattern. Complete Algorithm. Pheromone-based Heuristic.

RESUMO

CABREIRA, Tauã Milech. **Energy-aware Coverage Path Planning for Unmanned Aerial Vehicles**. Orientador: Paulo Roberto Ferreira Junior. 2019. 146 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

O problema de Planejamento de Caminhos de Cobertura (PCC) é um subtópico de planejamento de movimento na robótica, onde é necessário construir um caminho para um robô para explorar cada local em um determinado cenário. Veículos Aéreos Não Tripulados (VANT) têm sido empregados em diversas aplicações relacionadas ao PCC. Apesar do progresso tecnológico relacionado à sistemas de controle e monitoramento de energia, uma das maiores limitações dos VANTs é o tempo de voo, especialmente em multirotores. Estes veículos possuem uma capacidade de carga limitada, o que consequentemente restringe as suas baterias. Desta forma, minimizar o consumo de energia destes veículos é fundamental para prolongar e garantir a missão de cobertura. O consumo de energia é usualmente associado ao número de manobras de virada em uma determinada trajetória. A maioria dos estudos na literatura visa minimizar estas manobras para economizar energia e, consequentemente, aumentar o tempo de voo. No entanto, fatores importantes são normalmente ignorados, como as fases de aceleração e desaceleração, a velocidade ótima para percorrer uma distância reta, e a velocidade de entrada e o ângulo de virada ao executar uma manobra. Sendo assim, este trabalho propõe soluções de planejamento de caminhos de cobertura com preocupação energética baseadas em padrões de voo, algoritmos completos e métodos inspirados em feromônios. As abordagens lidam com áreas de formato regular e irregular contendo informação parcial e completa e consideram o impacto de diferentes aspectos no consumo de energia dos VANTs, como distância percorrida, tempo de execução da missão, manobras de virada, velocidade ótima e condições externas. Primeiramente, apresentamos um algoritmo de planejamento de caminhos de cobertura em espiral com preocupação energética chamado E-Spiral. O padrão de voo executa a missão em áreas regulares que consistem em polígonos côncavos e convexos. A abordagem proposta leva em consideração requisitos de aplicação específicos, como a sobreposição e a resolução de imagens, para garantir um mapeamento completo da área em aplicações de fotogrametria. Além disso, o algoritmo explora um modelo de energia aprimorado para adotar velocidades ótimas em segmentos retos da rota para minimizar o consumo total de energia. Na sequência, apresentamos um algoritmo de planejamento de caminhos de cobertura baseado em grade com preocupação energética chamado EG-CPP. O algoritmo completo gera rotas de cobertura para missões de mapeamento em áreas irregulares contendo zonas de exclusão aérea. Nossa solução aprimora um método baseado em grade existente através da substituição de sua função de custo original baseada na soma dos ângulos por uma função de custo com preocupação energética. Como contribuição adicional, duas técnicas de poda aprimoram o desempenho do algoritmo. Esta melhora acelera o tempo computacional do algoritmo em até 99%, permitindo a exploração de todas as diferentes posições iniciais no cenário e economizando ainda mais energia. Por fim, apresentamos uma solução baseada em feromônios com preocupação energética para missões de patrulhamento chamada NC-Drone. A abordagem aperfeiçoa um método de busca em tempo real existente com o intuito de minimizar o número de manobras de virada enquanto mantém o comportamento imprevisível durante a cobertura. Propomos dois tipos de NC-Drone, um algoritmo centralizado e um decentralizado com algumas variações. Estas variações exploram uma representação em matriz para armazenar as células visitadas na memória do veículo. Elas também adotam esquemas de sincronização para compartilhar informações entre os VANTs. Também propomos estratégias cooperativas para aprimorar o algoritmo e explorar ainda mais o problema considerando aspectos relevantes, como tempo, incerteza e comunicação. Combinamos todas as estratégias do NC-Drone em uma única solução para o problema de patrulhamento. Portanto, as três novas soluções propostas são capazes de abordar com sucesso diversos aspectos relacionados a diferentes circunstâncias do problema de planejamento de caminhos de cobertura, avançando o estado-da-arte nesta área.

Palavras-chave: Planejamento de Caminhos de Cobertura. Preocupação Energética. VANT. Padrão de Voo. Algoritmo Completo. Heurística baseada em Feromônio.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

A*	A Star
ACO	Ant Colony Optimization
BCD	Boustrophedon Cellular Decomposition
BF	Back-and-Forth Pattern
CDT	Constrained Delaunay Triangulation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CPP	Coverage Path Planning
GCS	Ground Control Station
E-BF	Energy-aware Back-and-Forth CPP
EG-CPP	Energy-aware Grid-based CPP
E-MoTA	Exterior Modification of Trajectory Algorithm
E-Spiral	Energy-aware Spiral CPP
ETHNOS	Expert Tribe in a Hybrid Network Operating System
FW	Fixed-wing
GA	Genetic Algorithm
GSD	Ground Sampling Distance
HM	Harmony Matrix
HS	Harmony Search
I-MoTA	Interior Modification of Trajectory Algorithm
IP	Information Points
IPP	Informative Path Planning
LiPo	Lithium Polymer
LRTA*	Learning Real-Time A*
MOPP	Multi-Objective Path Planning
NC	Node Counting
NC-Drone	Extended Node Counting

NFZ	No-fly Zone
NTM	Number of Turning Maneuvers
POI	Points of Interest
PCC	Planejamento de Caminhos de Cobertura
QMI	Quadratic Mean of Intervals
RC	Radio Control
ROS	Robot Operating System
RTL	Return-to-Launch
RTSM	Real-Time Search Methods
RW	Rotary-wing
SAR	Search and Rescue
SDF	Standard Deviation of Frequencies
SP	Spiral Pattern
TSP	Travelling Salesman Problem
TVUR	Thrun's Value-Update Rule
UAV	Unmanned Aerial Vehicle
VANT	Veículo Aéreo Não Tripulado
VRP	Vehicle Routing Problem
WVUR	Wagner's Value-Update Rule

CONTENTS

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) consist of aerial platforms with no pilots onboard the vehicle. These platforms are remotely and manually operated by a human, but they also perform automated pre-programmed flights. Autonomous flights can be executed using intelligent systems integrated with onboard sensors. These vehicles have increasingly been employed in several application domains, such as surveillance (BASILICO; CARPIN, 2015), smart farming (LOTTES et al., 2017), photogrammetry (CESETTI et al., 2011), disaster management and civil security (MAZA et al., 2011), wildfire tracking (PHAM et al., 2017), cloud monitoring (RENZAGLIA; REYMANN; LACROIX, 2016), structure supervision (GUERRERO; BESTAOUI, 2013), and power line inspection (CHANG et al., 2017).

The Coverage Path Planning (CPP) problem is a motion planning subtopic in robotics, where it is necessary to build a path for a robot to explore every location in a given scenario (CHOSET, 2001). Aerial platforms can also deal with this problem, but several additional aspects must be considered, such as endurance, maneuverability limitations, restricted payload, environmental conditions, among others. Specific applications demand certain types of cameras (simple, multispectral, thermal or 3D), sensors (temperature, humidity, pressure, proximity), and accessories. Onboard cameras and sensors can increase the vehicle's weight and reduce endurance, which is quite limited in certain types of UAVs, such as multirotors. In such vehicles, endurance is about 25~30 min, even in more sophisticated models released in 2019 (ADAMS, 2019). Moreover, turning maneuvers (DI FRANCO; BUTTAZZO, 2016) and wind fields (WARE; ROY, 2016) increase the energy consumption in outdoor missions.

Most UAVs nowadays engage in missions based on CPP using simple geometric flight patterns (ANDERSEN, 2014). The one that is employed the most in real-world scenarios is the back-and-forth (BF), also known as a zigzag move or lawnmower pattern. Following the BF, the UAV executes long straight movements and 180° turning maneuvers when it reaches the border of the area. The most popular flight-control software, the Mission Planner, implements this approach to allow flights based on an offline programmed plan (OBORNE, 2017). Following the same idea, one can design a spiral flight pattern (SP) where the UAV flies in circles, slowly decreasing the circle radius while flying towards the center. Both flight patterns deal with the problem requiring very low computation and no communication (ÖST, 2012).

Flight patterns based on simple geometric shapes, such as back-and-forth and spiral, can be efficiently employed over regular-shaped areas (squares, rectangles, and convex polygons) for specific applications such as photogrammetric sensing and crop field monitoring. However, these patterns are predictable and can be easily recognized by a fugitive target in patrolling missions or military surveillance. Such coverage approaches follow a pre-defined pattern which repeats the same behavior during the coverage. In this way, the object can avoid being tracked by heading to a different direction and staying out of sight of the UAV. Furthermore, these flight patterns may be inefficient, depending on the area's shape complexity.

The CPP problem with UAVs can also be explored using approaches classified either as a heuristic or complete, according to CHOSET (2001). In the heuristic approaches, the vehicles follow a set of simple rules which defines their behavior. Such methods are usually employed by vehicles which do not have full knowledge about the area. Since UAVs have only partial information of the map, they must use their onboard sensors to gather data while covering the scenario. These methods do not have coverage success guarantee due to the complexity of the task.

On the other hand, complete methods can provide these guarantees by using cellular decomposition. This technique consists of environment discretization into cells to simplify coverage in each subregion. Full coverage can be achieved by ensuring that the vehicle visits each decomposed cell. Thus, we are going to introduce pheromone-based approaches and complete methods. The former one consists of simple heuristics based on the natural behavior of ants to perform local searches, while the latter one corresponds to complex search algorithms.

Pheromone-based approaches interleave planning and plan execution, allowing fine-grained control over how much planning to perform between plan executions. Such methods explore pheromones to guide the vehicle's navigation. Pheromones can be dropped in and taken off from a local, evaporate through time and propagate to the neighborhood. Different flavors of pheromone can represent different kinds of information, while some types of pheromone can attract or repel vehicles. These approaches have been proposed to solve CPP problems with land vehicles since they have low computational cost and computational power represents a rigid constraint to any autonomous vehicle (KOENIG; LIU, 2001). However, most of the previous studies are limited only to simulations (ZELENKA; KASANICKÝ, 2014; KOENIG; LIU, 2001; FAMOURI; HAMZEH, 2014), and just a few methods have addressed the problem with aerial vehicles in real-world scenarios (NATTERO et al., 2014; ZELENKA; KASANICKY, 2014a).

Complete methods consist of complex algorithms performing exhaustive searches over a grid represented as a graph (BARRIENTOS et al., 2011; VALENTE et al., 2013). These methods deal with irregular-shaped areas splitting it into subareas and discretizing it via cell decomposition, which reduce the area complexity. Despite providing a feasible solution, these algorithms usually present high computational time during the search phase. Furthermore, they generally adopt simple cost functions based on the number of turning maneuvers, traveled distance, or mission execution time to evaluate the candidate solutions.

Despite the technology progress related to control systems and energy monitoring, one of the main limitations of UAVs is the endurance, due to the limited payload of the vehicles (HUANG et al., 2009). The energy consumed during a mission depends on several parameters, such as flying time, optimal speed, turns, and altitude. The number of turning maneuvers performed by these vehicles significantly impacts on energy consumption, so find a path with the minimum turns enhances its endurance (DI FRANCO; BUTTAZZO, 2015, 2016). It is also possible to save energy using optimal speeds depending on the length of the path segment. Besides, when flying at higher altitudes, it is possible to cover the same area with a smaller path (BANDEIRA et al., 2015). Nonetheless, flights at higher altitudes expose the vehicles to stronger winds (JOHNSON, 1985), consequently also reducing its endurance (WARE; ROY, 2016). In this way, coverage path planning must consider the energy limitations of the vehicles and environment external conditions.

Thus, considering that: (i) one of the main drawbacks of performing CPP missions with certain types of UAVs is the limited endurance; (ii) there are several scenarios and applications where different algorithms and flight patterns may be explored to carry out a mission; and (iii) there is no general agreement about which method is most suitable for each type of scenario or circumstance taking into account real energy limitations, **this work aims to propose energy-aware coverage path planning solutions for regular and irregular-shaped areas containing full and partial information considering the impact of different aspects in the UAV energy consumption.**

As specific goals, this work also aims to:

- Show that minimizing the turning maneuvers to indirectly minimize the energy is not the most appropriate method for generating energy-aware coverage solutions.
- Determine the relevant aspects that influence in the UAV's energy consumption.
- Define the most suitable coverage solutions for regular and irregular-shaped areas considering an accurate energy model.
- State that there is a tradeoff between the spatial and temporal uniform distribution of visits and the energy consumption in patrolling missions.

The main contributions and publications of this work can be summarized as follows:

- **Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications**. Published in *IEEE Robotics & Automation Letters* and presented at the *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018) (Qualis A1)*. Link: `https://ieeexplore.ieee.org/document/8411478`. Authors: Tauã Milech Cabreira, Carmelo Di Franco, Paulo R. Ferreira Jr., and Giorgio C. Buttazzo.

  An energy-aware spiral coverage path planning algorithm (E-Spiral) for photogrammetric sensing. The flight pattern algorithm covers regular-shaped areas (rectangles, convex and concave polygons) and applies the overlapping rates necessary to build a mosaic - commonly used in this type of application. The algorithm also uses different optimal speeds for each straight segment of the path, according to the energy model proposed by DI FRANCO; BUTTAZZO (2016).

- **Grid-Based Coverage Path Planning with Minimum Energy Over Irregular-Shaped Areas with UAVs**. Published in *2019 International Conference on Unmanned Aircraft Systems (ICUAS'19) (Qualis A1)*. Authors: Tauã Milech Cabreira, Carmelo Di Franco, Paulo R. Ferreira Jr., and Giorgio C. Buttazzo.

  An energy-aware grid-based coverage path planning algorithm (EG-CPP) aimed at minimizing the energy consumption during mapping missions over irregular-shaped areas. The complete algorithm exploits the energy model described by DI FRANCO; BUTTAZZO (2016) and improved by CABREIRA et al. (2018) to reformulate the cost function adopted by VALENTE et al. (2013). Furthermore, the algorithm includes two pruning techniques drastically reducing the computational time (in the order of seconds) and saving even more energy in a real-world coverage path.

- **Strategies for Patrolling Missions with Multiple UAVs**. Submitted to the *Journal of Intelligent & Robotic Systems (JINT)* as extended paper version. Current status: Accepted (Published in early 2020) (Qualis A2). Authors: Tauã Milech Cabreira, Kristofer Kappel, João L. Marins, Lisane Brisolara de Brisolara, and Paulo R. Ferreira Jr.

  Strategies to improve pheromone-based NC-Drone heuristic and further explore the patrolling problem considering relevant aspects, such as time, uncertainty, and communication. Four strategies are presented: Watershed Strategy, Time-based Strategy, Evaporation Strategy, Communication-Frequency Strategy. A series of strategy combinations and a final solution combining all strategies are also presented for the patrolling problem.

The marginal contributions and publications of this work can be summarized as follows:

- **Survey on Coverage Path Planning with Unmanned Aerial Vehicles**. Published in *Drones* in 2019. Link: `https://www.mdpi.com/2504-446X/3/1/4`. Authors: Tauã Milech Cabreira, Lisane B. de Brisolara, and Paulo R. Ferreira Jr.

- **Patrolling Strategy for Multiple UAVs with Recharging Stations in Unknown Environments**. Published in *2019 XVI Latin American Robotics Symposium and VII Brazilian Robotics Symposium (LARS/SBR)*. Authors: Patrick G. de Vargas, Kristofer S. Kappel, João L. Marins, Tauã Milech Cabreira, and Paulo R. Ferreira Jr.

- **Flying robots with wind awareness: estimation of the wind velocity and direction through the data log and a closed-form energy model**. Published in *2019 XVI Latin American Robotics Symposium and VII Brazilian Robotics Symposium (LARS/SBR)*. Authors: João L. Marins, Tauã Milech Cabreira, and Paulo R. Ferreira Jr.

- **An Energy-Aware Real-Time Search Approach for Cooperative Patrolling Missions with Multi-UAVs**. Published in *2018 XV Latin American Robotics Symposium and VI Brazilian Robotics Symposium (LARS/SBR)*. The paper has been qualified in the conference's Top 16 and has been extended to the *Journal of Intelligent & Robotic Systems (JINT)*. Link: `https://ieeexplore.ieee.org/document/8588560`. Authors: Tauã Milech Cabreira, Kristofer S. Kappel, Lisane B. de Brisolara, and Paulo R. Ferreira Jr.

- **Terrain Coverage with UAVs: Real-Time Search and Geometric Approaches Applied to an Abstract Model of Random Events**. Published in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. Link: `https://ieeexplore.ieee.org/document/7783518`. Authors: Tauã Milech Cabreira and Paulo R. Ferreira Jr.

- **Avaliação do Consumo Energético em Redes de Sensores Sem Fio Utilizando um Veículo Aéreo Não Tripulado como Coletor de Dados**. Published in *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*. Link: `http://sbesc.lisha.ufsc.br/sbesc2018/Proceedings`. Authors: Nicolas Borges da Silva, Jean Roberto Antunes, Tauã Milech Cabreira, Lisane B. de Brisolara, Paulo R. Ferreira Jr., and Julio C. B. de Mattos.

- **A Closed-form Energy Model for Multi-Rotors Based on the Dynamic of the Movement**. Published in *2018 VIII Brazilian Symposium on Computing Systems

Engineering (SBESC). Link: `https://ieeexplore.ieee.org/document/8691921`. Authors: João L. Marins, Tauã Milech Cabreira, Kristofer S. Kappel, and Paulo R. Ferreira Jr.

- **Análise do Consumo de Energia por Quadrirotores em Voos para Cobertura de Terrenos**. Published in *2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC)*. Link: `http://sbesc.lisha.ufsc.br/sbesc2016/Proceedings`. Authors: Tauã Milech Cabreira, Rodrigo Acosta, Nicolas Borges da Silva, Pedro Marchi, Paulo R. Ferreira Jr., and Julio C. B. de Mattos.

This work is organized as follows:

Chapter 2 This chapter presents a general overview of the coverage path planning problem with unmanned aerial vehicles, considering different types of areas of interest, decomposition techniques, and performance metrics.

Chapter 3 This chapter explores related work in coverage path planning with unmanned aerial vehicles classifying the surveyed approaches according to decomposition techniques and information availability.

Chapter 4 This chapter describes the proposed energy-aware coverage path planning algorithms exploring flight patterns, complete algorithms, and pheromone-based methods for regular and irregular-shaped areas containing full and partial information.

Chapter 5 This chapter explains the simulation and real flight experiments and presents the obtained results from the proposed approaches comparisons.

Chapter 6 This chapter presents the final considerations of this research and expands ideas for future work in coverage path planning with unmanned aerial vehicles.

2 THEORETICAL FOUNDATION

This chapter presents the theoretical foundation about Coverage Path Planning with UAVs. Section 2.1 defines and discusses the UAVs and their main characteristics, indicating which are in the scope of this thesis; Section 2.2 describes the coverage mission presenting its flight modes and mission phases. This section also explores relevant aspects, such as navigation data and failsafe mechanisms. Section 2.3 presents the Coverage Path Planning problem and explains the scenario representation form, describing the characterization of areas of interest. This section also shows the different decomposition techniques employed to split and discretize the areas of interest as well as the performance metrics used to evaluate the quality of planning and obtained coverage.

2.1 Unmanned Aerial Vehicles

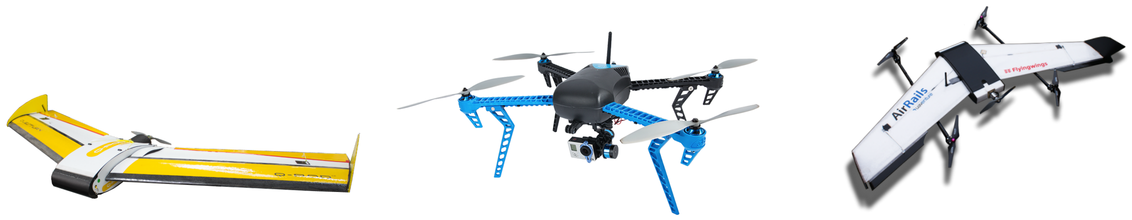
The UAVs are classified into two main top-level configurations, fixed-wing and rotary-wing. Both types present specific advantages and challenges, considering the control and guidance systems (ALVARENGA et al., 2015). Fixed-wing UAVs have rigid wings with an airfoil which allows flying based on the lift created by the forward airspeed. The navigation control is obtained using control surfaces in the wings (aileron, elevator, and rudder). The aerodynamics support longer endurance flights and loitering, also allowing high-speed motions. Besides, these vehicles can carry heavier payloads in comparison with rotary-wing vehicles. However, these platforms need a runway to take-off and land and are not able to perform hovering tasks, since they need to move during missions constantly (KANELLAKIS; NIKOLAKOPOULOS, 2017).

Rotary-wing UAVs present maneuverability advantages using rotary blades. These platforms can perform vertical take-off and landing (VTOL), low-altitude flights, and hovering tasks. The use of rotary blades produces aerodynamic thrust forces and does not require relative velocity (ALVARENGA et al., 2015). This type of aerial platform can also be classified into single-rotor (helicopter) and multirotor (quadcopter and hexacopter).

The single-rotor has two rotors, the main one for navigation and the tail one for controlling the heading. These vehicles are usually able to vertically take-off and land, and they do not require airflow over the blades in order to move forward. Instead, the blades themselves create the needed airflow. A gas motor enables even longer endurance flights in comparison with multirotors. This type of vehicle can carry high payloads, such as sensors and manipulators, while performing hovering tasks and long-time flights in outdoors missions. However, these platforms present mechanical complexity and elevated cost (KANELLAKIS; NIKOLAKOPOULOS, 2017).

The multirotor is classified into subclasses according to the number of rotor blades. The most commons are the quadcopter and the hexacopter, but tricopters and octocopters have also been developed. Multirotors are fast and agile platforms and can perform demanding maneuvers. They are also capable of hovering or moving along a target. Nevertheless, these platforms have limited payload and endurance. Mechanical and electrical complexity is quite low as these parts are abstracted away within the flight and motor controllers (KANELLAKIS; NIKOLAKOPOULOS, 2017).

There is also the hybrid UAV, which is a specific type of aerial platform including the advantages of both, fixed-wing and rotary-wing, thus having the capability of VTOL, high-flight speed, and increased flight time. These vehicles are classified into Convertiplanes and Tail-Sitters. The former one consists of a hybrid platform which performs the basic maneuvers keeping the aircraft reference line in the horizontal direction. The latter one is a platform able to vertically take off and land on its tail, tilting forward in order to achieve horizontal flight (SAEED et al., 2015). Finally, other types of classifications related to UAVs may be found in the literature considering mission requirements, such as altitude and endurance. In these cases, the aerial platforms can be categorized considering low, medium, and high altitude, and also considering short and long endurance (WATTS; AMBROSIA; HINKLEY, 2012). Figure 1 illustrates the main top-level configurations of UAVs, fixed-wing, rotary-wing, and the hybrid VTOL combining both characteristics of the first two types.



(a) (b) (c)

Figure 1 – Main top-level configurations of UAVs: (a) Fixed-wing, (b) Rotary-wing, and (c) Hybrid VTOL.

2.2 Flight Mission

The aerial system usually employed in flight missions is composed by a single or a fleet of UAVs and a Ground Control Station (GCS), which monitors the vehicle's information during mission execution. Despite the technological progress regarding autonomous flight, it is essential to emphasize that two people usually assist phases concerning take-off, mission execution, and landing for each UAV due to safety measures. The pilot supervises the mission and may change the flight mode to manual in case of a failure or an emergency during the flight, while the base operator monitors the navigation data during the mission execution, such as altitude variation and battery discharge, using the GCS (BARRIENTOS et al., 2011). The flight mission is further explored in the next subsections explaining the different types of flight controllers and flight modes, the mission phases and the acquisition of navigation data through the GCS, the failsafe mechanisms and the battery behavior.

2.2.1 Flight Modes

The autopilot consists of a flight controller which provides a stable flying platform, enabling precise manual and automated control over vehicle speed, position, orientation, and actions. This flight controller runs over a board with open/closed hardware, such as Pixhawk, Pixracer, Intel AERO, and Parrot Bebop Autopilot (ARDUPILOT, 2018). The main autopilot is the ArduPilot, widely adopted by most of the UAV models. It offers a wide variety of flight modes from fully manual to fully autonomous, such as Stabilize, Alt Hold, Loiter, Return-to-Launch (RTL), and Auto. These flight modes can be controlled through the radio (via a transmitter switch), via mission commands, or using commands from a GCS. Table 1 explains the main flight modes of the ArduPilot flight controller.

Table 1 – Main flight modes of the ArduPilot flight controller.

Flight Mode	**Description**
Stabilize	Allows the pilot to fly the vehicle manually, but self-levels the roll and pitch axis.
Alt Hold	Maintains a consistent altitude while allowing normal control over the roll, pitch, and yaw.
Loiter	Automatically attempts to maintain the current location, heading, and altitude.
Return-to-Launch	Navigates the vehicle from its current position to hover above the home position. This flight mode is also known as RTL.
Auto	Follows a pre-programmed mission script stored in the autopilot.

Non-professional pilots can use the Loiter flight mode for recreational activities since it enables a manual flight with automatic control over the location, heading, and altitude of the vehicle. The pilot, for instance, may freely fly the aircraft over an area of interest, but when the control sticks are released, the UAV will slow to a stop and hold the position. However, when carrying out coverage for specific research or commercial application, an Auto flight mode containing a pre-programmed script should be used to guarantee mission success. The mission script is made up of navigation commands and "do" commands. The former one affects the location of the vehicle while the latter one is for auxiliary functions and do not affect the vehicle's position (ARDUPILOT, 2018).

2.2.2 Performing a Mission

The home position usually represents the location where the UAV was last armed or where the GPS was locked, depending on the type of vehicle. This position is set up at the beginning of the mission. Then, the mission starts with the navigation command TAKEOFF, which makes the vehicle climb straight up from its current location to the altitude specified (in meters). A set of waypoint commands may be used to move the vehicle to given locations specified by latitude, longitude, and altitude. While the UAV performs the mission flying through waypoint commands, it is possible to execute "do" commands to perform actions, such as change speed or trigger a camera shutter. Finally, after exploring all the waypoints, the vehicle may employ an RTL command to come back to the home location or a LAND command to move to a different location.

A fully assembled UAV typically used during a coverage mission is presented in Figure 2. The system comprises a multicopter and a flight controller with a GPS module and a radio control receiver (RC receiver). The GPS allows localization properly while the RC receiver acts with the RC transmitter to manually control the vehicle and activate their flight modes. The system also contains a rechargeable lithium polymer (LiPo) battery. It is advisable to charge the battery only a few hours before the flight and keep the energy around 40%~60% to store it. Finally, there is a Ground Control Station (GCS), which can run on a tablet, smartphone, or a PC.

2.2.3 Ground Control Station

The mission commands can be set up and uploaded to the UAV using a Ground Control Station (GCS), which consists of a software application running on a ground-based computer that communicates with the UAV via wireless telemetry (ARDUPILOT, 2018). There are several options of GCS for all platforms of desktop and tablet/smartphone, including Mission Planner, QGroundControl, DroidPlanner, and AndroPilot.

The GCS monitors the UAV's flight showing real-time data on the UAV performance and position, such as speed, orientation, altitude, radio signal quality, drained current, and battery status. The pilot can send commands to the UAV and change the flight modes during a mission using a GCS. It is also possible to inspect all the flight data

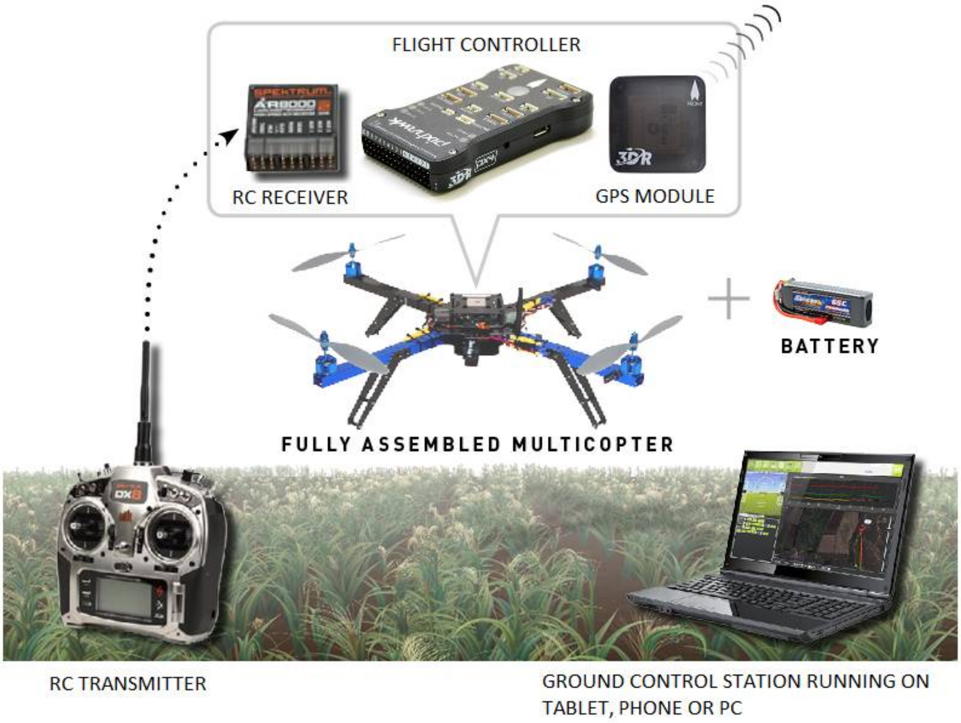

FLIGHT CONTROLLER
RC RECEIVER
GPS MODULE
BATTERY
FULLY ASSEMBLED MULTICOPTER
RC TRANSMITTER
GROUND CONTROL STATION RUNNING ON TABLET, PHONE OR PC

Figure 2 – Fully assembled multicopter along with its components: Ground Control Station, RC transmitter/receiver, GPS module, battery, and flight controller.

after the conclusion of the mission through the log analysis tool.

2.2.4 Flight Data

There are two similar types of flight data recorded during the UAV's mission execution, the dataflash logs, and the telemetry logs. The vehicle records the former one on its onboard dataflash memory. It is possible to download these logs to a PC/tablet/smartphone after the flight. The GCS records the latter one when there is a connection between the autopilot and the system running the GCS via a telemetry link. The flight data contains the following information details: ATT (attitude information, i.e., roll, pitch, yaw angles), CAM (time and position when camera shutter was activated), CMD (commands received from the GCS or executed as part of a mission), CURRENT (battery voltage, current and board voltage information), CTUN (throttle and altitude information), GPS (latitude, longitude, altitude, and speed), and IMU (accelerometer and gyro information) (ARDUPILOT, 2018).

Several problems may occur during the mission execution, such as mechanical failures, high vibrations, compass interference, GPS glitches, and power problems. The log information may be useful to identify the source of these issues. Moreover, it is possible to play back the logs and watch the vehicle's motion on the map, while individual data values are displayed and explored in a graph during flight progress. Finally, 3D images of the flight path can be viewed in Google Earth using KMZ files created from the logs.

2.2.5 Failsafe Mechanism

UAV's flight controller presents failsafe mechanisms to deal with communication and energy issues. These mechanisms are essential to avoid loss of control and crashes and include GCS, RC, and battery failsafe. The GCS and the RC failsafe are triggered when the communication signal between the UAV and the source controlling the vehicle is interrupted by an event. Signal interference may occur when UAV flies long distances away from the source or executes a mission in a place containing several buildings. Figure 3 illustrates the radio failsafe activated during a signal failure in the communication between the radio control (RC) receiver and the pilot's RC transmitter.

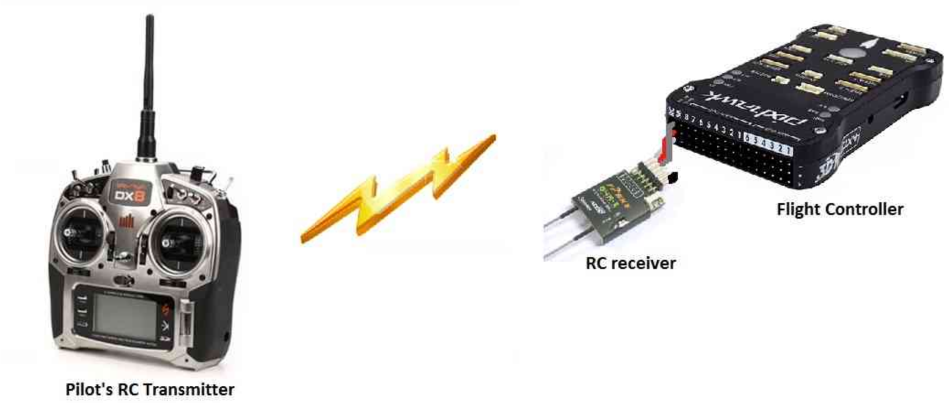


Figure 3 – Radio failsafe triggered when the communication signal between the radio control receiver connected to the flight controller and the pilot's RC transmitter is lost.

Other events causing interference are vehicles flying outside of the RC range, RC transmitter turning off by battery discharge or manually by the pilot, or RC receiver losing power or connection with the flight controller. Once the GCS and the RC failsafe are triggered, the UAV activates RTL mode and returns to the predefined home location. When the signal is re-established, the vehicle remains in its current flight mode (RTL) and the pilot needs to change the flight mode to Stabilize or Loiter to get back the control over the vehicle. It is essential to highlight that when UAV is in AUTO mode, the GCS and RC failsafe are not activated and the aircraft continues the mission.

UAVs also have a failsafe mechanism to prevent energy issues and battery exhaustion. This mechanism is essential not only to avoid crashes but also to keep the battery lifetime. The integrity of Lithium Polymer (LiPo) battery cells may be compromised if the battery is fully drained during flights. Thus, the battery should remain with 20~30% of charge at the end of each mission. It is necessary to manually set-up the battery parameters, such as voltage and estimated current consumed to determine two thresholds to trigger this failsafe. When the UAV crosses the first threshold, it stops the current mission and activates the RTL mode, returning to the home location. During this trajectory, the UAV may also cross the second threshold, depending on the distance from its current position and the home location. If it is not possible to safely reach the launch spot with the remaining energy, the UAV activates the LAND mode, getting down to the ground. However, this mode may lead the UAV to land in inappropriate places, such as trees or lakes. DI FRANCO; BUTTAZZO (2016) propose a novel failsafe mechanism to solve this issue, which continually measures the distance between the aircraft and the home location while estimating the remaining energy in the battery. The UAV triggers the RTL mode when it reaches a point where the energy is only sufficient to return to the home location.

2.3 Coverage Path Planning

Given an area of interest composed by the robot's free space and its boundaries, the CPP problem consists of planning a path which covers the entire target environment considering the vehicle's motion restrictions and sensor's characteristics, while avoiding passing over obstacles. In an aerial context, these obstacles can represent no-flight zones (NFZ) that the UAV should not consider during the planning phase, e.g., areas next to airports or irrelevant buildings.

A decomposition technique usually splits the target environment into non-intersecting regions called cells. A specific strategy should be applied to guarantee complete coverage, since the size and the resolution of the cells may change according to the type of decomposition. For larger cells, several motions are necessary to fully cover only one unit, while in smaller cells a single motion is enough. These cells typically have the same size as a robot (terrestrial coverage) or are proportional to the sensor's range (aerial coverage), representing only one point in the projected path. The CPP problem is further explored in the next subsections considering the area of interest definition, the cellular decomposition techniques, the performance metrics, and the information availability.

2.3.1 Area of Interest

The area of interest is modeled as a polygon described by an ordered set of p vertices $\{v_1, ..., v_p\}$. Each vertex v_i is represented by a pair of coordinates $(v_x(i), v_y(i))$ and its inner angle, denoted by γ_i. For each vertex v_i, the next vertex of the polygon in the considered order is denoted by $v_{next(i)}$, where $next(i) = i(\text{mod } p) + 1$. The edge between a pair of consecutive vertices v_i and $v_{next(i)}$ is denoted by e_i, and its length by $l_i = ||v_i - v_{next(i)}||$, as illustrated in Figure 4. Furthermore, the area of interest can be convex ($\forall i, \gamma_i < \pi$) or concave ($\forall i, \gamma_i > \pi$) and present internal NFZ described by an ordered set of p obstacle-vertices $\{u_1, ..., u_p\}$.

The shape of the area of interest is an important factor to be concerned in the

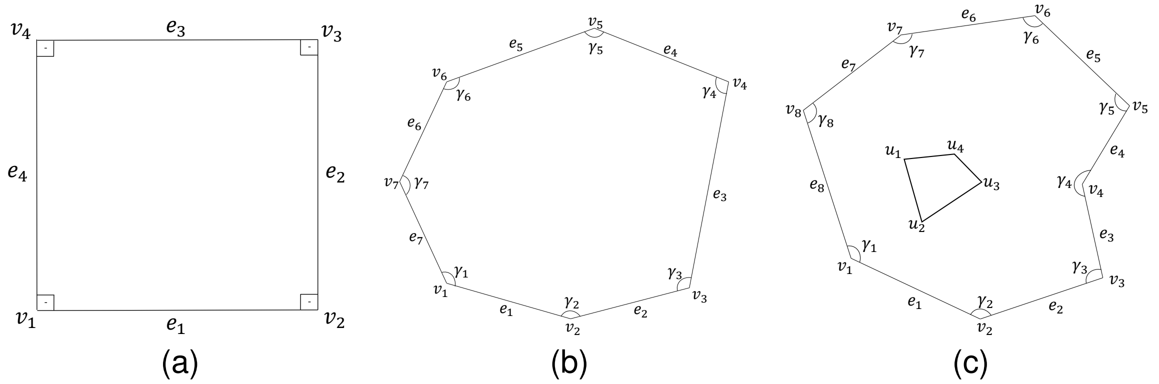


Figure 4 – Different areas of interest explored during CPP missions with vertices, edges and angles: (a) Rectangular, (b) Convex Polygon, (c) Concave Polygon with NFZ.

coverage path planning. Some approaches explore only rectangular areas or simplify their shape to a rectangle, while other ones support more complex shapes, such as concave and convex polygons representing irregular areas. Some methods can deal with areas of interest containing NFZ. These no-fly zones can represent regions where coverage is unnecessary or locations where UAVs are not allowed to fly. Different decomposition techniques are usually adopted to reduce the concavities of complex areas or to split them into smaller cells to facilitate the coverage task.

2.3.2 Cellular Decomposition

Cellular decomposition is a technique employed to divide the target free space into smaller pieces, also known as cells, to simplify the coverage (CHOSET, 2001). This technique is helpful to guarantee complete coverage of the area of interest, one of the major concerns about the CPP problem. Covering all the cells leads to complete coverage of the workspace. There are different types of cellular decomposition in the literature, where the most commonly used are exact and approximate cellular decomposition.

Exact cellular decomposition consists of splitting the area of interest into sub-areas, whose reunion precisely fills the target area. This method comprises drawing a line from one side to another, triggering events every time this line crosses an obstacle boundary. An adjacency graph stores the resulting decomposition, as depicted in Figure 5, where nodes represent the cells and edges represent the relationship between neighbor cells. Simple motions such as back-and-forth usually cover inside the cells. In this case, the CPP problem is reduced to motion planning from one cell to another, i.e., planning a path between adjacent cells sharing a mutual border (CHOSET, 2001). A search can be executed to find a connected-path exploring each node only once. The final coverage path is composed of simple motions performed inside the cells and the inter-cell connections (GALCERAN; CARRERAS, 2013).

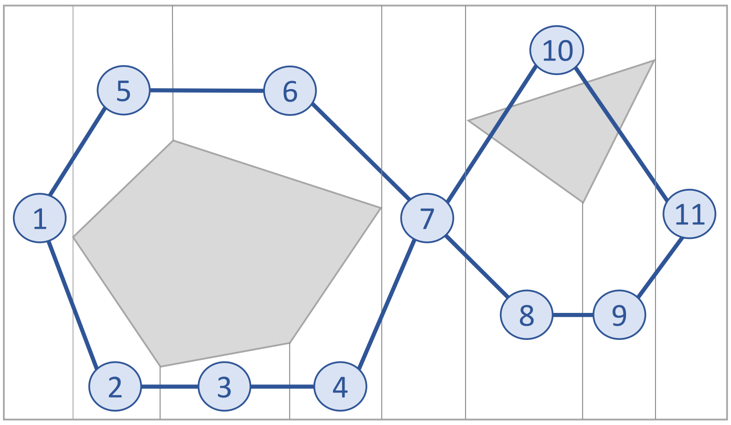


Figure 5 – Adjacency graph describing the workspace, where nodes represent the cells and edges represent the adjacency relationship between neighbor cells.

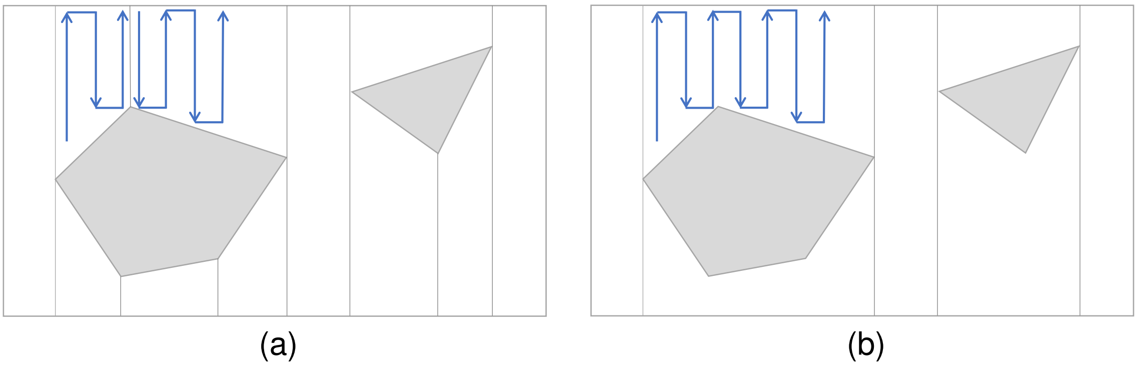


Figure 6 – Two types of exact cellular decomposition: (a) Trapezoidal decomposition, (b) Boustrophedon decomposition.

There are two main exact cellular decomposition techniques worth to be mentioned: trapezoidal decomposition and boustrophedon decomposition (Boustrophedon refers to “the way of the ox”, an analogy to the animal’s motion while dragging a plow in a field), as shown in Figure 6a,b, respectively. The former one divides the area of interest into convex trapezoidal cells and uses an exhaustive walk to determine the cells exploration sequence to fulfill the coverage. The latter one creates non-convex larger cells considering only the obstacle-vertices, also known as critical points. The boustrophedon decomposition can reduce the number of trapezoidal cells and minimize the coverage path length in comparison to the trapezoidal decomposition. Both methods can explore back-and-forth motions to cover the cells.

Approximate cellular decomposition discretizes the area into a set of regular cells (CHOSET, 2001). These cells usually assume a square form, but they can also be represented either in a triangle (BALAMPANIS; MAZA; OLLERO, 2017a) or a hexagonal shape (LIM; BANG, 2010). Grid-based methods can be applied over approximate areas to generate full coverage paths (GALCERAN; CARRERAS, 2013). The size of the cells usually fits the robot’s dimensions in coverage using land robots. However,

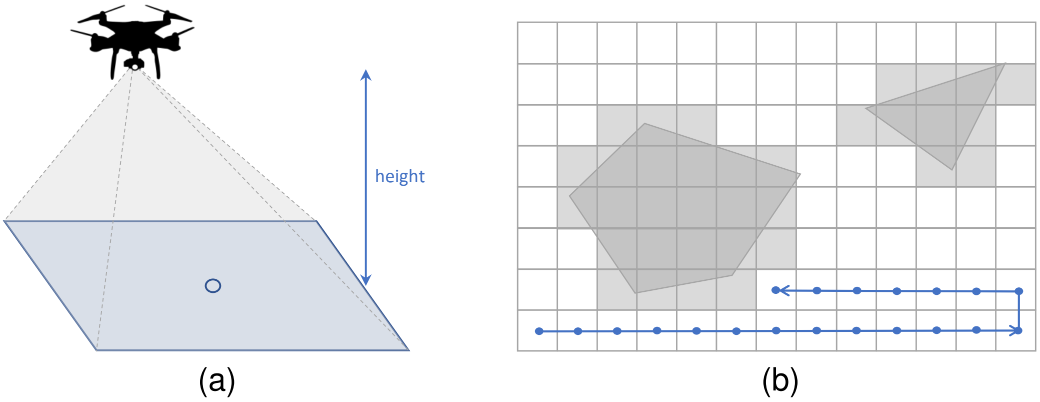


Figure 7 – Approximate cellular decomposition: (a) The UAV's projected area depends on the vehicle's flight altitude and the onboard camera sensor. (b) The area is discretized into a regular grid composed by a set of projected areas (cells), where the center of each one denotes a waypoint.

in aerial coverage, the UAVs fly at a certain altitude from the ground carrying a camera to perform the task. In this case, the size of the cells is proportional to the footprint of the camera, as illustrated in Figure 7a. It is possible to obtain the grid resolution using the image requirements and sensor characteristics, such as resolution and overlapping rates.

The UAV coverage path is composed by a set of k waypoints $\{w_1, ..., w_k\}$. Each waypoint w_i represents a navigation command to the vehicle, such as take-off, change of speed, or move to a specific location. These waypoints contain all the necessary localization information to guide the UAV, such as latitude, longitude, and altitude. Since the cells are proportional to the footprint of the camera, it is possible to simplify the problem, assuming that the center of each cell refers to a waypoint, as shown in Figure 7b. Once the area of interest is discretized into a grid, a complete path that covers the entire workspace can be determined using a coverage algorithm.

2.3.3 Performance Metrics

Coverage algorithms must consider several issues to guarantee the success of a coverage mission, such as the complexity of the area of interest, the presence or not of no-fly zones and the possibility to employ cellular decomposition techniques. Furthermore, coverage algorithms should generate coverage paths according to the application requirements. For instance, the main goal of a photogrammetric sensing application is to create an orthomosaic composed by a set of overlapping aerial photographs. In this case, an application requirement is to ensure a sufficient amount of both forward and lateral photographic overlap. These overlaps allow post-processing software to identify common points between each image. Another requirement for this type of application is the resolution, which can be calculated as ground sampling distance (GSD) (KAKAES et al., 2015). The GSD is the length on the ground corresponding to the side of one pixel in the image or the distance between pixel centers measured on the ground. The lower the flight altitude of the UAV, the smaller the GSD and the better the image quality.

The performance metric used to evaluate the candidate solutions for a coverage path must fulfill the application requirements. Moreover, it should take into account whether the coverage is simple or continuous. In a simple coverage, the area of interest is covered only once, while in a continuous coverage the scenario is swept several times. In both cases, the coverage can be performed by a single or multiple vehicles. The Quadratic Mean of the Intervals (QMI) and Standard Deviation of the Frequencies (SDF) performance metrics were introduced by SAMPAIO; SOUSA; ROCHA (2016) for continuous coverage. QMI corresponds to frequency-regularity equilibrium, an application requirement which combines frequency of visits and evenness of intervals between these visits. This performance metric balances values of average, maximum, and standard deviation of intervals. Equation (1) presents QMI performance metric:

$$QMI = \left[\frac{1}{N_{intervals}} \sum_{x \in cells} \left(\sum_{j=1}^{visits(x)+1} (i_j^x)^2 \right) \right]^{1/2} \tag{1}$$

where i_j^x represents the intervals between the visits j to the cells x and $N_{intervals}$ corresponds to the total number of intervals.

SDF computes the distribution equality of visits over the cells of the scenario, satisfying the application requirement of uniform visitation. The lower the value, the more uniform the frequency is. Zero is the ideal condition, where all cells have the same frequency of visits. QMI and SDF do not depend on each other, having potential applications separately. Equation (2) illustrates the SDF performance metric:

$$SDF = \left[\frac{1}{|cells|} \sum_{x \in cells} (\ freq(x) - F_{avg})^2 \right]^{1/2} \tag{2}$$

where $freq(x)$ corresponds to the frequency of visits on each cell x, F_{avg} is the average frequency of all cells and $cells$ is the total number of cells present in the scenario.

The most common performance metrics for UAV coverage found in the literature are the total traveled distance or the path length (ÖST, 2012; XU; VIRIYASUTHEE; REKLEITIS, 2011), the time-to-complete a mission (ARTEMENKO et al., 2016; VALENTE et al., 2013), the area coverage maximization (LI; WANG; SUN, 2016), and the number of turning maneuvers (MAZA; OLLERO, 2007; TORRES et al., 2016). Minimizing the coverage path length lies in a trade-off with the area coverage maximization. In a workspace split by cellular decomposition, the path length should not be only minimized inside each cell, but also in the intermediate routes between adjacent cells, i.e., the path connecting the end of one cell and the start of the next one. Keeping the UAV inside the area of interest, avoiding flying over locations previously visited and flying at higher altitudes also minimize the total coverage distance.

Another important performance metric is the time-to-complete a mission, where one can consider the area covered per unit path length traveled. Minimizing this value improves the mission execution time for both single and multi-robot coverage (CHOSET, 2001). Using a group of vehicles usually requires a coordination process, which includes splitting the area of interest and assigning the resulting sub-areas among the UAVs. These two steps can be performed separately (MAZA; OLLERO, 2007) or simultaneously (BARRIENTOS et al., 2011). In the former one, the path planner considers the relative capabilities of the vehicles to divide the sub-areas and compute the coverage trajectories after the decomposition. In the latter one, the path planner adopts a negotiation protocol through a distributed way to split the workspace and plan the trajectories at the same time (BARRIENTOS et al., 2011). However, this is a cooperative control problem, usually classified as NP-hard, which requires a reconfiguration process in case of a vehicle failure or a scenario modification. This process divides and re-assigns the areas to the remaining UAVs. Many studies consider vehicles flying at distinct altitudes to avoid collisions and simplify the problem.

The main performance metric adopted in coverage missions with UAVs is the number of turning maneuvers, according to the literature. When an aerial vehicle executes a turning maneuver, it should reduce its speed, rotate and increase its speed again. Thus, the higher the number of executed maneuvers, the greater the time and the energy spent. Equation (3) presents the NTM performance metric:

$$NTM = \sum_{u \in uavs} \left(\sum_{k=1}^{turns(u)} t_k^u \right) \tag{3}$$

where t_k^u denotes the number of turning maneuvers performed by each UAV u.

The authors usually connect performance metrics, such as path length, time-to-complete a mission, and the number of turning maneuvers with energy consumption. In this case, these performance metrics should be minimized to save energy. However, for an efficient energy saving regarding UAVs, further features need to be investigated as vehicle's motions and constraints, turning angles, and optimal speeds. As stated by DI FRANCO; BUTTAZZO (2016), different distances may have different optimal speeds with minimum energy consumption depending on the segment length. Therefore, as the major technological boundary using UAVs, the energetic consumption has attracted the interest of researchers (ARTEMENKO et al., 2016; DI FRANCO; BUTTAZZO, 2015; LI; WANG; SUN, 2016; NATTERO et al., 2014; VALENTE et al., 2013) and has become the main optimization criteria due to the limited endurance of UAVs in coverage path planning missions.

2.3.4 Information Availability

The information availability influences the searching for a solution in coverage missions with UAVs. The workspace information is not fully available or may regularly change in dynamic environments. In such cases, the vehicle must use its onboard sensors to gather information while performing coverage. This online coverage, also known as sensor-based coverage, interleaves between the planning and the execution of the path. The UAV uses the sensor information to reconstruct a full map to execute the mission successfully. The challenge is to keep the data updated while dealing with dynamic behavior, e.g., the localization of a moving target.

On the other hand, UAV path planners are aware of the scenario's layout and have access to all information in static environments. In such cases, the searching is performed offline, i.e., before the execution phase. This type of coverage is usually composed of three sequential main steps: decomposition, planning, and execution. A cellular decomposition technique discretizes and splits the workspace, while a coverage path planning algorithm searches for a solution. This algorithm has full knowledge about the environment and computes the coverage trajectory according to the predefined performance metric. Then, the UAV executes the resulting path and completes the mission. It is important to highlight that during the path execution, there is no external interference. Only exceptional cases, such as pre-programmed failsafe, can change the established path.

3 RELATED WORK

This chapter presents the related work on Coverage Path Planning with UAVs, using the classic taxonomy defined by CHOSET (2001) to classify the existing approaches according to the cellular decomposition technique. Approaches with no decomposition and methods using exact and approximate cellular decomposition are considered. Section 3.1 explores the simple flight patterns adopted in areas of interest with no decomposition technique. Section 3.2 addresses coverage solutions for areas of interest discretized using the exact cellular decomposition. Section 3.3 presents coverage approaches for areas of interest discretized into a grid using approximate cellular decomposition. This section is divided into two subsections, full and partial information. The full information subsection explores algorithms which guarantee the completeness of the mission covering all the decomposed cells, while the partial information subsection presents bio-inspired methods performing coverage under uncertainties.

3.1 No Decomposition

Coverage missions performed over regular-shaped and non-complex areas of interest with a single UAV usually do not require any decomposition. Simple geometric patterns are sufficient to explore such areas, and the most common are back-and-forth (BF) and spiral (SP). The most popular flight-control software, Mission Planner (OBORNE, 2017), adopts the former one to enable the area coverage using a standard pattern. In this pattern, the motions consist of straight lines crossed in both directions with closed-angle maneuvers at the end of each round. The latter one usually performs movements passing by the external vertices of the area and reducing the radius towards the central point.

Some approaches explore only rectangular areas, according to ANDERSEN (2014), where the author compares different types of flight patterns. In this work, the back-and-forth pattern is classified into parallel and creeping line, as illustrated in Figure 8a,b, and it is preferable when the search area is large, and there is no information about the likely target meeting point. The square flight pattern consists of straight lines and

90° turning maneuvers to the right side. The pattern starts at the central location and extends towards the borders, following a pattern similar to an ellipse shape, as shown in Figure 8c. This pattern provides a uniform area coverage.

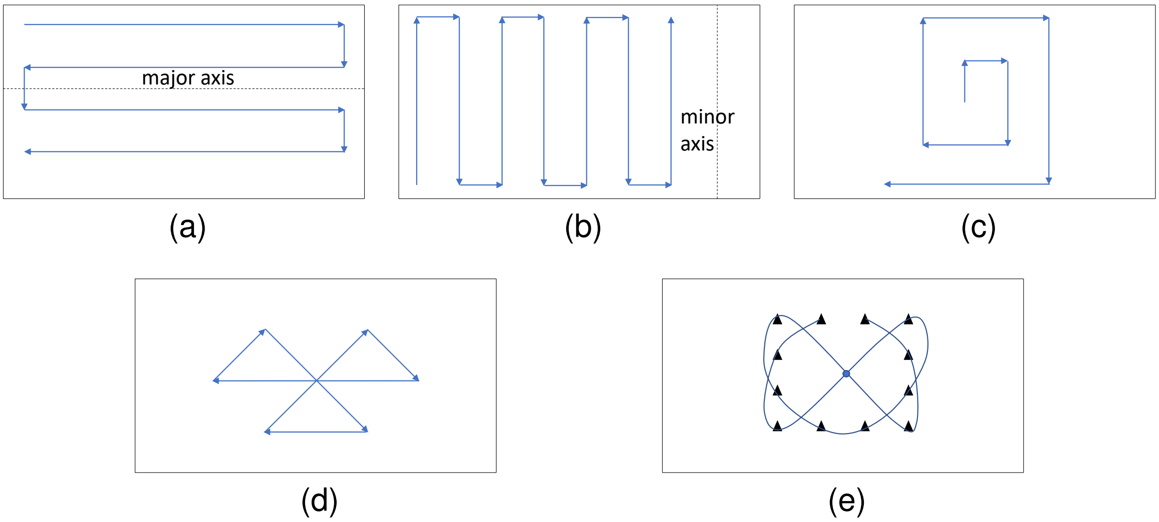


Figure 8 – Simple flight patterns in rectangular areas with no decomposition: (a) Parallel, (b) Creeping Line, (c) Square, (d) Sector Search, (e) Barrier Patrol.

The sector search pattern, presented in Figure 8d, consists of a straight line with 120° turning maneuvers to the right when the vehicle reaches the border of the area. After three sectors, the path returns to the initial point at the center of the scenario. Then, the pattern repeats the same behavior with 30 degrees of displacement. The barrier patrol consists of the definition of 12 points spatially distributed in the search area, as illustrated in Figure 8e. The vehicle initiates its trajectory in the starting point and using a circular movement achieves the next location. From this point, instead of continuing the circular path, it follows to the position closer to the right-corner and reaches the center point.

COOMBES; CHEN; LIU (2017) present an analysis of the effect of wind disturbances in the mission execution time of coverage paths performed by a fixed-wing UAV. Using a circular area of interest covered with BF motions, the authors explore different sweep directions varying from 0 to 360 degrees in increments of 10 degrees with a predefined wind direction with six different speeds. According to the simulated experiments, the coverage direction must be perpendicular to the wind direction to minimize flight time. However, turning maneuvers are directly affected by choice of the course (clockwise or counterclockwise). The authors believe that in more complex scenarios decomposed into cells, the transition distance between these cells has more impact in the flight time than the wind direction.

Recent studies present energy-aware solutions that explore the dynamics and the behavior of the UAVs to save energy. DI FRANCO; BUTTAZZO (2016) present an energy-aware back-and-forth CPP approach (E-BF) for photogrammetry in regular-shaped areas as rectangles and convex polygons. The algorithm determines the best configuration of BF motions at maximum altitude according to resolution constraints while minimizing the number of turns. Moreover, the authors claim that it is possible to reduce the amount of energy flying at an optimal speed. This optimal speed varies according to the traveled distance.

The algorithm finds the first vertex of the longest edge and computes the scan direction parallel to it. Then, it calculates the number of stripes and waypoints, the distance between the strips and consecutive waypoints, and the overlapping rates. Finally, a straight line connects the farthest vertex to the first vertex. An algorithm improvement is also presented to avoid previously explored covered zones, as shown in Figure 9a. Turning the number of stripes in even and increasing the overlapping rate, the returning path can also be used as a scanning path, as illustrated in Figure 9b. The authors also propose offline and online failsafe measures. The former one is checked offline to verify if the battery has enough energy to execute the mission. The latter one is checked online during the flight and continuously analyzes if the remaining energy is capable of bringing back the aerial vehicle to the starting point.

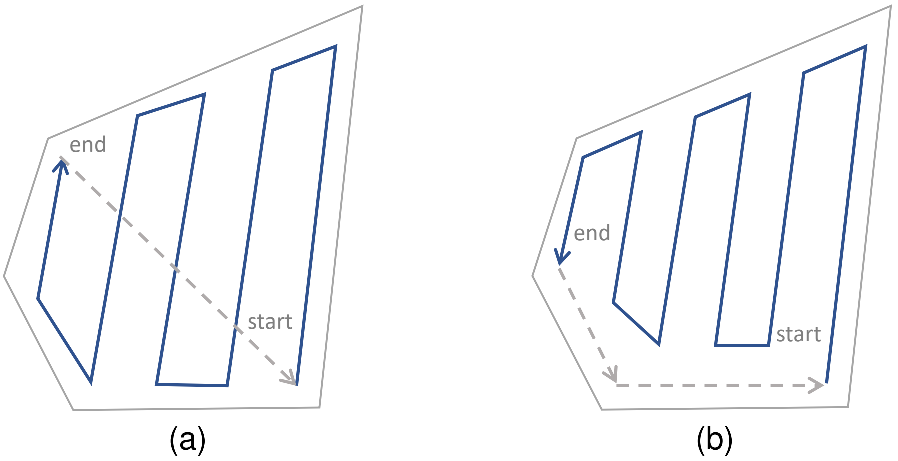


Figure 9 – Energy-aware back-and-forth coverage path planning algorithm: (a) Odd number of stripes, (b) Even number of stripes.

LI; WANG; SUN (2016) present a triple-stage CPP algorithm for UAVs, where it is explored important features not addressed by DI FRANCO; BUTTAZZO (2016), such as payload and power variation. The first step is to build a 3D terrain model using control points to obtain an analytical model. Then, stable power consumption is calculated, considering take-off weight, flight speed, and air friction. The authors believe that the vehicle moves with constant velocity in a steady state, deriving the optimal speed aimed at minimizing energy. An energy consumption map is built to show the amount of energy spent on every part of the path. Finally, a Genetic Algorithm performs an optimization to discover minimum-cost trajectories comprising all vertices.

A UAV spends a considerable amount of time making turning maneuvers since it has to decelerate, rotate, and accelerate to perform such movement. In this way, ARTEMENKO et al. (2016) propose energy-aware algorithms for smoothing trajectories with minimum deceleration using the concept of Bézier curves. The algorithms modify paths such as SCAN (back-and-forth), HILBERT, and LMAT, providing more effective turning maneuvers. The authors compare the energy-aware trajectories with the original ones presented in Figure 10 and conclude that the novel paths reduce the amount of time and energy while keeping the level of the localization accuracy (LoLA).

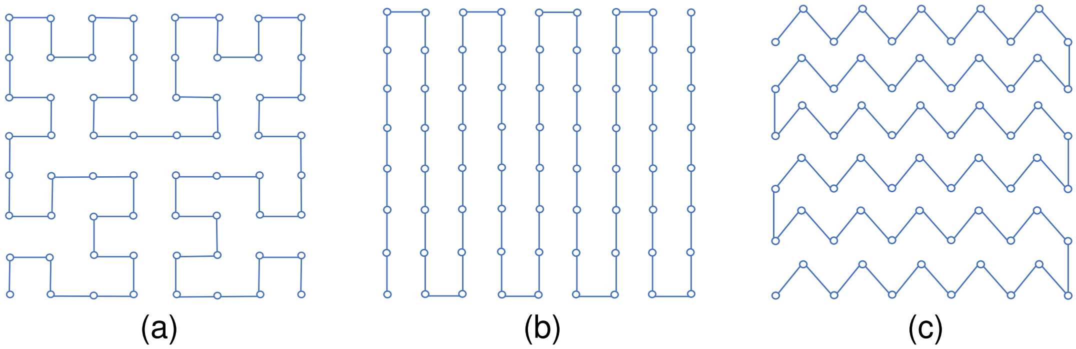
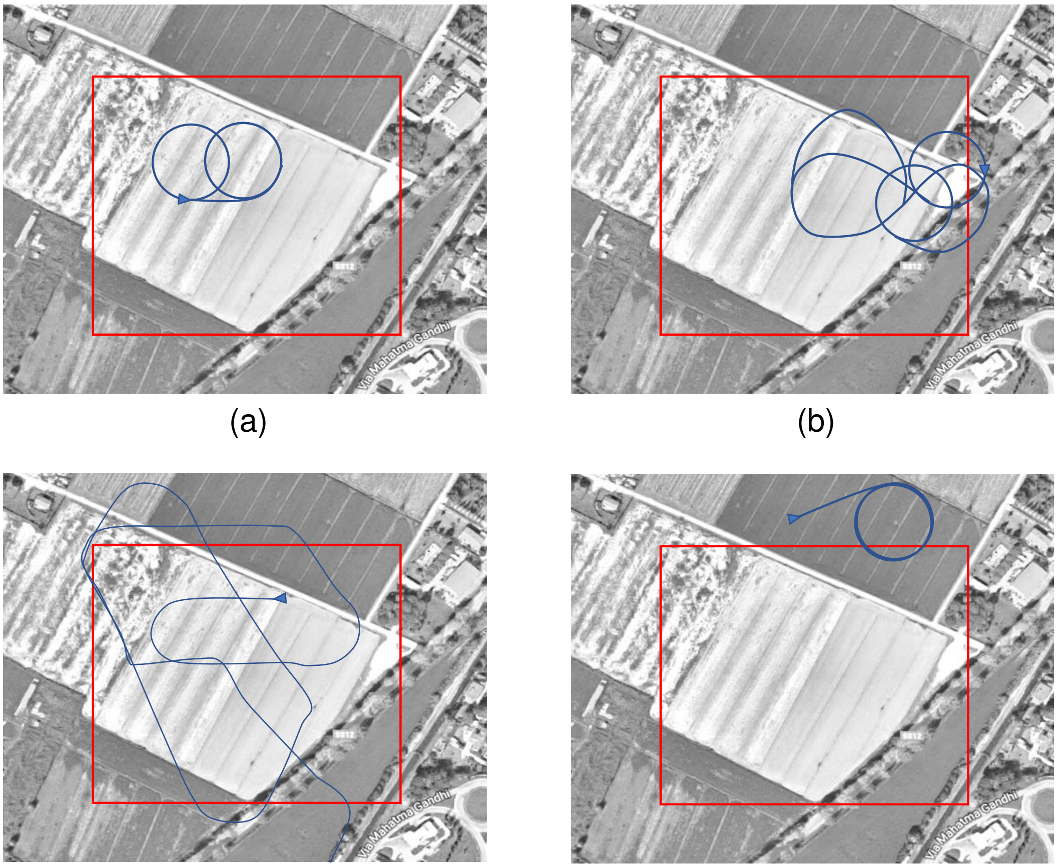


Figure 10 – Traditional CPP paths: (a) Hilbert curves, (b) SCAN, (c) LMAT.

FORSMO et al. (2013) employs Mixed Integer Linear Programming (MILP) for coverage missions in rectangular areas with UAVs. The algorithm considers the UAV onboard camera for distributing waypoints over a particular area to obtain full coverage. However, the solution deals only with rectangular obstacles and places the aerial vehicles in distinct zones of the scenario, not adequately handling the collision avoidance issue among the UAVs. The authors simplify the problem and do not adopt any decomposition. Simulation experiments evaluate the proposed solution considering different cases with constraints, such as waypoint visitation order and camera range reduction.

Coverage missions may require a team of aerial vehicles working cooperatively to improve the task performance given the complexity and size of different scenarios. A cooperative coverage algorithm with critical time for rectangular areas using multiple fixed-wing heterogeneous UAVs is presented by AHMADZADEH et al. (2008). The vehicles fly at distinct and fixed altitudes, such as 80m, 90m, 100m, and 110m, and with constant velocity, as illustrated in Figure 11. Furthermore, the UAVs present maneuverability restrictions and fixed cameras either in the front or in the left wing. An approach based on Integer Linear Programming is used to generate a solution considering these restrictions.

The paths of the UAVs with the frontal camera are circular, while the trajectories of the UAVs with the left side camera are composed of straight lines and left turns. When this vehicle turns right, the camera focuses on the horizon and does not capture any image of the coverage area or image resolution drastically decreases. The authors



(a) (b)

(c) (d)

Figure 11 – Cooperative coverage with multiple fixed-wing UAVs flying at distinct altitudes: (a) 80m and frontal camera, (b) 90m and frontal camera, (c) 100m and left side camera, (d) 110m and frontal camera.

compare the proposed approach using the four fixed-wings vehicles against simple methods such as back-and-forth. Due to motion constraints and field of view, simple patterns such as back-and-forth presented coverage of about 80% of the area of interest, while the proposed approach obtained 100% of coverage. The proposal was tested and evaluated in simulations performed in MATLAB and real flights.

3.2 Exact Cellular Decomposition

Exact cellular decomposition is necessary for large and complex areas of interest. This technique consists of splitting irregular-shaped areas into sub-areas to reduce the concavities and simplify coverage. Single or multiple UAVs can cover these sub-areas. In the former case, the CPP approach must concern the coverage path in each one of the sub-areas and the intermediate paths connecting them. In the latter, the CPP approach must worry about the relative capabilities of vehicles to compute the size of each sub-area. Furthermore, a safety margin should be considered to prevent collision among the UAVs. First, we revise some CPP approaches for single UAVs using back-and-forth and spiral patterns for convex and concave areas. Next, we explore cooperative strategies dealing with multiple UAVs.

3.2.1 Single Strategies

The works of JIAO et al. (2010); LI et al. (2011) explore an exact cellular decomposition for concave polygonal areas. The technique adopts an approach called minimum width sum to decompose the workspace into non-concave sub-areas, exploring a greedy recursive method, which was previously proposed by LEVCOPOULOS; KRZNARIC (1996). The algorithm computes the sweeping direction by finding the minimum distance between an edge and a vertex of the polygonal area and employs back-and-forth motions perpendiculars to this direction to cover the area while minimizing the turning maneuvers (HUANG, 2001). Two sub-regions obtained from the convex decomposition (see Figure 12a) with the same sweeping direction and entirely adjacent are combined into the sub-region P4 to avoid unnecessary back-and-forth movements, as revealed by Figure 12b. It is also possible to change the motion direction from one sub-region to another to obtain a coverage improvement, as shown in Figure 12c. Finally, the optimal sequence of sub-regions is defined to join the final trajectory for a single UAV, as illustrated in Figure 12d.

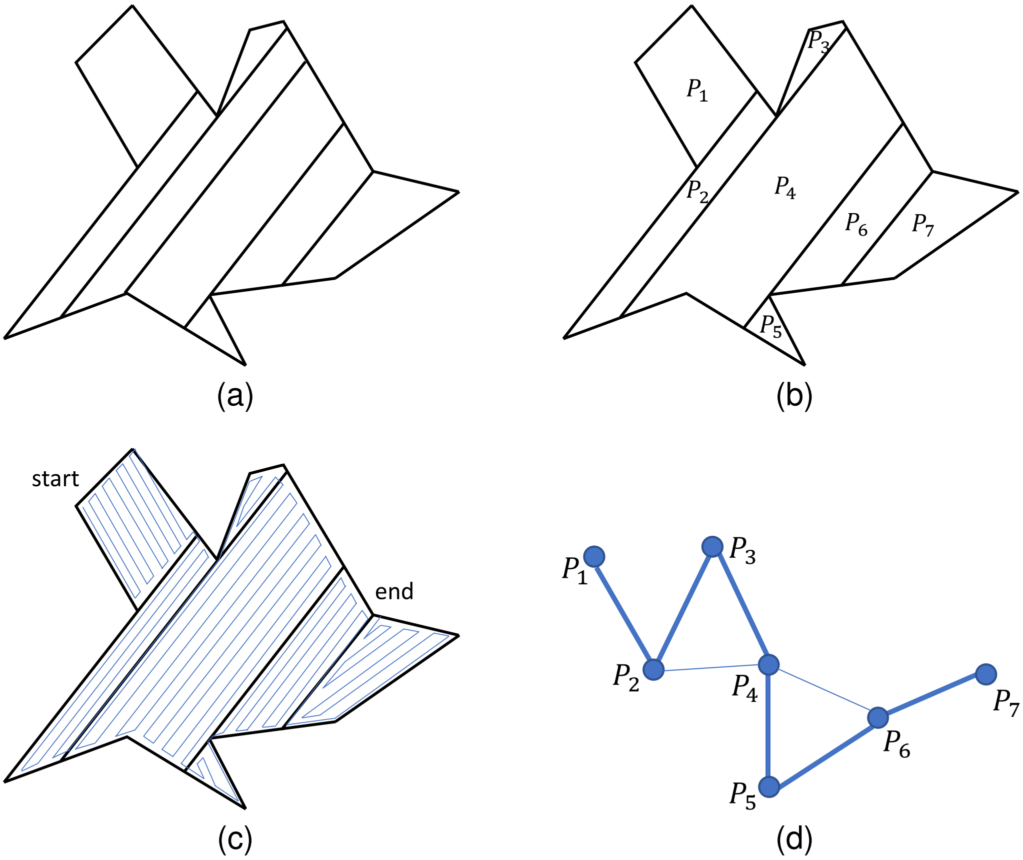


Figure 12 – Exact cellular decomposition for polygonal areas: (a) Convex decomposition, (b) Sub-region combination, (c) Coverage path, (d) Undirected graph.

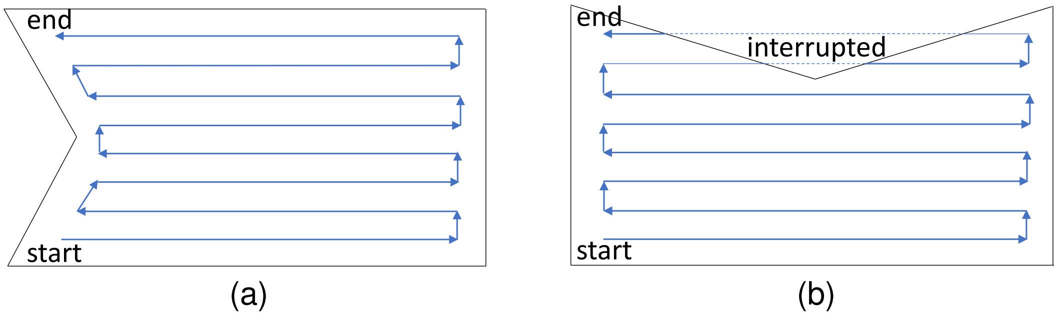


Figure 13 – Coverage using back-and-forth motions in concave polygons: (a) Non-interrupted path, (b) Interrupted path.

TORRES et al. (2016) present a coverage approach to capture pictures of convex and concave areas for 3D reconstruction using aerial vehicles. The solution explores the exact cellular decomposition to perform BF motions following the optimal direction. These motions can cover inside convex polygons without exploring unnecessary external regions. However, in more complex areas as concave polygons, it is needed to check if the mission can be performed in the same way with no gaps during the stripes, i.e., none of the strips cross outside the polygon. Figure 13a illustrates this particular case in a concave polygon. When the path is interrupted, as shown in Figure 13b, an exact decomposition of the polygon is used to simplify the area creating sub-regions.

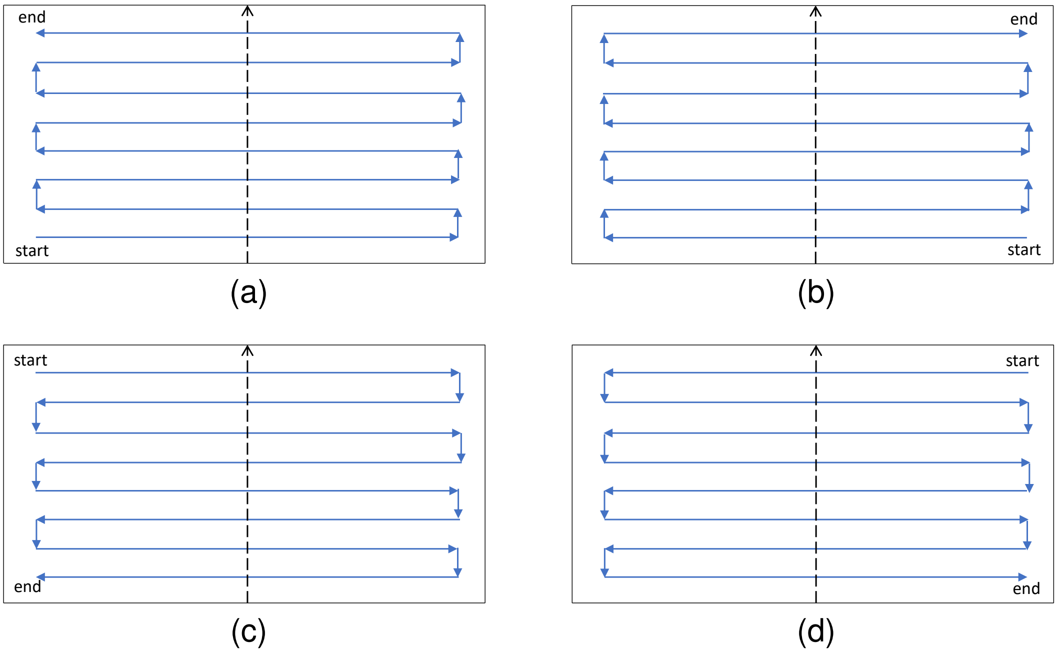


Figure 14 – Four alternatives of back-and-forth movements: (a) line direction/ccw, (b) line direction/cw, (c) opposite way/cw, (d) opposite way/ccw.

The solution explores four different back-and-forth motions after defining the optimal sweeping direction for each sub-region. The alternatives consider two criteria related to the course and the orientation. The former explores if the coverage is going to follow the optimal motion direction or the opposite way. The latter one considers the orientation of the first turning maneuver, clockwise (left) or counterclockwise (right). The alternatives influence the transition distances between the last point of a given sub-region A and the first point of a given sub-region B. It is possible to minimize the transition distances and, consequently, the path length by permuting the sub-regions coverage order with the alternatives of each one. The approach directly connects the final point with the first one using a straight line at the end of the coverage. Figure 14 presents the four back-and-forth alternatives.

Exploring all permutations may consume a high computational time, depending on the number of sub-regions. Thus, the authors use only the adjacent sub-regions for the transitions, reducing the number of permutations. They evaluate the proposed approach in two scenarios. In the first scenario, a concave area is decomposed into five sub-regions pondering only four adjacencies. The authors drastically reduce the number of permutations and the computational time of the algorithm, generating a solution with a slight increment in the total distance. In the second scenario, the authors compare their approach with the one proposed by LI et al. (2011) using the same area, as shown in Figure 15. The novel method decomposes the polygon into four sub-regions, computing only 80 turning maneuvers against 87 of the original work.

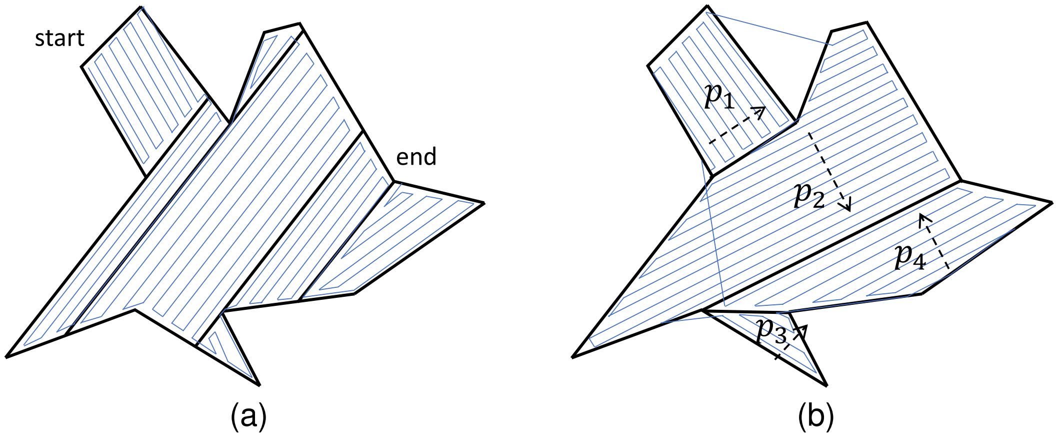


Figure 15 – Comparison between the decomposition approaches in concave areas: (a) Convex decomposition (LI et al., 2011), (b) Concave and convex decomposition (TORRES et al., 2016).

COOMBES et al. (2018) propose a coverage path planning technique for fixed-wing UAVs exploring wind to decrease the flight time. The authors incorporated the wind in the model to compute the coverage paths in a previous work (COOMBES; CHEN; LIU, 2017) and extended their work by proposing a decomposition method to split the complex region into convex polygons. Trapezoidal decomposition divides the area of interest, exploring several rotations of the polygon and a cell recombination is employed using Dynamic Programming to merge cells into convex polygons. This decomposition method also considers optional cells external to the region to find different decompositions with lower flight times.

The UAVs explore the area using back-and-forth motions perpendicular to the wind direction and are allowed to fly outside the area of interest. Both initial and final waypoints of each straight line intersect the contour of the polygon. The UAV performs a motion called trochoidal turn, which consists of 180 degrees turning maneuvers in the presence of wind. This maneuver is the shortest curve connecting the last waypoint of one line to the first of the next one, considering the fixed-wing restricted turning rate. The algorithm incorporates the transition distance between adjacent cells during the path computation. The authors present a cost function called Flight Time in Wind (FTIW) to compute the flight time needed by the aircraft to cover an area. The total time is the sum of the time to fly the straight lines, to perform the trochoidal turns, and to make the transitions between the cells. They compare the proposed method (FTIW) with previous techniques which aim to minimize the number of turns (NT) (LI et al., 2011) and the sum of the altitudes (MSA) (HUANG, 2001). Using a real field and several random polygons generated by a Monte Carlo simulation, the authors claim that their FTIW approach overcomes both previous methods considering the flight time needed to execute coverage.

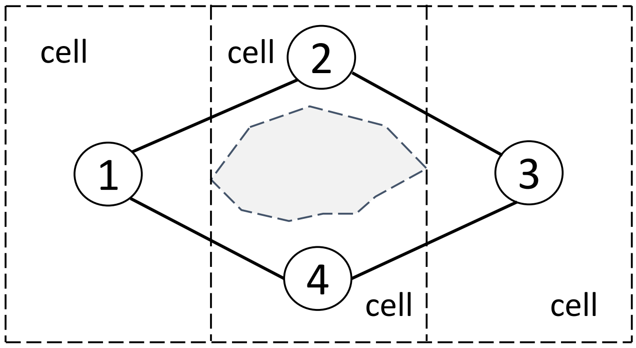


Figure 16 – Area of interest with obstacles decomposed into cells and the adjacency graph representation.

XU; VIRIYASUTHEE; REKLEITIS (2011) and XU; VIRIYASUTHEE; REKLEITIS (2014) present an optimal coverage algorithm for fixed-wing UAVs able to avoid flights in obstacle-regions with arbitrary shapes and previously explored regions. The area of interest is decomposed into a simple set of cells using Boustrophedon Cellular Decomposition (BCD), initially proposed by CHOSET; PIGNON (1998), in an offline phase through a bitmap representation. The BCD is an exact cellular decomposition able to work with non-polygonal obstacles, which presents more efficient coverage paths than the trapezoidal decomposition. An adjacency graph can be built with the vertices representing the cells and the edges connecting the adjacent cells, as illustrated in Figure 16.

Back-and-forth motions explore the cells in an online phase, and the coverage order between the cells follows a Eulerian circuit with start and end at the same vertex. Depending on the next cell location, it is necessary to go through sub-regions previously explored in the scenario. Thus, a more efficient technique is proposed to eliminate the disjunctions adding an extra sweep line. This line guarantees the continuity of the path when it reaches the border of the cells and prevents repeated explorations. The authors compare the original and modified approaches in simulations and real flights, evaluating the performance regarding the total path length and the time to perform the coverage. The method proposed by the authors was 10% more efficient in both criteria.

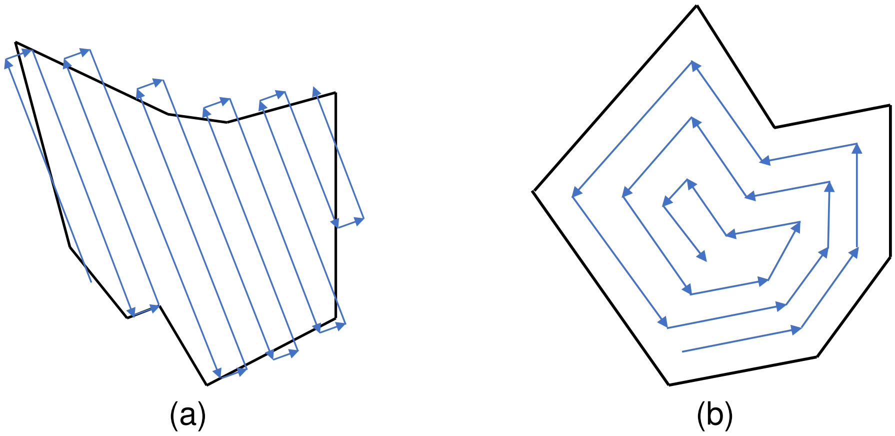


Figure 17 – Simple flight patterns in polygonal areas: (a) Back-and-Forth, (b) Spiral.

ÖST (2012) proposes a decomposition method to split complex shapes into smaller ones and transform concave forms with sharp edges into convex shapes. The author explores flight patterns, such as back-and-forth and spiral motions, as illustrated in Figure 17, combining them with the proposed area decomposition technique. They claim that the back-and-forth with no area decomposition presents trustworthy outcomes in comparison with different combinations. This confidence is because all maneuvers have 90-degree turns, allowing to predict the pattern after four moves. Despite generating slight longer paths, this pattern is capable of handling complex shapes without losing coverage. However, the algorithm consumes considerable time testing all different rotations in the polygon between 0 and 180 degrees to find the optimal motion direction. Spiral pattern generates shortest paths in rounded shapes with large inner angles, according to ÖST (2012). However, this motion does not conclude the coverage in complex areas sometimes. Mixed variations comprising patterns and decomposition can generate a path with minor distance, but these combinations do not always correctly deal with all the instances. The mixtures have proved to be effective only when the area has too many protrusions in different directions, such as a star shape. In some cases, the decomposition may contain more vertices with a large number of small inner angles, generating self-intersections during the junction of the areas.

3.2.2 Cooperative Strategies

Cooperative strategies employ multiple UAVs to cover an area of interest. This type of strategy is usually applied when the workspace is too large to be covered by a single UAV. Depending on the complexity of the problem, the coverage approach may only split the area into sub-areas and plan coverage paths individually for each UAV. More sophisticated approaches deal with motion synchronization, decentralized information and communication issues, and different levels of local priority.

3.2.2.1 Back-and-Forth

MAZA; OLLERO (2007) presents a cooperative strategy in a convex polygonal area using a team of heterogeneous UAVs. A ground control station decomposes the area and assigns the resulting subregions to the vehicles considering their relative capabilities and original locations. Each vehicle computes back-and-forth motions according to the sweeping direction to perform the coverage, minimizing the number of turning maneuvers. The number of parallel stripes and the distance between them depends on the camera footprint of the vehicle. In case of a failure in one of the vehicles, a reconfiguration process splits the area again and re-assigns the subregions to the remaining UAVs. Figure 18 shows the decomposed area of interest with three vehicles and their respective coverage paths with different sweeping directions.

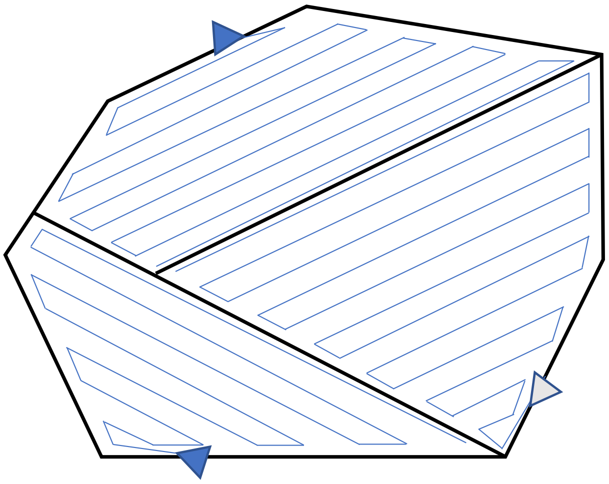


Figure 18 – Cooperative coverage in convex polygon area using a team of heterogeneous UAVs.

3.2.2.2 Spiral

A spiral CPP algorithm for missions in coastal regions using multiple heterogeneous UAVs is explored by Balampanis et al. in several works. The authors discretize the workspace considering the sensing capabilities of the aerial vehicles using a Constrained Delaunay Triangulation (CDT) (KALLMANN; BIERI; THALMANN, 2004) introduced by BALAMPANIS; MAZA; OLLERO (2016) and BALAMPANIS; MAZA; OLLERO (2017a). The authors state that the classical grid decomposition creates regular square cells which are partially over no-fly zones or outside the workspace. In this way, the CDT provides triangle cells within the area of interest, matching almost the exact shape of the area.

In order to generate more symmetrical triangles, they applied the Lloyd optimization (RINEAU, 2016), which is a technique that approximates the cell angle to 60 degrees, enhancing the uniformity. A spiral algorithm generates the coverage paths in the resulting sub-areas. The authors introduced this algorithm in a previous work (BALAMPANIS; MAZA; OLLERO, 2017b) and improved it by introducing a smoothing parameter in another work (BALAMPANIS; MAZA; OLLERO, 2017c). The authors tested the proposed strategy in Software-In-The-Loop simulation. The further analysis compares the spiral-like pattern using the CDT/Lloyd optimization (triangle cells) with the classical grid decomposition (square cells) using back-and-forth motions. The authors claim that their approach is capable of covering a given area fully with a smoother trajectory without entering into no-fly zones or going outside the area. However, the approach presents longer coverage paths with a higher number of turning maneuvers compared to the back-and-forth.

3.2.2.3 Line Formation

A cooperative coverage strategy exploring line formation is proposed by VINCENT; RUBIN (2004) to detect intelligent targets moving in dangerous environments. The authors consider that the targets try to deliberately escape from the search performed by the vehicles inside a rectangular area. Five criteria are guiding the mission, such as maximize the probability of detecting a target, minimize the tracking time and the number of aerial vehicles employed in the mission, provide robustness on the assumption of a failure of one or more UAVs and minimize the amount of information shared among the vehicles.

The vehicles are organized in a line formation and execute long straight movements, as illustrated in Figure 19. As vehicles explore the area in a close formation, the communication among them is simplified, and it favors the continuity of the mission in case of a failure in one of its components. There are two types of control messages exchanged between the vehicles, maintenance messages and update messages. If a vehicle fails to transmit the maintenance message, the adjacent vehicles detect that it is out of action. Then, the group of UAVs exchange update messages to achieve an agreement for the pattern reconfiguration. The coverage success depends on the return to the previously explored neighborhood areas, since the target may move to such areas in an attempting to escape from the sensors of the vehicles.

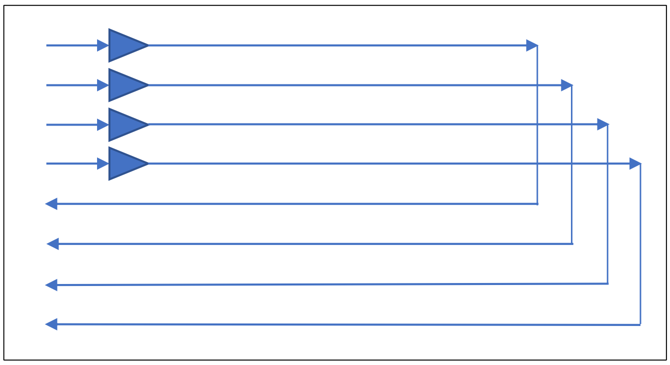


Figure 19 – Cooperative coverage strategy in rectangular areas of interest with intelligent targets.

3.2.2.4 Decentralized Technique

A decentralized algorithm for partitioning rectangular areas during surveillance missions is presented by ACEVEDO et al. (2013a). Homogeneous vehicles distribute the subregions using a one-to-one coordination technique and explore adjacent regions. The UAVs have a short range of communication, as shown in Figure 20, and must share information passing by near points in a synchronized way. The sub-perimeter method generates paths with the same size in subregions equally distributed. This method uses the information about the area to produce a similar inner region, such that the maximum distance from the internal zone to the border of the area is less or equal than the coverage range. The system is adaptable to modifications in the team, such as the departure of a vehicle for repairs or battery recharge. The main goal is to develop a cooperative patrolling strategy to optimize the observation interval among consecutive attempts in any spot and minimize the latency, which is the sharing time of detected information with the other members.

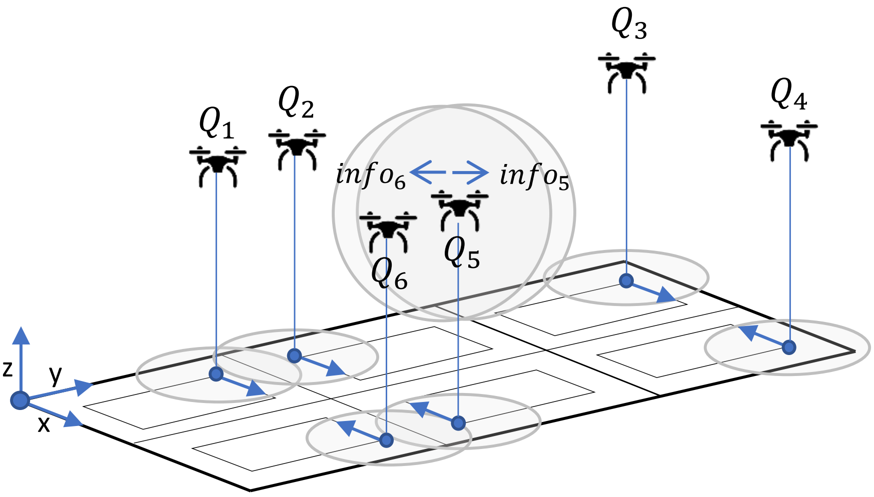


Figure 20 – Decentralized algorithm for surveillance task in rectangular areas with information exchange using a team of homogeneous UAVs.

More recently, ACEVEDO et al. (2013b) extend their approach for surveillance in irregular-shaped areas with heterogeneous vehicles. The proposed method discretizes the area of interest containing obstacles and irregular borders into regular regions, as shown in Figure 21a. The numbered black rectangles in this picture represent forbidden flight zones used to delimit boundaries and indicate obstacles. Instead of partitioning the area of interest and producing several paths, the approach creates a single path segmented and distributed to the vehicles for covering the entire region. Faster vehicles cover more significant segments of the path than ordinary vehicles, and all vehicles invert the patrolling direction at the end of each part. The authors execute simulations in an urban scenario containing constructions and with four vehicles flying in low altitude while avoiding obstacles. The system can adapt itself to the entrance and the exit of vehicles during the mission.

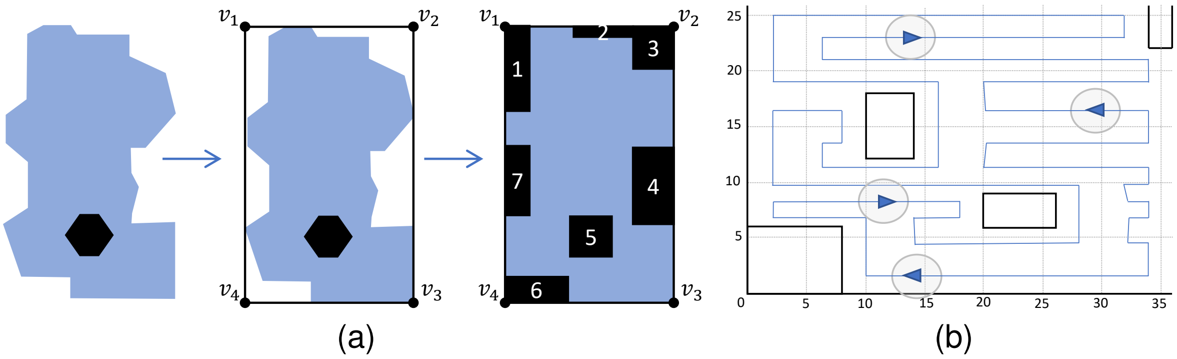


Figure 21 – Surveillance mission in irregular-shaped areas with path segmentation using a team of heterogeneous UAVs: (a) Irregular area approximation, (b) Segmented single path.

Finally, ACEVEDO et al. (2014) explore the one-to-one coordination approach using a grid-shape area partition strategy. This strategy divides the area of interest into non-overlapping subregions monitored by heterogeneous vehicles following distinct paths. The technique allows the vehicles to self-adjust the partitioning according to their maximum capacities, maintaining the synchronization in a distributed and decentralized way. Furthermore, this solution can rearrange initial conditions, such as area shape and vehicles capacity.

3.2.2.5 Local Priority

A continuous coverage and decomposition approach for convex polygonal areas with local priority is proposed by ARAUJO; SUJIT; SOUSA (2013). A sweeping method decomposes the workspace into minor areas and assigns the resulting subregions to the vehicles according to their relative capability. This capability can be measured, for instance, as the amount of space covered per unit of time. The authors developed a method to generate an optimal number of stripes inside each subregion, considering the kinematic constraints of the fixed-wing UAVs. A diameter function describes the polygon altitude and computes the perpendicular optimal sweep direction, minimizing the number of stripes and, consequently, the number of turning maneuvers. The strips have the same width of the onboard camera footprint, and overlapping between the lines is necessary due to positioning errors and slightly variations in the trajectory.

The authors discard the use of some flight patterns, such as back-and-forth and spiral motions, claiming that these patterns can neither deal with different local priorities nor perform multiple visits in specific areas. Thus, the authors propose a back-and-forth/Zamboni flight pattern (Zamboni refers to the machines which repair the ice in hockey arenas), illustrated in Figure 22. The proposed flight pattern allows visiting previously explored stripes since the degree of uncertainty of a locality raises as time passes since the last visit. They also consider that the locations present different degrees of priority managed by a human operator. In this way, after completing coverage in one of the stripes, the vehicle must select the next line considering the uncertainty and the priority.

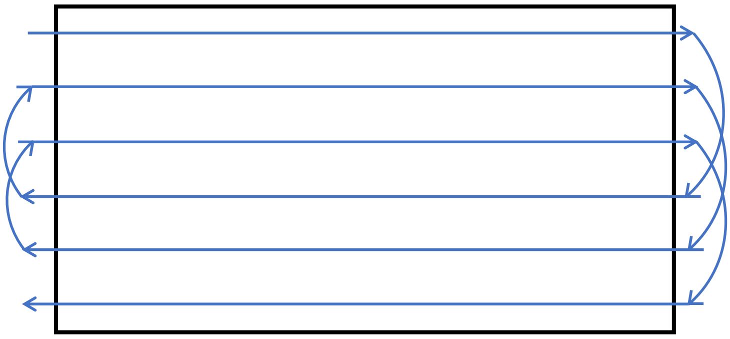


Figure 22 – Back-and-Forth/Zamboni flight pattern with local priority and degrees of uncertainty for continuous coverage missions.

3.3 Approximate Cellular Decomposition

A coverage trajectory is usually planned before its execution, in an offline phase, considering that the aerial platform has full knowledge of the workspace to be covered. However, in some cases the UAV has to interleave between planning and execution, gathering information through its sensors to build an internal map as it moves around the area. In both cases, the approximate cellular decomposition can be employed to discretize the space into a grid, while a grid-based solution can be used to perform the coverage mission. First, it is revised some complete algorithms proposed to deal with irregular-shaped areas, considering approaches based on single (as VALENTE et al. (2013)) and multiple UAVs (as BARRIENTOS et al. (2011)) with full information about the scenario. Next, it is explored some cooperative bio-inspired approaches to deal with environments containing only partial information.

3.3.1 Full Information

Coverage missions are usually performed using the back-and-forth flight pattern, especially in applications like agriculture, but this type of motion generates inefficient trajectories when dealing with complex areas of interest containing an irregular shape format. In this way, VALENTE et al. (2013) propose a coverage path planning algorithm for image mosaicing in precision agriculture over irregular-shaped fields. The approximate cellular decomposition discretizes the area of interest into a regular grid, as shown in Figure 23. Each cell represents a waypoint of the path, and its size depends on the picture dimensions of the onboard camera. The characteristics of the image sensor determine the grid configuration.

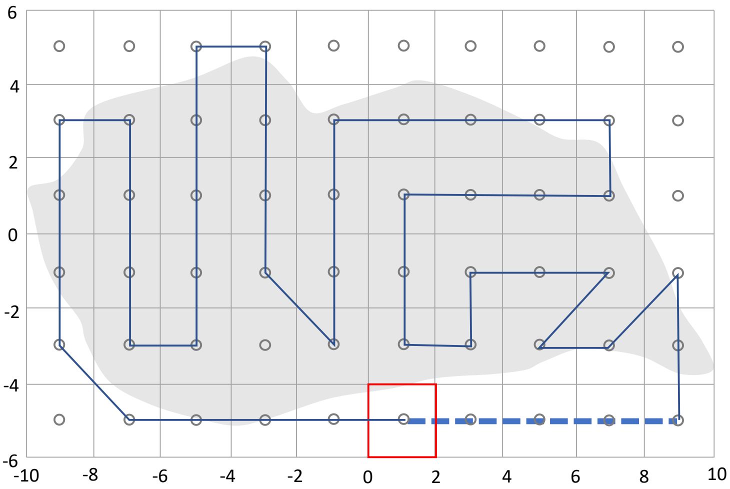
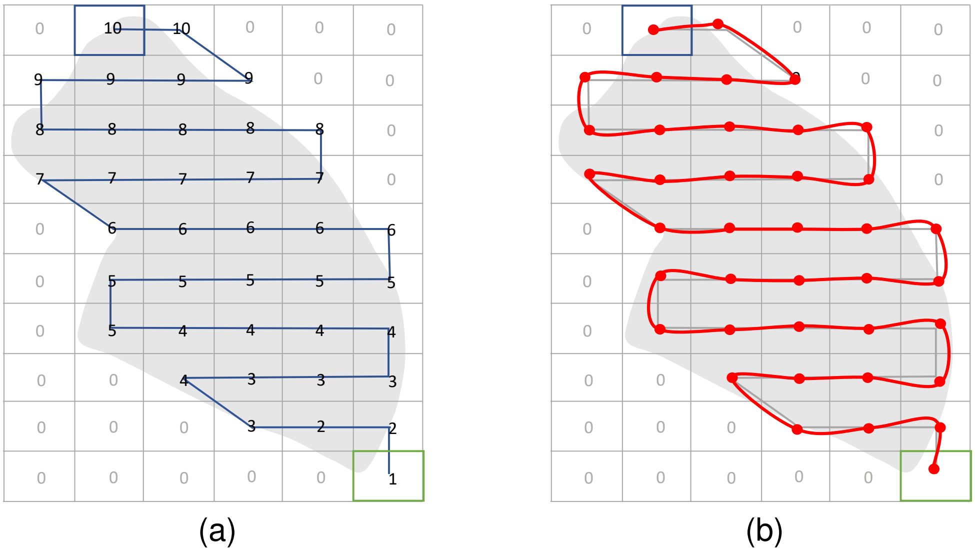


Figure 23 – Grid-based method in an irregular-shaped area of interest. The path starts at the lower right corner, covers the entire area, and ends at the red cell. The dashed trajectory represents the returning path to the initial position.

The decomposed area is converted to a regular graph numerically labeled by the Wavefront algorithm, which is a flooding algorithm that marks the neighborhood adjacency of cells. A Deep-limited search (DLS) (LAVALLE, 2006) computes a full coverage path without revisiting previously explored nodes, and a backtracking procedure solves issues, such as the choice among neighbors with the same weight. The proposed method enables a simple and faster solution to achieve near-optimal results in complex areas with certain constraints.

NAM et al. (2016) present another approach exploring the Wavefront algorithm and the approximate cellular decomposition for coverage in agricultural areas. The coverage path is obtained over an area of interest labeled according to the Wavefront, as shown in Figure 24a, and smoothed through a cubic interpolation algorithm, as illustrated in Figure 24b. Different from VALENTE et al. (2013), the authors present a new optimization criterion for the mission execution time based on the path length and the number of turning maneuvers.



0	10	10	0	0	0
9	9	9	9	0	0
8	8	8	8	8	0
7	7	7	7	7	0
0	6	6	6	6	6
0	5	5	5	5	5
0	5	4	4	4	4
0	0	4	3	3	3
0	0	0	3	2	2
0	0	0	0	0	1

Figure 24 – Coverage trajectory smoothed over an irregular-shaped area: (a) Wavefront algorithm, (b) Cubic interpolation algorithm.

BOUZID; BESTAOUI; SIGUERDIDJANE (2017) propose an optimal CPP algorithm for exploring a map containing Points of Interests (POIs). A quadrotor UAV should perform a minimum path connecting the POIs while avoiding obstacles with distinct formats to guarantee complete coverage in the area. The proposed algorithm computes the cost to explore the points in the neighborhood, determining the visitation order to minimize the total distance. After visiting the POIs a single time, the vehicle should return to the initial position. In this way, the problem is treated as the Traveling Salesman Problem (TSP), and the overall shortest path can be computed using a Genetic Algorithm (GA). The authors consider the accumulative Euclidean distance among the points as the primary performance metric. Moreover, they assume the energy consumption to be constant during the whole mission, measuring it in terms of time.

In real-world applications, the quadrotor may need to recharge or replace the battery a couple of times while performing a task considering its limited onboard energy. Thus, the authors explore a different possibility inspired by the Vehicle Routing Problem (VRP). This solution finds the minimum group of shortest trajectories in situations with only one or several initial positions. In this case, the aerial vehicle performs the designated task and keeps coming back to the base station every time it needs to recharge its battery. At the same time, the UAV also downloads the acquired information during each part of the mission.

A coverage approach for precision agriculture involving a team of heterogeneous quadrotors is discussed by BARRIENTOS et al. (2011). The mission includes two main phases, task subdivision and allocation, and coverage path planning. In the first phase, the division of the area and the assignment of the resulting subregions are simultaneously accomplished in a distributed way using a negotiation protocol.

The vehicles have to analyze the cost and the reward of a task. They have internal parameters and use them to evaluate the cost of task execution and the initial cost to move from the current location to the search area. They analyze the prize related to the mission accomplishment and also have to consider some restrictions as no-fly zones, turning angles, and embedded sensors to perform the coverage. The objective function sums several terms with different weights, including the task dimension and the distance from the starting point to the mission location. Furthermore, a penalty can be applied if one task superposes another or the exploration goes beyond the general area.

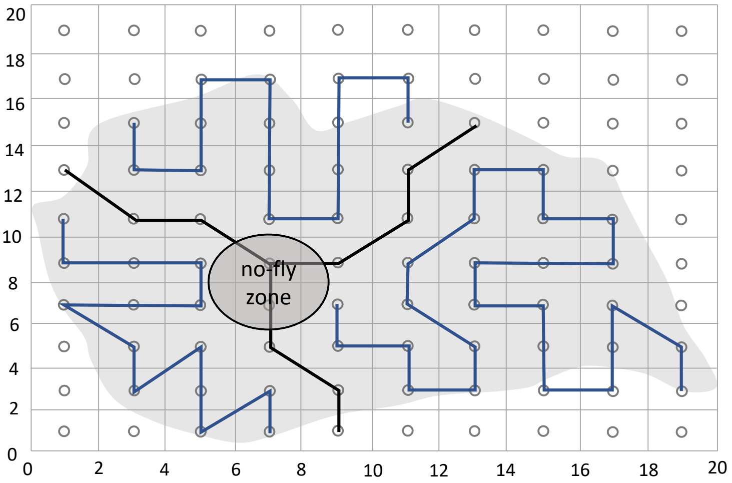


Figure 25 – Coverage path planning in irregular-shaped areas containing no-fly zones in the subregions boundaries for collision avoidance.

In the second phase, the area is discretized into a regular grid using approximate cellular decomposition. Each vehicle uses a Wavefront algorithm to compute a coverage path for its assigned subregion attempting to minimize the flight time, the number of turning maneuvers, and the amount of revisited cells. Furthermore, the vehicles keep the altitude constant to guarantee the desired resolution according to the field of view of the onboard camera. The authors also present a control system to improve the vehicle's altitude stabilization during high-speed maneuvers. They performed experiments using three UAVs in a vineyard field, characterized by its irregular-shaped format and its changeable altitude profile. The subregions have boundaries delimiting a safe zone to separate them, and the vehicles should not cross these areas to avoid collisions, as illustrated in Figure 25.

A meta-heuristic exploring the jazz musician's improvisation and focusing on precision agriculture is proposed by VALENTE et al. (2013). This algorithm is called Harmony Search (HS) and attempts to minimize the number of turning maneuvers in irregular-shaped areas. The algorithm contains a Harmony Memory (HM), which consists of a matrix where the lines are composed of vectors comprising possible solutions, while the columns represent decision variables. The last column contains the cost function value.

The proposed approach randomly initializes the matrix and starts a process called

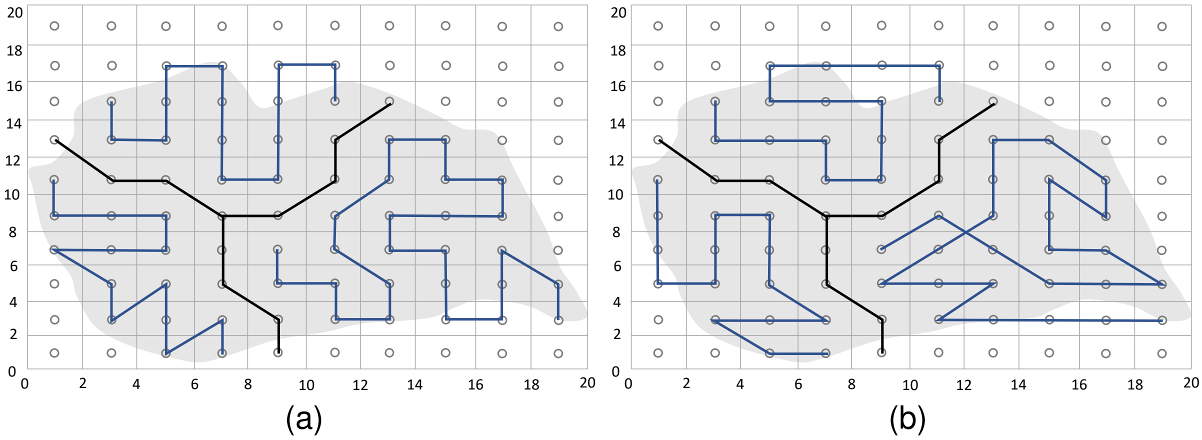


Figure 26 – Comparison between different coverage path planning approaches in irregular-shaped areas using three UAVs: (a) Wavefront Algorithm (BARRIENTOS et al., 2011), (b) Harmony Search (VALENTE et al., 2013).

improvisation, where the exchanging between neighboring cells creates new vectors according to a certain probability. These new vectors are also known as harmonies. If a new harmony has an improvement in comparison to the current worst solution, it replaces the old one in the matrix. Otherwise, the matrix remains unchanged. The authors compare the HS approach against the Wavefront algorithm employed by BARRIENTOS et al. (2011) using the same scenario with three subregions, as shown in Figure 26. The HS computes coverage paths containing a fewer number of turning maneuvers than the Wavefront but requires extra computational time. However, the authors do not consider computational time as a problem since the planning is offline.

SADAT; WAWERLA; VAUGHAN (2014) and SADAT; WAWERLA; VAUGHAN (2015) propose approaches for online non-uniform coverage path planning. In this context, the vehicle can change its altitude during the coverage according to the importance of each part of the area. The authors use a tree structure to deal with grids considering different resolutions. The closer the node is to the leaves, the higher is the resolution. SADAT; WAWERLA; VAUGHAN (2014) introduce three methods to explore the tree, Breadth-First strategy, Depth-First, and Shortcut Heuristic, while SADAT; WAWERLA; VAUGHAN (2015) propose an Hilbert-based approach, as shown in Figure 27, comparing it with the previous strategies and the back-and-forth pattern. The resolution increases by traversing down the tree when the UAV is visiting a zone of interest. In this way, the area that was being explored by a parent node now is being fully covered by their child nodes at an increased resolution. On the other hand, if the zone is not interesting, the search goes up in the coverage tree and decreases the resolution. Therefore, a single vehicle moves around the area, exploring zones at different altitudes to perform the mission.

High-resolution aerial sensing with multiple heterogeneous UAVs for non-convex areas is discussed by SANTAMARIA; SEGOR; TCHOUCHENKOV (2013). The workspace

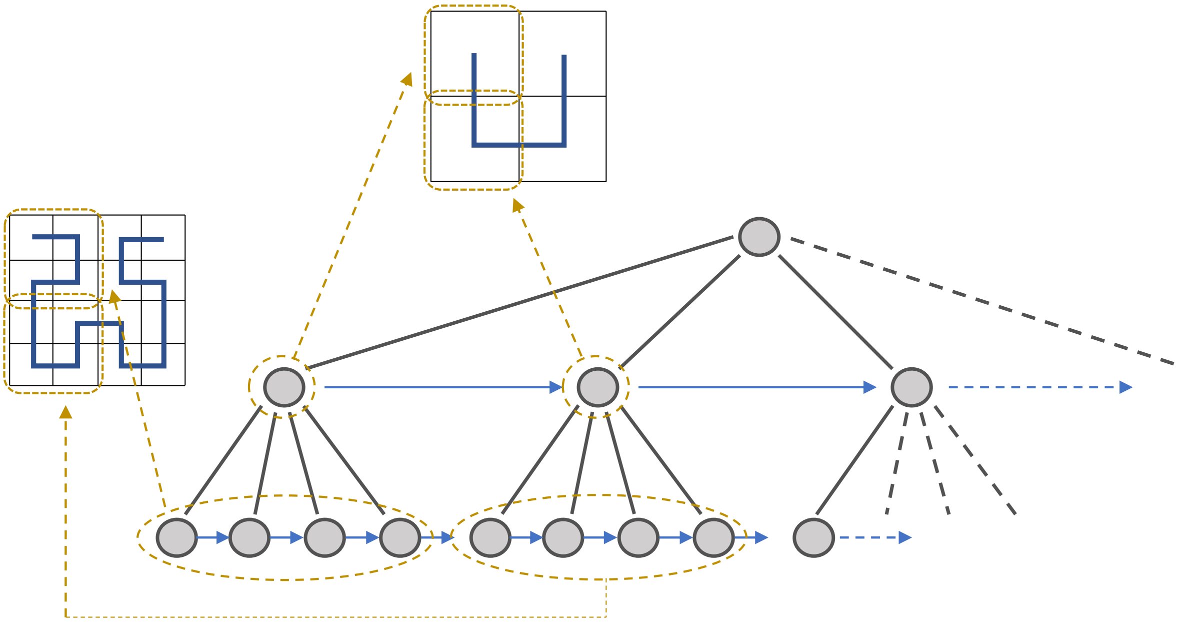


Figure 27 – Coverage performed at distinct altitudes with different resolutions according to the importance of each zone. The closer the node is to the leaves, the higher is the resolution.

is discretized into a grid of cells with different sizes through the approximate cellular decomposition because the vehicles have varied coverage range and image sensors, as illustrated in Figure 28. Initially, a rough estimation determines the portion of the area explored by the UAVs considering their onboard sensor footprint. A flooding technique selects a starting position and extends the neighborhood area until it reaches the estimated amount of cells. Places located outside the scenario or within no-fly zones are invalid cells. After the initial estimation, unsigned cells join the nearest area, and a re-balance is executed to distribute the cells uniformly.

Following the approach presented by SANTAMARIA; SEGOR; TCHOUCHENKOV (2013), flight paths to cover each of the sub-areas are computed using an algorithm to find long-distance segments of free cells, called strides. The algorithm computes the strides for all unexplored adjacent cells of the current position, selecting the longest line. The last cell of the chosen line turns into the current cell, repeating the process. When there are no unvisited neighbor cells, the algorithm generates trajectories from the current location to the closest unexplored cells, selects the smallest path, and adds the corresponding cells to the coverage plan. The shortest distance is also computed to return to the landing location. The authors integrate the proposed approach with the AMFIS (SEGOR et al., 2010), which consists of a ground control station for real-time vehicle controlling and monitoring.

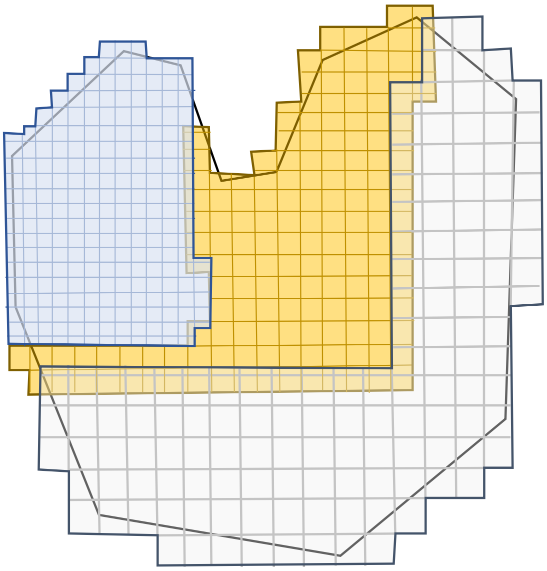


Figure 28 – Area of interest discretized into a grid through the approximate cellular decomposition with cells of different sizes.

3.3.2 Partial Information

Biological organisms are capable of adapting, evolving, and surviving in wild nature, a complex environment characterized by rapid changes, high uncertainty, and limited information (PFEIFER; LUNGARELLA; IIDA, 2007). Such sophisticated organisms have been successful in developing smart strategies to perform complicated tasks. Thus, these organisms have been the inspiration for many computational systems over the years to find solutions for optimization problems (FLOREANO; MATTIUSSI, 2008).

Biologically-inspired approaches consisting of algorithms based on fundamental aspects of natural intelligence, such as behavioral autonomy and social interaction, evolution, and learning, have emerged (FLOREANO; MATTIUSSI, 2008). Considering the CPP problem with aerial vehicles, several authors have explored different approaches in the literature, including real-time search methods (NATTERO et al., 2014), random walk (ALBANI; NARDI; TRIANNI, 2017), cellular systems (ZELENKA; KASANICKÝ, 2014; ZELENKA; KASANICKY, 2014b,a), evolutionary computation (PARADZIK; İNCE, 2016; TRUJILLO et al., 2016), and swarm intelligence (CHENG et al., 2009; KUIPER; NADJM-TEHRANI, 2006; ROSALIE et al., 2016). Coverage with uncertainty considering information points is also addressed (KHAN; YANMAZ; RINNER, 2014; LIM; BANG, 2010; POPOVIĆ et al., 2017,b; RAMASAMY; GHOSE, 2016, 2017). Most of the approaches are pheromone-based and explore the natural behavior of ants to guide the vehicles through a grid-discretized scenario.

3.3.2.1 Real-Time Search Methods

NATTERO et al. (2014) evaluates the performances of real-time search methods (RTSM)[1] using simulations with land robots and real experiments with UAVs. The authors analyze four different methods, including Edge Counting (KOENIG; SIMMONS, 1996), PatrolGRAPH* (CANNATA; SGORBISSA, 2011), Node Counting (PIRZADEH; SNYDER, 1990), and Leaning Real-Time A* (KORF, 1990). The simulations consist of a coverage time analysis, which is directly related by the authors to the energy necessary to execute coverage. Different experiment designs were conducted changing parameters such as grid size and topology, number of vehicles, and required visits. Due to the good results of the Node Counting over other approaches in simulated experiments, the authors implemented it in a real environment containing two UAVs, a quadcopter and a hexacopter.

The decision-making process is fully centralized in an off-board station using the Robot Operating System (ROS) (QUIGLEY et al., 2009), while the dynamics and the localization of the vehicles are controlled and monitored onboard with the assistance of the ETHNOS framework (PIAGGIO; SGORBISSA; ZACCARIA, 1999). The station is responsible for sending target positions to the vehicles while guiding their motions during the coverage mission. It also receives data about the current position of the vehicles. The communication between the base and the aerial vehicles is done using a ROS/ETHNOS interface. The system is not robust to failures since the UAVs depend on the communication with the off-board station to perform their motions. Signal interference or communication loss can compromise the system and put the mission in risk.

3.3.2.2 Random Walk

A reinforced random walk approach is employed as an exploration strategy for a weed mapping application by ALBANI; NARDI; TRIANNI (2017). A UAV swarm has the task of detecting the presence and the number of weeds on a field. It uses attraction and repulsion vectors to guide the coverage along the area. Each swarm member divides its search area into two semi-planes, tending to choose the semi-plane ahead. The more aligned a vehicle is in the sweeping direction, the more likely it is to move in this direction. The presence of other vehicles also influences the motion of a vehicle, pushing it to less frequently covered areas. The swarm trades information among its members to avoid cover already explored locations. It is also able to call upon its members using an attraction vector, leading them to potential areas to be explored.

[1] The term “real-time search method" refers to a heuristic that interleaves between planning and execution during a coverage task. The term was defined by KOENIG; LIU (2001).

3.3.2.3 Cellular Automata

A cellular automata approach based on a pheromone strategy is proposed for coordinating land robots in coverage missions by ZELENKA; KASANICKÝ (2014) and further explored for controlling two quadcopters in monitoring applications (ZELENKA; KASANICKY, 2014a). The approach consists of a set of rules which determines the actions of the robots. The area of interest acts as a shared memory among robots, containing virtual marks for coordinating them. These virtual marks emulate pheromone traces left during motion and symbolize the number of visits on each location. A GCS splits the area into digital cells, tracks the location of the UAVs, and avoids crashes among them.

The pheromone evaporation is emulated by ZELENKA; KASANICKY (2014b) to analyze the system reaction in case of loss of communication. The authors faced problems in implementing the pheromone degradation because it drastically drops the algorithm's efficiency, causing several collisions among UAVs. Another issue detected during the experiments was the elevated concentration of pheromones. When UAVs cover small places in the scenario, they may pass by the same cells multiple times in brief intervals. This situation increasingly raises the amount of pheromone in these places, blocking the passage of the remaining vehicles. The authors consider their approach as a decentralized system capable of adapting to different circumstances. However, UAVs do not decide which places to visit. They depend on a GCS containing a global vision of the map to process the information and indicate which locations they need to go in order to perform coverage.

3.3.2.4 Coverage with Uncertainty

LIM; BANG (2010) present a waypoint coverage solution for a surveillance mission over a hexagonal grid containing Information Points (IP). The IPs hold a value representing certainty about the scenario locations, as shown in Figure 29a. Each region has to be continuously monitored to keep the confidence level. When this value is sufficiently elevated, the point possesses reliable information and monitoring around these places is not necessary. However, this value decreases when a given location remains without any visitation for a while. Lower values mean not trustworthy information and demand attention for more observations to update the confidence level in these locations.

A set of surrounding points determines the confidence level of each region. In this way, the vehicles consider the average certainty of neighboring IPs, moving not only to the point but to the region with the lowest certainty. The authors also explore different cost functions, such as distance and interest level, to guide the vehicles to specific zones. They previously introduced these cost functions in another work (LIM; BANG, 2009). The A* algorithm is employed to generate the path to reach the next point chosen using the cost functions. As a cooperative surveillance mission, a subregion is assigned

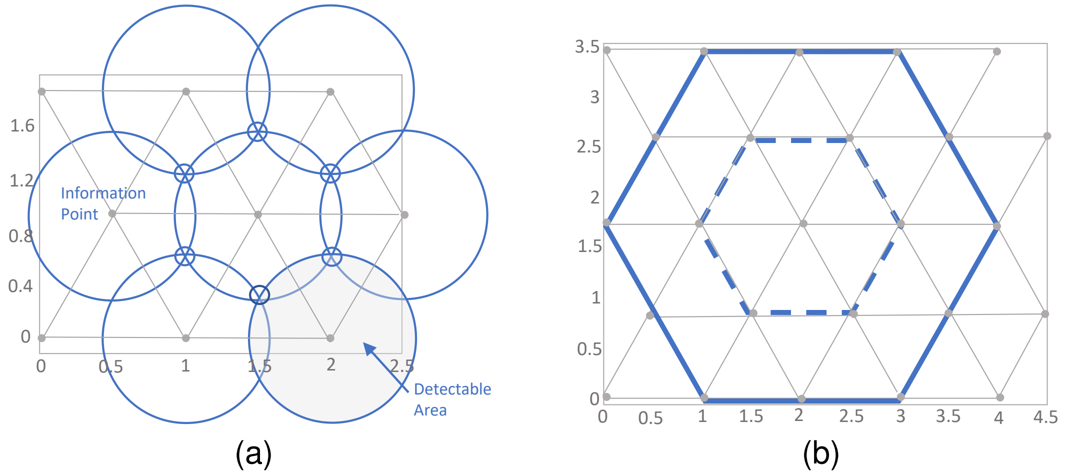


Figure 29 – Hexagonal grid method with information points representing the uncertainty: (a) Information Points, (b) Certainty calculation with IPs.

to each vehicle to avoid coverage overlapping.

A cooperative coverage strategy for a fleet of small-scale UAVs searching for target locations is explored by KHAN; YANMAZ; RINNER (2014). The approach consists of merging distributed information from multiple vehicles with restricted sensing capabilities. Combining local maps allow vehicles to improve their sensing skills and accelerate coverage. These vehicles can move in four directions or remain in the current cell while performing coverage over a regular grid area. The mission displays false alarms triggered according to a certain probability, which can be detected by the vehicle's onboard sensors.

POPOVIĆ et al. (2017) introduce an Informative Path Planning (IPP) for weed detection in precision agriculture using UAVs. The authors use a fixed-horizon approach for adaptive planning, alternating between plan execution and replanning. They employ a two-stage replanning continually refining the 3D path of an aerial vehicle while respecting its motion restrictions. This optimization is obtained using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

The authors also present a multi-resolution IPP for terrain monitoring with aerial platforms (POPOVIĆ et al., 2017b), focusing on biomass monitoring instead of binary classification of weed occupancy. The method builds a probabilistic map with all the visual data gathered in flights at distinct altitudes. The authors compare and evaluate both proposed methods against the back-and-forth flight pattern and the state-of-the-art IPP algorithm in simulated experiments. Performing real flights, they run their IPP approach in an AscTec Pelican UAV (POPOVIĆ et al., 2017) and on a DJI Matrice 100 (POPOVIĆ et al., 2017b), emulating the weed classifier through the use of AR tags and mimicking the terrain monitoring on painted green sheets.

RAMASAMY; GHOSE (2016) address a learning-based algorithm for persistent surveillance using UAVs. The problem consists of a continuous CPP, where UAVs should periodically monitor areas and minimize the interval between the visits on each location. The frequency of visits depends on the area profile and should be increased in attractive regions and reduced in hazardous locations. In some cases, the coverage should be avoided depending on the risk. RAMASAMY; GHOSE (2017) extend their work dedicating more attention to the preferential surveillance over a known area. They explore an approach considering a variety of priority specifications. Besides, they present further simulation results to support more detailed analysis.

3.3.2.5 Genetic Algorithm

A pheromone-based strategy combined with Genetic Algorithm (GA) is explored by PARADZIK; İNCE (2016) for coordinating a fleet of UAVs. The GA splits the area of interest and allocates the resulting rectangular subregions to the UAVs, as illustrated in Figure 30. Two types of pheromones are used by the authors: search and path. The former one is related to uncertainty and represents the needing coverage of a zone. The latter one marks the places covered by other UAVs, preventing crashes among them, while decreasing revisitation. The pheromone-based GA generates the subregions by maximizing the search pheromone and minimizing the path pheromone. The coverage trajectories performed by UAVs inside each subregion consist of simple back-and-forth motions. Furthermore, the trajectory to move the UAV from its current position to the closest vertex of the designated subregion is planned using A* algorithm.

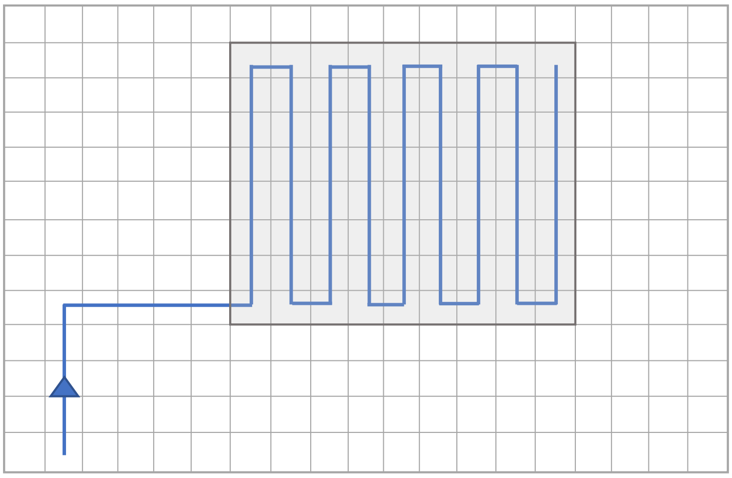
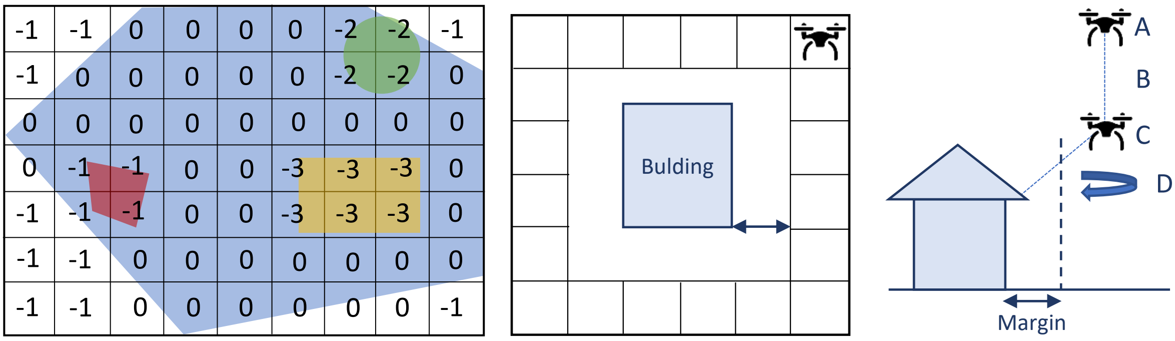


Figure 30 – Area of interest decomposed into rectangular subregions using a pheromone-based strategy combined with Genetic Algorithm. The subregions are covered with back-and-forth motions.

Considering another effort based on a GA, TRUJILLO et al. (2016) propose a coverage path planning approach with 3D structure mapping to handle scenarios with obstacles and buildings. The area of interest is a polygon discretized as a grid containing buildings (yellow), tall vegetation (green) and forbidden flight zones (red), as shown in Figure 31a. The following values label the cells according to the type of area: 0 for free areas, -1 for forbidden or external zones, -2 for tall vegetation which demands altitude adjustments, and -3 for buildings. GA generates the coverage path considering only the free spaces and the areas with vegetation below the altitude flight.



-1	-1	0	0	0	0	-2	-2	-1
-1	0	0	0	0	0	-2	-2	0
0	0	0	0	0	0	0	0	0
0	-1	-1	0	0	-3	-3	-3	0
-1	-1	-1	0	0	-3	-3	-3	0
-1	-1	0	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0	-1

(a) (b)

Figure 31 – Grid representation of the area of interest and main steps of 3D mapping method: (a) Labeled grid area, (b) Structure mapping.

During the path execution, adjacent cells containing buildings may be detected, triggering the 3D mapping. In this case, the vehicle stops the coverage and surrounds the building at a certain distance to photograph it, following the next steps: hovers at the current altitude (A), goes up/down according to the height of the building (B), changes the camera angle (C) and flies surrounding the building (D), as shown in Figure 31b. The vehicle continues the original coverage path after completing the mapping. Figure 32 illustrates a complete path avoiding forbidden flight zones while surrounding the buildings.

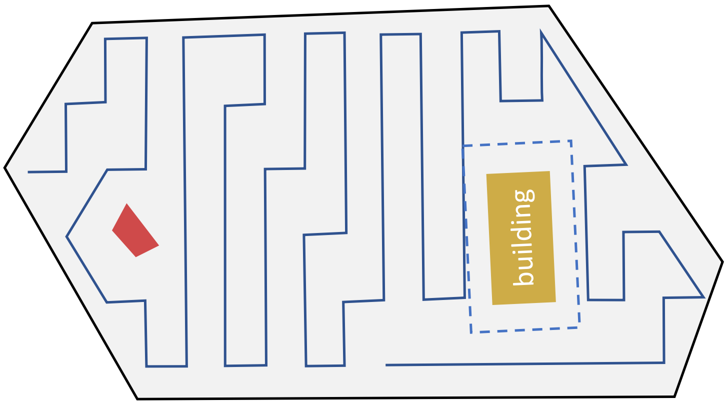


Figure 32 – Complete coverage path with forbidden flight zone and building generated by the Genetic Algorithm. The dashed line marks the 3D mapping around the building.

The approach proposed by TRUJILLO et al. (2016) is extended by DARRAH et al. (2017) for coverage missions over larger locations using multiple multirotors. The area of interest is partitioned into equitable sub-areas to be covered by various vehicles or by several flights performed by a single UAV. The partitioning method applies the flood fill algorithm integrated with game theory. In this scenario, each UAV is a player and has a starting position. These players take turns flooding the neighbor cells according to a predefined pattern in a diamond shape, as illustrated in Figure 33. The corresponding order is from left to right, flooding the closest neighbors. The players can not flood building cells or cells previously occupied by other players, and the sub-areas are not equally sized. A smaller sub-area may contain building structures to be mapped, while a more extensive area may present areas of avoidance. Thus, the partitioning method balances the tasks and guarantees an approximate amount of work for each assigned UAV. The coverage paths for each sub-area are obtained using an improved version of the approach proposed by TRUJILLO et al. (2016). The improvements ensure that the multirotor's trajectory ends as close as possible to where it started.

45	39	29	18	31	41	47
37	27	16	**8**	20	33	43
25	14	**6**	**2**	**10**	22	35
13	**5**	**1**	**S**	**4**	**12**	24
26	15	**7**	**3**	**11**	23	36
38	28	17	**9**	21	34	44
46	40	30	19	32	42	48

Figure 33 – Flood fill pattern with the starting position S and the ordered neighbor cells to be flooded from dark blue to light blue.

HAYAT et al. (2017) propose a Multi-Objective Path Planning (MOPP) with a Genetic Algorithm for Search and Rescue (SAR) missions using multiple UAVs. The mission is composed of two steps, search and response. The former one monitors an event (e.g., stationary target) by guaranteeing the full coverage in a given area. The latter spreads detection updates on the network. The planning task that occurs during the search phase is performed in a centralized way by the MOPP algorithm, while the time-to-complete the mission is minimized by the GA. This time comprises the period to discover the target and the period to configure a communication trajectory. Thus, the approach needs to optimize two main features, the area coverage, and network connectivity.

3.3.2.6 Ant Colony Optimization

Ant Colony Optimization (ACO) is adapted for coverage missions with multiple UAVs by KUIPER; NADJM-TEHRANI (2006). In this approach, the vehicles share a virtual pheromone map indicating recently visited areas. High rates of repulsive pheromones guide the UAVs to unexplored regions. Based on this study, ROSALIE et al.

(2016) introduce the Chaotic Ant Colony Optimization to Coverage (CACOC), which is a technique integrating the ACO with the chaotic dynamical system to surveillance missions in a military context. The approach allows the GCS operator to forecast the UAVs paths while keeping it unpredictable to the enemies. Although there is no need for communication among vehicles and the base to track the position of the UAVs, the swarm still shares a virtual pheromone map. ROSALIE et al. (2017) further explore the CACOC approach by evaluating its coverage performance by using the V-Rep simulation environment (ROHMER; SINGH; FREESE, 2013).

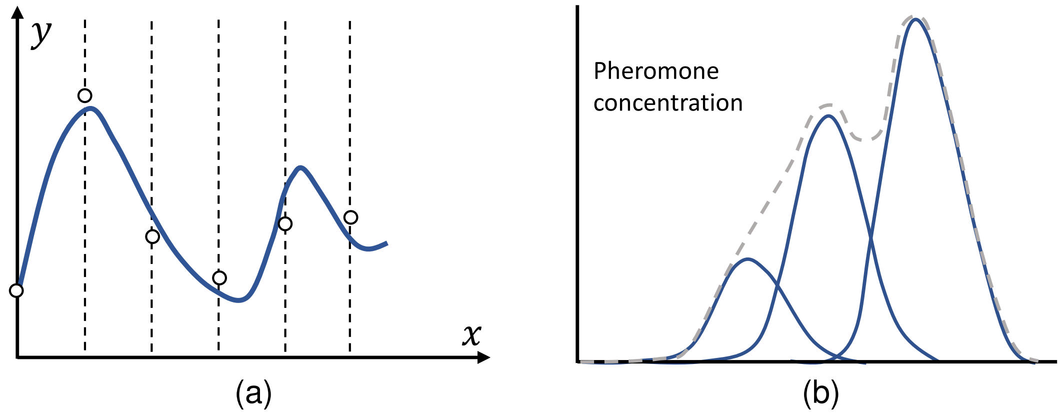


Figure 34 – Cooperative approach using ACO with Gaussian distribution functions: (a) Control points, (b) Gaussian distribution functions.

CHENG et al. (2009) propose another bio-inspired approach for cooperative coverage. This approach represents the path of each vehicle as a B-spline curve containing control points, as illustrated in Figure 34a. This optimization problem consists of maximizing the desirability of a path combining four functions: (i) path length, (ii) minimum turning angle, (iii) maximum pitch rate, and (iv) the superposition of the actual trajectory over different UAVs trajectories. The authors consider that the vehicle always moves from left to right, so the first and the last control points are at the borders of the area. The ACO algorithm optimizes the y-axis in the intermediate control points to maximize the coverage. During the algorithm iterations, several ants are launched in the scenario, passing by the initial, intermediate, and final points.

Gaussian distribution functions represent the pheromone concentration left by each ant at the control points, being superposed to create a joint distribution function, illustrated in Figure 34b. The resulting function is rescaled to generate the probability density function, which indicates the amount of pheromones in different y-positions. During subsequent iterations, the ants select the control point positions following this probability density function. This function employs a pheromone evaporation factor to avoid local optimal points. At the end of the algorithm, the zones comprising the most significant amount of pheromones are selected as intermediate control points to create a full path for the UAV.

The coverage path planning approaches for unmanned aerial vehicles presented in this section are summarized in Table 24, Table 25, and Table 26 at the appendix of this thesis.

4 PROPOSED APPROACHES

This chapter presents a detailed description of the proposed energy-aware approaches for the Coverage Path Planning problem with UAVs. The solutions consist of flight patterns, complete algorithms, and pheromone-based methods suitable for regular and irregular-shaped areas of interest. The first two methods deal with scenarios where the UAVs contain full information to perform the coverage, while the latter one handles with a dynamic environment having only partial information.

Section 4.1 presents an energy-aware coverage path planning algorithm called E-Spiral. This algorithm performs missions in regular-shaped areas consisting of concave and convex polygons. The coverage path starts at the vertices of the polygon and slowly decreases its radius towards the center of the area. The approach considers specific requirement applications, such as overlapping and image resolution, to guarantee mission success in photogrammetric applications. Furthermore, the algorithm explores an energy model to adopt optimal speeds in straight segments of the path to minimize energy consumption.

Next, Section 4.2 describes an energy-aware grid-based solution called EG-CPP. This algorithm generates coverage paths for mapping missions over irregular-shaped areas containing no-fly zones. The algorithm improves an existing grid-based method by replacing its original cost function to an energy-aware cost function. As a further contribution, two pruning techniques improve the performance of the algorithm. This improvement speeds up the computational time of the algorithm, allowing to explore all the different starting positions in the workspace and saving even more energy.

Finally, Section 4.3 depicts an energy-aware pheromone-based solution for patrolling missions called NC-Drone. This approach extends an existing real-time search method aiming at minimizing the number of turning maneuvers while maintaining the unpredictable behavior during coverage. The algorithm explores a matrix-representation to store the visited cells in the vehicle's memory and adopts synchronization schemes to share the information between the UAVs. Cooperative strategies improve the algorithm and further explore the problem considering relevant aspects, such as time, uncertainty, and communication.

4.1 E-Spiral

Most UAVs nowadays engage in missions based on CPP using simple geometric flight patterns, such as back-and-forth (BF) and spiral (SP). Despite the extensive use of these flight patterns, there is no general agreement about which one is more appropriate. Some studies highlight that the BF pattern is better (ÖST, 2012), while other works claim that the SP pattern overcomes the BF (ANDERSEN, 2014), depending on the adopted criteria. Furthermore, the patterns are usually evaluated using the number of turning maneuvers as the primary performance metric (LI et al., 2011; TORRES et al., 2016). However, when dealing with UAVs on CPP missions, the number of turns is not an accurate metric for different flight patterns, and several additional aspects must be considered, such as the vehicle dynamics, the distance traveled, and the optimal speed adopted during the path. These aspects directly affect the energy consumption of UAVs during the missions, especially in quadrotors whose flight autonomy is limited. Moreover, it is important to highlight that, in literature, it is still missing an energy-aware spiral pattern with accurate overlapping for photogrammetric applications using different speeds to optimize and save energy.

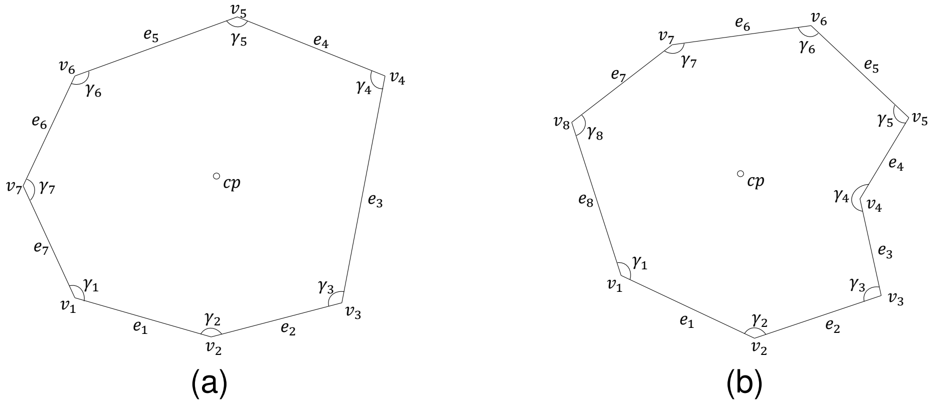


Figure 35 – Areas of interest supported by the E-Spiral CPP algorithm: (a) Convex Area, (b) Concave Area.

The energy-aware spiral flight pattern algorithm (E-Spiral) consists of a sequence of maneuvers performed by the aerial vehicle to cover an area of interest using approximate circular motions. E-Spiral computes coverage paths for regular-shaped areas. Such areas comprise convex ($\forall i, \gamma_i < \pi$) and concave ($\forall i, \gamma_i > \pi$) polygons without inner no-fly zones, as illustrated in Figure 35. The polygonal area $A = \{v_1, ..., v_p\}$ can be described by an ordered set of p vertices, each one being represented by a pair of coordinates $(v_x(i), v_y(i))$ and containing an inner angle, denoted by γ_i. For each vertex v_i, the next vertex of the polygon in the considered order is denoted by $v_{next(i)}$, where $next(i) = i(\text{mod } p) + 1$. The edge between a pair of consecutive vertices v_i and $v_{next(i)}$ is denoted by e_i, and its length by $l_i = ||v_i - v_{next(i)}||$.

4.1.1 Spiral Coverage Path for Photogrammetric Sensing

E-Spiral considers application requirements, such as overlapping and resolution, to compute the coverage path for a photogrammetric sensing application. The main goal of this application is to generate an orthomosaic composed of a set of overlapping aerial photographs. In this way, the spiral coverage algorithm must generate a trajectory guaranteeing the necessary amount of both forward and lateral photographic overlap. These overlaps are crucial to identify common points between the pictures in a post-processing phase. In this phase, all pictures are stitched together to create one large high-quality picture.

The centroid point cp is computed based on the vertices of the polygon. Then, the minimum distance d_{cp} from the centroid point cp to the edges e_i is calculated. The distance d_{cp}, the lateral overlapping ov_x, and the frontal overlapping ov_y are employed to calculate the number of inner layers needed to cover the area fully. The lateral and the frontal overlapping determines the distance between each inner layer and the distance between two consecutive waypoints, respectively. The greater the amount of lateral overlapping, the higher the number of inner layers. In this way, the UAV needs to execute a longer path to fully cover the same portion of the area, which increases the mission execution time and, consequently, the energy consumption to perform the mission.

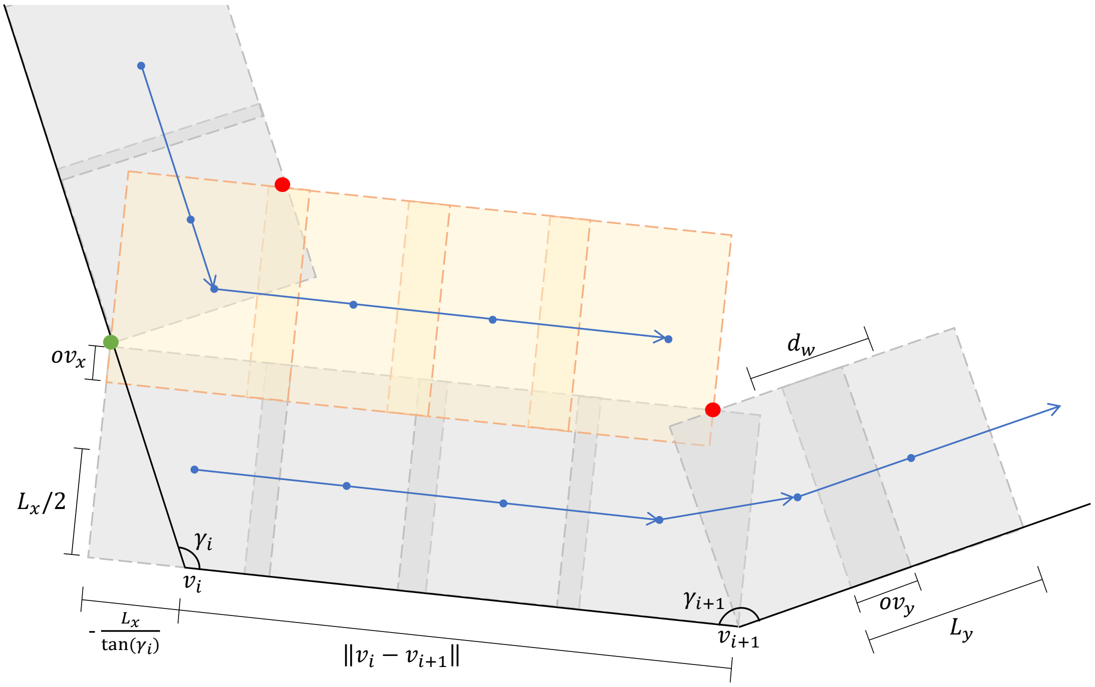


Figure 36 – E-Spiral pattern with inner layers, turning angles, and overlapping rates.

Another critical requirement for this type of application is the resolution necessary to obtain pictures with high quality for the final mosaic. Resolution is identified by the width w_{img} and height h_{img} of the image as well as the total number of pixels in the image. This requirement influences the altitude h that the aerial vehicle must fly over the area of interest. If the application needs a high resolution is necessary to fly at a lower altitude, which will also increase the number of inner layers. Therefore, the overlapping and the resolution are both constraints imposed by the mission. There is a trade-off between these two application requirements and energy consumption. The greater the overlapping and the resolution, the higher the total energy spent to perform the coverage mission.

The image resolution (w_{img}, h_{img}) and the field of view fov depends on the camera sensor characteristics. These parameters, along with the altitude h and the aspect ratio a_{ratio}, determine the size of the projected area of each picture on the ground. The width L_x and the height L_y of the projected area are computed as

$$a_{ratio} = \frac{w_{img}}{h_{img}},\ L_x = 2 \times h \times \tan\frac{fov}{2},\ L_y = \frac{L_x}{a_{ratio}}. \tag{4}$$

The spiral coverage path passes by the vertices of the area of interest at the beginning of the mission. Turning maneuvers with different inner angles are necessary to cover the workspace fully. After the first completed layer, the vehicle starts to decrease the radius at each step, flying towards the center, as illustrated in Figure 36. The gray color represents the first layer, which starts near v_i. The distance between the inner layers is set as $d_{layer} = L_x - ov_x$ and the number of layers n_{layer} along the distance d_{cp} is computed as

$$n_{layer} = \left\lceil \frac{d_{cp} - ov_x}{d_{layer}} \right\rceil. \tag{5}$$

Once n_{layer} is rounded up, it is necessary to recompute ov_x and d_{layer} as follows:

$$\hat{ov}_x = \frac{n_{layer}L_x - d_{cp}}{n_{layer} - 1},\ \hat{d_{layer}} = \frac{d_{cp} - L_x}{n_{layer} - 1}. \tag{6}$$

The distance d_w between consecutive waypoints in a straight line is set as $d_w = L_y - ov_y$. The number of waypoints n_w along a straight path of length $d = l_i$ is computed as

$$n_w = \left\lceil \frac{d - ov_y}{d_w} \right\rceil. \tag{7}$$

Note that an additional part $max(0, -L_x/tan(\gamma_i))$ is added to d only at the beginning of the path to connect the final part of the first layer with the beginning of the second layer, marked as a green circle in Figure 36. Since there are different distances between each pair of consecutive vertices, different overlapping rates are necessary at each straight line, respecting the minimum value defined by the mission. Also, once n_w is rounded up, it is possible to increase the overlap at the value $\hat{ov}_y$ such that $n_w(L_y - \hat{ov}_y) + \hat{ov}_y$

becomes exactly equal to d. That is

$$\hat{ov}_y = \frac{n_w L_y - d}{n_w - 1}. \tag{8}$$

In this way, the distance between two waypoints becomes

$$\hat{d}_w = \frac{d - L_y}{n_w - 1}. \tag{9}$$

Recomputing the overlapping based on the number of waypoints will increase the picture superposition without incrementing the number of layers, and hence, the traveled distance. The higher the overlap of the images, the easier it will be to identify points in common for the reconstruction of the mosaic. Intersection points are marked between the projected area of the last picture of the current line and the first of the next line. These intersection points are illustrated as red circles in Figure 36. After covering an entire layer, the intersection points form the new vertices of the area of interest, and the coverage continues in the next segment represented by the orange color. The procedure for generating the spiral coverage path is summarized in Algorithm 1.

Algorithm 1 E-Spiral Algorithm

```
Input: A set of vertices ${v_1, ...v_p}$
Output: A set of waypoints ${w_1, ...w_p}$
Compute the $cp$ of the polygon specified by $A$
Calculate the $d_{cp}$ from $cp$ to the borders of the area
Compute $n_{layer}$ by Eq. (5)
Compute $\hat{ov}_x$ and $\hat{d}_{layer}$ by Eqs. (6)
for $j = 1$ to $n_{layer}$ do
    for $i = 1$ to $p$ do
        Compute the distance $d_i$ and the turning angle $\gamma_i$
        Compute $n_w$ by Eq. (7)
        Compute $\hat{ov}_y$ and $\hat{d}_w$ by Eqs. (8) and (9), respectively
        Rotate and place the first waypoint at a distance $(L_x/2, L_y/2)$ from $v_i$ and all
            other $n_w - 1$ waypoints at a distance $\hat{d}_w$ from each other
    end for
    Calculate the intersection points of the projected areas of $layer_j$
    Update the vertices with the intersection points reducing the area
end for
Go from the final waypoint $w_p$ to the initial waypoint $w_1$ through a straight line
```

4.1.2 Energy Estimation and Optimal Speed

The energy cost prediction of a given path is fundamental for proper and safe coverage path planning operations. A wrong estimation may lead to unexpected crashes, while a too simplistic one may underuse the full potential of the battery, decreasing the task performance. DI FRANCO; BUTTAZZO (2016) split a generic UAV flight path into simple basic measurable maneuvers, i.e., climb/descend, accelerate/decelerate, fly at a constant speed, and rotate. They measured the power consumed during such operations and built an energy model upon them. The model is treated as a black box and, thanks to the real measurements, also considers external forces, such as the drag of the vehicle thus making the energy model more accurate. In particular, it is possible both to estimate the energy cost of a path (splitting it into multiple straight lines) and also to find the optimal speed that minimizes such amount of energy.

The relationship between speed and power is well known in the literature (JOHNSON, 2013). DI FRANCO; BUTTAZZO (2016) exploit the energy model to compute the integral of the energy for a given distance d to find the optimal speed that minimizes the energy necessary to travel that portion of the path. This optimal speed exists due to the total drag force curve, which combines the parasite drag and the induced drag. As the speed increases during steady flights, the parasite drag increases while the induced drag decreases (JOHNSON, 2013). This behavior leads to a minimum value for the drag curve, where the optimal speed requires less power to perform the flight, consequently saving energy. The energy model can also be used to estimate a generic path. Given m waypoints, it is possible to estimate the energy (and time) of the path as follows:

$$
\begin{aligned}
E_{path} = E_{climb}(0, h) + E_{desc}(h, 0) +\\
\sum_{i=0}^{m} \big(E_{acc}(0, v_i^*) + E_v(d_i, v_i^*) + E_{dec}(v_i^*, 0)\big) +\\
\sum_{i=0}^{m} E_{turn}(\gamma^{\{i\}})
\end{aligned}
\tag{10}
$$

where E_{climb}/$E_{descend}$ computes the energy for climbing/descending to/from an altitude h, the first summation splits a straight line in three different phases (acceleration, deceleration, and constant speed) where v_i^* can be constant or set to an optimal value that minimizes the straight distance for that line. Finally, the second summation takes into account the energy required for performing a rotation. All these components are pre-calculated and stored in a look-up table to speed up the computational time of the energy estimation.

4.1.3 Optimal Speed for Spiral Coverage Paths

The energy model proposed by DI FRANCO; BUTTAZZO (2016) presents an accurate energy estimation considering back-and-forth coverage paths performed with an IRIS quadcopter. In this scenario, the model considers that the UAV starts from zero speed, reaches and keeps a constant speed, and then decelerates until zero before the turning maneuver at the end of each straight line. In this thesis, we improve the energy model to deal with more complicated maneuvers, such as the ones performed during spiral paths, where the UAV decelerates until a given speed different from zero, performs the turn while moving, and then accelerates again.

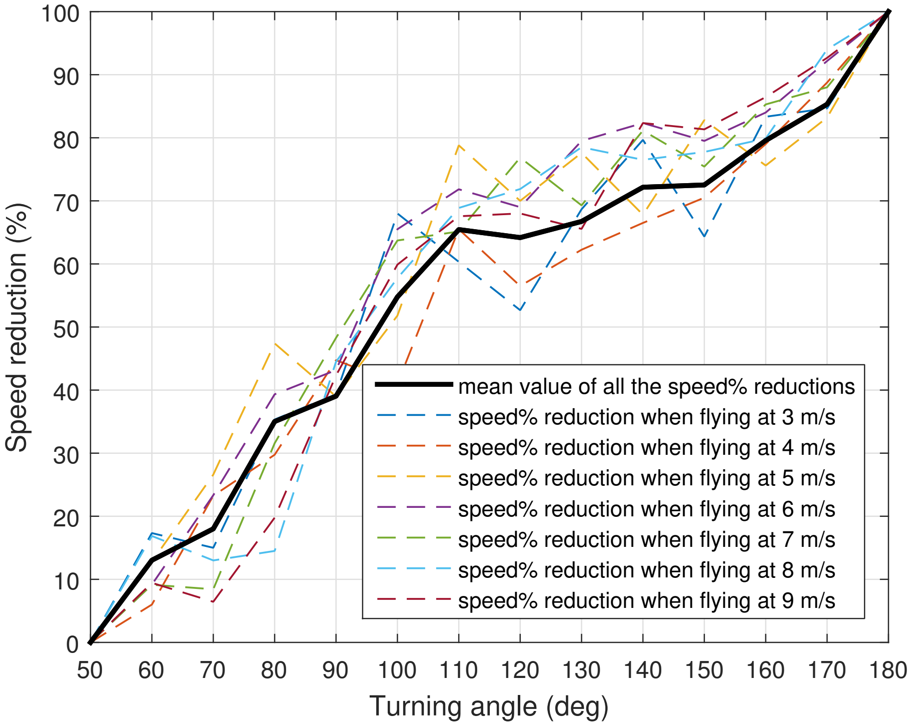


Figure 37 – Entrance speed as a function of the turning angle

Performing a set of real flight experiments with an IRIS quadcopter, we evaluate the speed variation according to the executed turning angle. We performed seven flights, each one at a different speed varying from $3m/s$ to $9m/s$ but covering the same path containing turning maneuvers varying from 60° to 180°. Results showed that the UAV decreases the flight speed according to a certain percentage when entering a curve with a specific angle, as shown in Figure 37. The figure illustrates the speed reduction percentage as a function of the turning angle, given the different initial speeds. The dashed colored lines represent the percentage reduction when performing a turn given an initial speed, while the solid black line represents the mean value for each angle.

Note that this information can be extracted by analyzing the UAV controller, but depends on the firmware/model version of the vehicle. It is possible to improve the energy model, maintaining it as a black-box independently of the controller by observing the UAV behavior through real experiments. One may apply the equations of this energy model for different vehicles and controllers. However, it is necessary to reproduce the same experimental procedure of gathering initial data by performing real flights under different circumstances. Such circumstances include performing turning maneuvers with variable angles and flying at different speeds.

Knowing the entrance speed of the vehicle when it performs a given angle allows us to modify the energy model when performing the integrals for the acceleration/deceleration/constant-speed phases. The optimal speed can be computed as follows:

$$E_d(v, d, \gamma) = \int_{v_{in}}^{v} P_{acc}(v)dv + \int_{0}^{t(v)} P(v)dt + \int_{v}^{v_{out}} P_{dec}(v)dv \tag{11}$$

where $v_{out} = f(v, \gamma)$ is the entrance speed when performing the next turn with angle γ. The function $f(v, \gamma)$ is a look-up table based on the data showed in Figure 37.

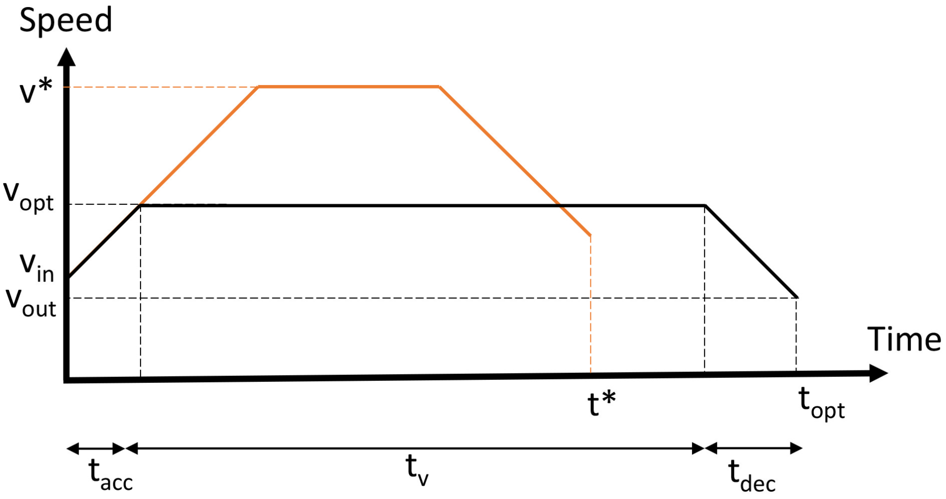


Figure 38 – The time needed to travel a straight distance can be split into three components. During t_{acc}, the quadrotor accelerates from v_{in} to a final v. Then, during t_v it flies at a constant speed and during t_{dec} it decelerates until v_{out}, that is computed considering v and the rotation angle. The optimal speed can be computed using Equation (11).

Given an initial speed v_{in}, the traveled distance d, and the turning angle γ, the energy model can estimate the optimal speed v and the amount of energy E_d required to fly the traveled distance d in spiral paths. Figure 38 shows a graphical illustration of the modified equations. The distance d can be covered at different speeds, leading to different travel times. However, we are interested in minimizing the energy, not the time, since a shorter time may require a more significant amount of energy depending on the circumstances. Different from the original energy model, v_{in} and v_{out} depend on the specific path and are not always zero.

4.1.4 Discussion

The E-Spiral is an energy-aware coverage path planning algorithm specially designed for a photogrammetric sensing application. This application presents two major requirements consisting of overlapping and resolution. Such constraints directly affect the total energy consumption of a UAV when performing a coverage mission. It is necessary to fly at lower altitudes to obtain pictures with better quality and with more points in common to ease the post-processing phase of generating an orthomosaic. However, flying at lower altitudes increments the path length needed to cover the same area, leading to increased flight time to perform the mission. Therefore, the greater the overlapping, and the higher the resolution, the higher the energy consumption.

Our algorithm generates coverage solutions for regular-shaped areas while dealing with the specific constraints imposed by the mission. The coverage starts at the vertices of the area. It slowly decreases its radius towards the center until the area is fully covered. We also improved the energy model previously proposed by DI FRANCO; BUTTAZZO (2016) through a set of real flight experiments. The improved energy model was able to estimate the energy spent by our spiral path correctly, as will be stated in Section 5.1.

Furthermore, we were able to determine the optimal speeds to be adopted in each straight segment of the path, minimizing the total energy consumption according to the mission constraints. We can obtain these optimal speeds in steady flights due to the total drag force curve, which combines the parasite drag and the induced drag. The higher the speed, the higher the parasite drag and the lower the induced drag. This particular behavior during the real flights leads to a minimum value in the drag curve, where there is an optimal speed that demands less power to perform the mission.

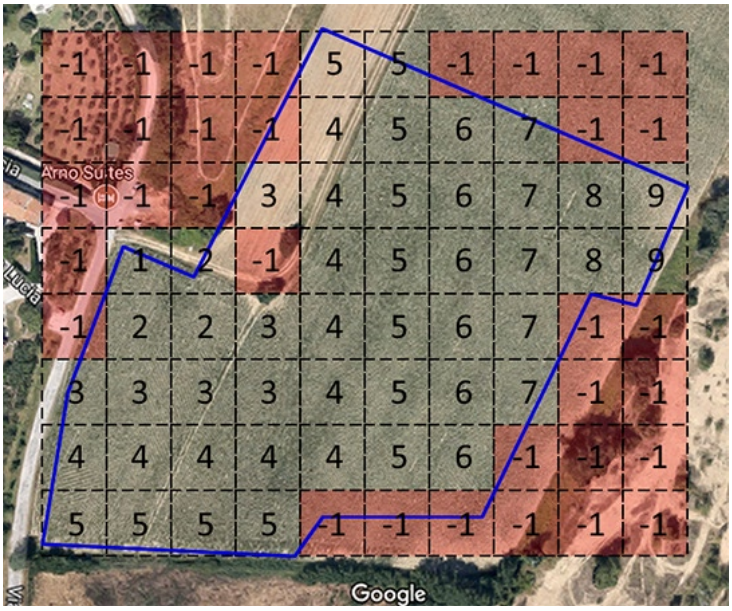
The E-Spiral algorithm presents two limitations. First, the adopted energy model requires an initial data gathering through real flights to work with different vehicles and controllers. Second, the generated coverage paths are not suitable for more complex areas of interest containing inner no-fly zones. In Section 4.2, we present an Energy-aware Grid-based Coverage Path Planning Algorithm (EG-CPP) able to handle such complex irregular-shaped areas.

4.2 EG-CPP

Geometric flight patterns, such as back-and-forth and spiral, have been widely employed in coverage path planning problems using UAVs in regular-shaped areas. These areas consist of simple rectangular forms or concave/convex polygons usually without inner no-fly zones. However, these flight patterns may be inefficient depending on the complexity of the shape of the area. In order to deal with irregular-shaped areas in CPP missions, VALENTE et al. (2013) proposed a more sophisticated algorithm using a Deep-limited search (DLS) with a backtracking procedure over a scenario discretized into a regular grid through approximate cellular decomposition (CHOSET, 2001).

An irregular-shaped area consists of a concave/convex polygon, which may contain obstacles and no-fly zones inside or outside. Despite providing a feasible solution in CPP missions over such areas, the algorithm proposed by VALENTE et al. (2013)

presents high computational time during the offline search phase due to the complexity of the area. The approach may need several hours to find a path on a scenario with a reasonable number of cells, e.g., 50. The approach is impractical for real-world applications considering the amount of time spent for a single run. Moreover, the quality of the solution is based on an unreliable cost function that relies exclusively on the number of turns, which may provide more expansive paths in terms of real measured energy, while discarding promising solutions during the minimization process. Several authors consider the minimization of the turning maneuvers as the primary performance metric for CPP problems aiming at indirectly minimizing energy consumption (MAZA; OLLERO, 2007; LI et al., 2011; ARAUJO; SUJIT; SOUSA, 2013; JIAO et al., 2010; TORRES et al., 2016). We intend to show that it is possible to use an energy estimation to generate more economic paths in terms of energy instead of minimizing the turns.



-1	-1	-1	-1	5	5	-1	-1	-1	-1
-1	-1	-1	-1	4	5	6	7	-1	-1
-1	-1	-1	3	4	5	6	7	8	9
-1	1	2	-1	4	5	6	7	8	9
-1	2	2	3	4	5	6	7	-1	-1
3	3	3	3	4	5	6	7	-1	-1
4	4	4	4	4	5	6	-1	-1	-1
5	5	5	5	-1	-1	-1	-1	-1	-1

Figure 39 – Irregular-shaped area discretized into a regular grid with the starting position marked with number 1 and the surrounding neighbors with number 2, and so on. Obstacles and no-fly zones are marked with -1.

The proposed Energy-aware Grid-based Coverage Path Planning approach (EG-CPP) exploits the energy model improved in Section 4.1.3 to reformulate the cost function adopted by VALENTE et al. (2013). Furthermore, our algorithm includes two pruning techniques, drastically reducing the computational time (in the order of seconds) and saving even more energy in a real-world coverage path. From now on, we are going to call the approach proposed by VALENTE et al. (2013) as the grid-based method, while our approach will be called as EG-CPP.

Figure 39 illustrates an irregular-shaped area of interest in a region near the city of Pisa, Italy. The area of interest contains no-fly zones marked in red with the number -1. The grid-discretized area is converted to a regular graph numerically labeled by the

Wavefront algorithm. It consists of a flooding algorithm that marks the neighborhood adjacency of cells. The starting cell is marked with number 1, and all neighbor cells are marked with number 2, and so on. The farther the cell is from the initial position, the higher it is value. The search length can be limited to the number of vertices, and consequently, the search neither goes around in endless cycles nor visits a node twice. Exploring this area, we performed the real flight experiments and compared the grid-based method with the EG-CPP proposed in this thesis. Further details about our proposed approach will be presented in the next subsections, and the experimental results will be presented in Section 5.2.

4.2.1 Energy-aware Cost Function

The grid-based method presents a cost function based on the sum of angles to find the minimum-cost path to perform a complete coverage, as shown in Equation (12):

$$\Gamma = \sum_{i=1}^{m} \gamma_k^{\{i\}}, k \in \{135°, 90°, 45°, 0°\} \tag{12}$$

where m represents the number of waypoints of the path $\{i_1, i_2, ..., i_m\}$ and γ represents the angle of the i-th waypoint compressed by the k set. However, this is an unreliable cost function, which may provide more expansive paths in terms of real measured energy, while discarding other promising solutions during the minimization process. In literature, several authors often connect the number of turning maneuvers with the power consumption trying to minimize it to save energy. Such cost functions are simplistic and do not explore important details as acceleration and deceleration phases during the turns.

The energy model splits the path into a set of straight segments and rotations to predict the energy cost of a path. The energy-aware cost function exploits the energy model to account not only for the energy required for every turn but also for the energy needed when accelerating/decelerating and flying at a constant speed. Thus, it is possible to evaluate the path and estimate the total energy (and time) using the Equation (13) as follows:

$$\Gamma_E = \sum_{i=1}^{m} \big(\int_{0}^{v_i} P_{acc}\, \mathrm{d}v + P_{v_i} \Delta T_i + \int_{v_i}^{0} P_{dec}\, \mathrm{d}v \big) + \sum_{i=1}^{m} E_{turn}(\gamma^{\{i\}}) \tag{13}$$

where the first summation computes the energy consumed during a set of straight lines i by splitting it into three phases (acceleration, deceleration, and constant speed) and the second summation considers all the rotations of the path. P_{acc}, P_{dec}, and P_v define the power consumed when accelerating, decelerating, and flying at a constant speed. $E_{turn}(\gamma^{\{i\}})$ is the energy to rotate an angle γ at the i-th waypoint (computed as the power consumed when turning P_{turn} multiplied by the duration of the rotation). ΔT_i is the time when flying the portion of the path at a constant speed, and it is computed considering the total distance and constant acceleration and deceleration. The terms in Equation (13) are polynomial functions obtained through real measurements and allow reaching high accuracy in the energy prediction of a given trajectory (DI FRANCO; BUTTAZZO, 2016).

4.2.2 Algorithm Optimization

The proposed Energy-aware Grid-based Coverage Path Planning approach (EG-CPP) is presented in Algorithm 2. The green lines represent the additions made during the optimization process, and the red lines represent the modifications in the original algorithm. The next subsections present the two pruning techniques applied to the algorithm.

4.2.2.1 Pruning Technique #1: Storing the path with the minimum-cost

In the grid-based method, the nearest neighbor cell with the highest value is chosen to find a full coverage path. If two or more cells have the same highest value, all the cells must be explored. Every full coverage path passes through all nodes in the adjacency graph only once. A Deep-limited search is used to build a tree with all possible coverage paths.

During the recursive search (lines 6-29), the current cell is obtained from the last added neighbor (line 7) and marked as visited in the temporary matrix (line 8). Then, the neighbor cells with the highest values are selected (line 9). If there is no available neighbor, the search in the current path ends and it returns to the previously recursive call (line 10), where the path is added to the FIFO data structure (line 21). Otherwise, the search continues in all selected neighbors (lines 13-28) with a new recursive call for every one of them (line 20).

The grid-based method computes the entire cost of all possible paths at the end of the algorithm (line 30), including those paths whose cost are much higher than the minimum value. This repetitive operation is massive and can be prevented. Thus, we modified the algorithm to store the minimum-cost and the path associated with it during the search (lines 25-26). Then, it is checked if the current path has a cost that is already higher than the cost of the stored path at each iteration (line 17). By adopting this technique, it is possible to drastically reduce the number of unnecessary recursive calls, pruning a vast number of partial paths (and all the sub-trees starting from that current cell) that already present higher costs than the full current minimum path.

Algorithm 2 Energy-aware Grid-based Coverage Path Planning Algorithm

```
Input: Area composed of a set of vertices $\{v_1, ...v_p\}$
Output: Minimum path composed of a set of waypoints $\{w_1, ...w_p\}$
 1: $grid \leftarrow convertAreaToGrid(area)$
 2: $matrix \leftarrow floodingMatrix(grid)$
 3: $firstPoint \leftarrow getStartingPosition()$
 4: $prevCost \leftarrow 0$
 5: $path \leftarrow recursiveFunc(firstPoint, matrix, prevCost)$
 6: loop
 7:     $cell \leftarrow getLastCell(path)$
 8:     $tempMatrix \leftarrow computeTempMatrix(matrix, cell)$
 9:     $neighbors \leftarrow computeNeighbors(cell, tempMatrix)$
10:     if There is no neighbors then
11:         return
12:     end if
13:     for each neighbors(i) do
14:         $path \leftarrow path + neighbors(i)$
15:         $cost \leftarrow computeCost(path)$
16:         $cost \leftarrow cost + prevCost$
17:         if cost $>$ minimumCost then
18:             return
19:         end if
20:         $path \leftarrow recursiveFunc(path, tempMatrix, cost)$
21:         ~~$FiFo \leftarrow FiFo + path$~~
22:         $path \leftarrow path + firstPoint$
23:         $cost \leftarrow computeCost(path)$
24:         if cost $<$ minimumCost then
25:             $minimumCost \leftarrow cost$
26:             $minimumPath \leftarrow path$
27:         end if
28:     end for
29: end loop
30: ~~$minimumPath \leftarrow min(ComputeCost(FiFo))$~~
31: return minimumPath
```

4.2.2.2 Pruning Technique #2: Computing the cost introduced by new neighbors

The first pruning technique depends on the improvement provided by the second pruning. In order to prune the partial paths with current cost higher than the minimum-path, it is necessary to compute the cost of a path during the search phase, i.e., while nodes are being explored. In this way, there is no need to compute the cost of the path from the starting cell to the current one at each iteration, but only the additional cost introduced by choosing the next neighbor (line 15).

By applying the pruning technique to the algorithm with the original cost function based on the sum of angles, as shown in Figure 40a, it is necessary to consider only three cells A, B, and C. In the mentioned example, the cost of traveling the path from

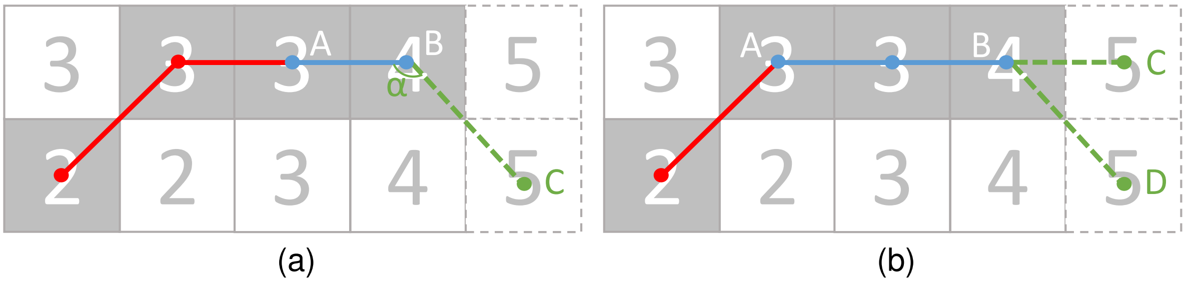


Figure 40 – Example of the algorithm adopting the pruning technique #2, where only the cost of a partial path is computed: (a) sum of angles cost function (b) energy-aware cost function

the first cell to A (red path) was already computed. The current cell is B, and the next chosen neighbor is the cell C. Thus, the updated cost to reach C is the sum of the previous cost and the new angle $\widehat{ABC}$.

On the other hand, to compute the partial cost of the path using the algorithm with the energy-aware cost function, as illustrated by Figure 40b, it is only necessary to consider the current straight line and the last performed turn. In the example, the current position is the cell B, and there are two potential neighbor cells, C and D, to be explored. By choosing the cell D, the energy of the partial path can be computed by Equation (14) as follows:

$$E_{partial} = E_{partial} + E_{turn}(\widehat{ABD}) + E_{line}(\overline{BD}) \tag{14}$$

where $E_{partial}$ is the previously computed energy to travel from the starting position to the current point B, $E_{turn}(\widehat{ABD})$ is the energy needed to perform a turn at B, and $E_{line}(\overline{BD})$ corresponds to the energy necessary to travel the straight distance from B to D, i.e., the first term of Equation (13) considering a single line. However, when choosing the cell C, the straight distance increases, and the energy of the partial path should be computed by Equation (15) as follows:

$$E_{partial} = E_{partial} - E_{line}(\overline{AB}) + E_{line}(\overline{AC}) \tag{15}$$

where $E_{line}(\overline{AB})$ is the previously computed energy to travel from A to B (blue line) and $E_{line}(\overline{AC})$ is the energy needed to travel from A to C. In this case, since we are going forward to the cell C, it is necessary to remove the energy computed considering the old distance ($\overline{AB}$) and add the energy of the new straight line ($\overline{AC}$).

The two proposed pruning techniques can be applied to the algorithm using the original and the energy-aware cost function. The techniques drastically reduce the computational time of the algorithm during the offline planning phase, making it time-affordable for real-world applications.

4.2.3 Discussion

The EG-CPP is an energy-aware grid-based coverage path algorithm able to deal with irregular-shaped areas containing no-fly zones. The algorithm performs a DLS over a grid-discretized scenario through approximate cellular decomposition. The area of interest is labeled by the Wavefront algorithm, which marks the adjacency neighborhood of the cells.

Our algorithm improves the grid-based method proposed by VALENTE et al. (2013). We replace the original cost function based on the sum of the angles, for a cost function based on the enhanced energy model presented in Section 4.1.3. Several works in the literature seek to minimize the number of turning maneuvers to minimize the total energy consumption indirectly. We intend to show that it is possible to generate more economical coverage paths using our novel energy-aware cost function. In this way, we consider different relevant aspects of the energy consumption of UAV, not only the number of turning maneuvers. Such aspects include vehicle dynamics, kinematic energy, and optimal speed.

Furthermore, we also present two modifications to prune a considerable number of partial paths and drastically reduce the computational time of the original algorithm. We assume that the coverage paths are split into a set of straight segments and turning maneuvers. Thus, the energy consumption is computed in each portion of the path separately. In this way, there is no loss in discarding partial paths whose costs are already higher than the minimum-cost path. Therefore, our solution is sub-optimal as the original algorithm. There is no trade-off between the saved computational time and the quality of the solution in terms of energy.

The EG-CPP presents the same limitation as to the E-Spiral since it explores the energy model to generate coverage solutions. One needs the initial data gathering to work with different vehicles and controllers. Besides, even with the pruning techniques, the algorithm takes a considerable amount of time to solve large and complex scenarios containing more than 50 cells, as will be stated in Section 5.2.

4.3 NC-Drone

The patrolling problem is a specific domain of the CPP problem dealing with monitoring and inspecting areas of interest. In such activities, a cooperative fleet must visit and revisit areas at regular intervals in order to supervise it (CHEVALEYRE, 2004). This surveillance task may include searching for intelligent targets (VINCENT; RUBIN, 2004) and monitoring climate changes (STALMAKOU, 2011). Different metrics can measure the patrolling performance, such as probability of detecting a target and tracking time (VINCENT; RUBIN, 2004), quadratic mean of the intervals and standard deviation of frequencies (SAMPAIO; SOUSA; ROCHA, 2016), number of complete coverage and number of turning maneuvers (CABREIRA et al., 2018b), information latency (ACEVEDO et al., 2013a), certainty and interest level (LIM; BANG, 2009, 2010), distribution and interval between visits (RAMASAMY; GHOSE, 2016). Some approaches deal with different local priorities (ARAUJO; SUJIT; SOUSA, 2013) and preferential surveillance (RAMASAMY; GHOSE, 2017). The performance metric depends on the primary goal of the task and must fulfill the application requirements.

Flight patterns based on simple geometric shapes, such as back-and-forth (BF) (MAZA; OLLERO, 2007; TORRES et al., 2016; JIAO et al., 2010; LI; WANG; SUN, 2016; DI FRANCO; BUTTAZZO, 2016) and spiral (SP) (CABREIRA et al., 2018; BALAMPANIS; MAZA; OLLERO, 2016, 2017c,b,a), are generally adopted for coverage tasks in regular-shaped areas. However, these flight patterns are not useful for patrolling missions and military surveillance aiming at monitoring intelligent targets. Such coverage approaches follow a pre-defined pattern which repeats the same behavior during the coverage. In this way, the object can avoid being tracked by heading to a different direction and staying out of sight of the UAV.

Pheromone-based methods interleave between path planning and path execution during the coverage and have been addressed to solve patrolling problems with land robots. Each robot, for instance, can use an RFID tag to read and write in the cells on the floor (NATTERO et al., 2014). When performing a motion, the robot must read the neighbor cells and count the number of visits on each one of them. Then, it chooses the least visited cell to move in. Finally, it updates the value of its current cell according to the adopted method. These methods have also been addressed in patrolling missions with aerial vehicles. Such methods can prevent intelligent targets from escaping the UAV since they are unpredictable to an external observer. However, the unpredictability demands a large number of changes in the vehicle's direction, increasing energy consumption in aerial platforms (CABREIRA et al., 2018b). Thus, it is necessary to minimize the number of turning maneuvers of such methods to obtain an energy-aware robust-to-failure solution.

Computational power represents a rigid constraint to any autonomous vehicle (KOENIG; LIU, 2001). Since pheromone-based methods have low computational costs, they have been recently applied in aerial coverage using UAVs (NATTERO et al., 2014; CABREIRA; FERREIRA, 2016; CABREIRA et al., 2018b). A Ground Control Station (GCS) is responsible for splitting the scenario into virtual cells. The GCS has a global map of the area of interest and coordinates the movement of the vehicles. The station monitors the positions of the UAVs and reads the neighbors cells, sending the corresponding coordinates to guide their motions. The main goal is to lead the UAVs to the less frequently explored areas. The GCS is also in charge of avoiding crashes between the aircrafts (ZELENKA; KASANICKÝ, 2014; ZELENKA; KASANICKY, 2014a). However, this centralized decision-making process requires massive communication among the UAVs and the station. Each action of a vehicle depends on the GCS. It is necessary to periodically update the station about the current status of the fleet. Interference or failure in the communication may compromise the mission, resulting in inefficient coverage. Loss of a UAV, a collision between vehicles or a crash in an inhabited place are some of the problems which may also occur. This fashion of decision-making process imposes several limitations to its application in real-world scenarios.

4.3.1 Pheromone-based Methods

Pheromone-based methods are suitable for missions where UAVs contain incomplete information about a partially known environment. These methods comprise algorithms based on colonies of insects, specifically ants, as their natural behavior consists of leaving traces of pheromones while exploring areas outside the anthill (FLOREANO; MATTIUSSI, 2008). The pheromones are useful to guide the swarm during the forage for food. When an ant finds food, it returns to the anthill using the same path and reinforces the pheromone traces previously left along the way. Stronger traces of pheromones attract other ants to the prosperous area, which explains the success of the colonies.

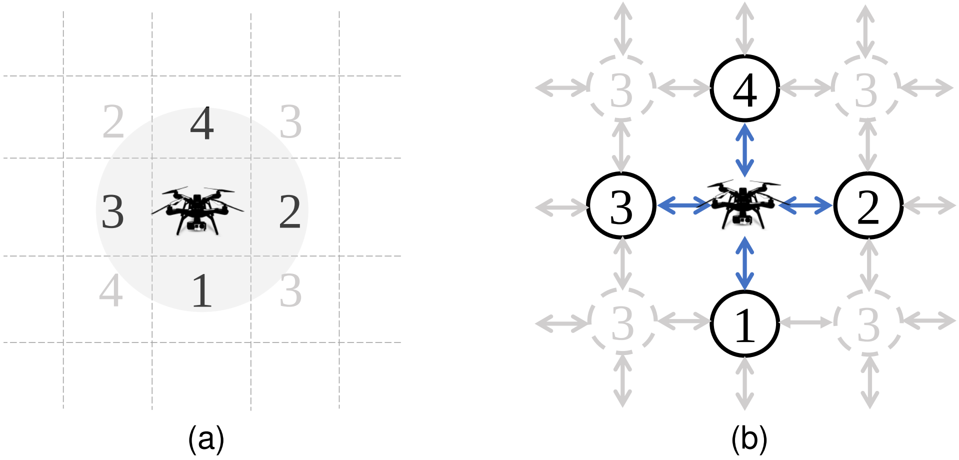


Figure 41 – The scenario is discretized into a regular grid using the approximate cellular decomposition. (a) Field of view, (b) Connected graph.

These methods mimic the ants' pheromones by applying virtual marks on a grid-discretized scenario. The area of interest is decomposed into a set of cells using the approximate cellular decomposition (CHOSET, 2001). The size of the cells depends on the resolution of the onboard camera of the UAV and the flight altitude adopted during the patrolling task. The higher the altitude, the larger the footprint of the UAV and the size of each cell, and the lower the quality of the images. The balance between these elements depends on the application requirements. The UAV's field of view for reading and writing in the cells is restricted to the von Neumann neighborhood, which includes the current cell and the for adjacent neighbor cells, as illustrated in Figure 41a. The pheromone-based methods usually only consider the 4-neighborhood at the same

Manhattan distance of 1 to preserve the low computational cost. In this way, these simple heuristics do not take into consideration the different costs to move from one cell to another, but only the value of the cell itself. A connected graph represents the adjacent cells explored for searching and displacement in Figure 41b. The coverage is performed by adopting local search and exploring only the nearest neighborhood area.

While moving around the area of interest, UAVs leave virtual marks in the visited cells. Each cell works with an associated value, called *u-value*, to count the total number of visits performed by UAVs. Thus, the area of interest acts as an indirect communication channel. The *u-value* information is useful to help other vehicles in the next destiny decision-making. To obtain full coverage, UAVs should move to unvisited or less frequently visited locations, which means exploring cells with the least amount of pheromone. At each motion, UAVs read the *u-value* of neighbor cells, choose the one with the minor value, and update the current cell according to a specific rule. As time passes by, uncertainty about a given location increases since it is a dynamic scenario (LIM; BANG, 2009). The degradation of the virtual marks over time emulates this behavior and represents the natural process of evaporation of pheromones (ZELENKA; KASANICKY, 2014b). This feature allows the locations to be covered several times during the coverage. Furthermore, different types of pheromones can be used to guide the vehicles, such as attraction and repulsion pheromones (ALBANI; NARDI; TRIANNI, 2017).

Table 2 – Update rules of the main real-time search methods

Method	**Update rule**
Node Counting	*u(s) := u(s) + 1*
Learning Real-time A*	*u(s) := u(next(s)) + 1*
Thrun's Value-Update Rule	*u(s) := max(u(s), u(next(s)) + 1*
Wagner's Value-Update Rule	*if u(s) $\leq$ u(next(s)) then u(s) := u(s) + 1*

Real-time Search Methods (RTSM) are pheromone-based approaches which explore a variety of update rules to change the amount of pheromone in the visited places, as presented in Table 2. The four heuristics Node Counting (NC), Learning Real-Time A* (LRTA*), Thrun's Value-Update Rule (TVUR), and Wagner's Value-Update Rule (WVUR) were originally compared by KOENIG; LIU (2001) in simulations with ant-robots for land coverage. Recent studies also compared, extended, and applied those heuristics in simulations and real experiments using UAVs (NATTERO et al., 2014; CABREIRA; FERREIRA, 2016; CABREIRA et al., 2018b).

The NC consists of a heuristic using a basic rule to increase the *u-value* of the visited cells. While performing the coverage, the algorithm verifies the *u-value* of the neighbor cells and selects the one with the lowest value - at the beginning, all cells contain zero. Thus, it updates the current cell by adding one unit to the original value *u(s)*. Finally, it moves the vehicle to the selected cell. All vehicles repeat this process at each iteration while moving around the scenario. The algorithm keeps running until a stop criterion, such as the minimum number of complete coverage or the predefined number of cycles. Algorithm 3 presents the Node Counting.

Algorithm 3 Node Counting

```
Input: A set of cells $\{c_1, ...c_p\}$
Output: A set of metric values $\{m_1, ...m_p\}$
repeat
    Read the u-value of the neighbor cells
    Select the cell with lowest u-value
    Update with +1 the u-value of the current cell
    Move to the selected neighbor
until Number of cycles is reached
```

LRTA* heuristic adopts a different rule to update the *u-value* of the current cell. After reading the neighbor cells and choosing the next location to move, the algorithm changes the value of the current cell *u(s)* by the value of the selected cell *u(next(s))* added by one unit. TVUR heuristic compares the value of the current cell *u(s)* with the value of the next cell to be visited *u(next(s))*. The highest value is increased by one unit and replaces the original value of the current cell *u(s)*. WVUR heuristic presents a conditional update rule where the value of the current cell *u(s)* is increased by one unit if it is not larger than the value of the next cell *u(next(s))*. Otherwise, the value of the current cell is not updated.

The next subsections present the proposed NC-Drone algorithm, which is an extension of the Node Counting (NC) algorithm for patrolling missions using multiple UAVs. This extension reduces the massive number of turning maneuvers performed by the UAVs running the NC algorithm to save energy. At the same time, the NC-Drone keeps the unpredictable behavior for an external observer, which is a fundamental aspect of patrolling missions. To achieve this goal, we developed two types of NC-Drone, a centralized algorithm and a decentralized one with a few variations. These variations use internal matrices as the representation of the global map and employ synchronization schemes to merge and combine the data stored in the matrices.

We also propose a set of cooperative strategies, further exploring relevant aspects of the patrolling problem, such as time, uncertainty, and communication. First, we introduce the Watershed Strategy (WS), a technique used to represent matrices as topographic relief. In this relief, the lower elevations correspond to the minor values in the matrix, i.e., less frequently explored cells. The UAVs are attracted to clusters of cells formed in these needy areas. Second, we present the Time-based Strategy (TS), an approach exploring not only the number of visits but also the time when the last visit occurs to guide the UAVs during coverage. Both strategies can be combined. Then, we propose the Evaporation Strategy (ES) to model uncertainty due to the absence of visits in certain places during patrolling. We also explore the concept of full-range communication with Communication-Frequency Strategy (CFS), where UAVs reduce communication, exchanging matrix-information from time to time. We are interested in studying the impact of communication in the patrolling performance in terms of the spatial and temporal distribution of visits and the trade-off between communication and performance. Finally, we combine all strategies into a single solution for the patrolling problem.

4.3.2 Centralized NC-Drone

Real-time Search Methods (RTSM) are usually 1-range strategies capable of sensing the closest surrounding cells. During the coverage, these methods always choose the locations with the smallest amount of pheromone (number of visits) as the next place to be visited by the UAV. When there is more than one cell with the same minimum-value, the algorithm randomly chooses among one of the cells. All cells have zero at the beginning of the mission, which causes several ties and continuously changes in the orientation of the UAVs. As stated by DI FRANCO; BUTTAZZO (2016), the higher the number of turning maneuvers, the higher the energy consumption.

Centralized NC-Drone works precisely like the original Node Counting in terms of sensing neighbor cells and choosing the least visited one. However, this extension adopts a simplified version of the energy model to consider the energy as a decision factor. The algorithm can identify when there is a tie between two or more cells with the minimum-value. It verifies if one of these cells is aligned with the sweeping direction of the UAV. In this case, the correspondent cell is selected as the next place to be visited, keeping the UAV in the same direction and avoiding unnecessary turns. In this way, we can prolong the straight segments of the path and minimize the number of turning maneuvers to reduce energy consumption. If there is no aligned cell, the algorithm randomly takes the decision and sustains the unpredictable behavior. Algorithm 4 describes Centralized NC-Drone.

4.3.3 Decentralized NC-Drone

Decentralized NC-Drone explores a model where the UAVs have a matrix-form internal map of the area of interest to be covered. The matrix representation is straightforward since the area is decomposed into a regular grid using approximate cellular decomposition. A pair of coordinates (line and column) denotes a cell and keeps the number of visits performed in the corresponding location. Each UAV stores the visited places in its internal matrix and selects the next destinations based on the number of

Algorithm 4 Centralized NC-Drone

```
Input: A set of cells $\{c_1, ...c_p\}$
Output: A set of metric values $\{m_1, ...m_p\}$
repeat
    Read the u-value of the neighbor cells
    Select the neighbor cell with lowest u-value
    if there is a tie and one of the cells is aligned with the sweep direction then
        Choose the aligned cell as the next place to be visited
    else
        Randomly choose one of the cells
    end if
    Update with +1 the u-value of the current cell
    Move to the selected neighbor cell
until Number of cycles is reached
```

visits stored in the neighbor positions of its current position in the matrix. The vehicle always chooses the minor value and respects the tie rule implemented in the centralized version to maintain the sweep direction.

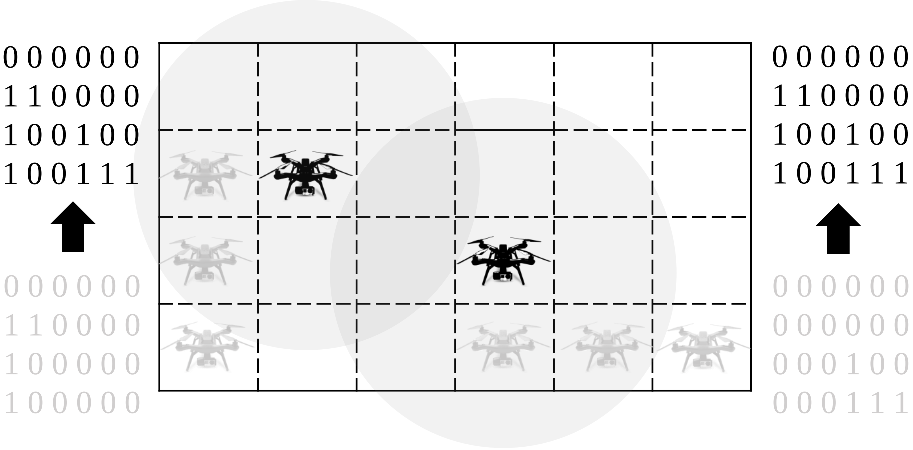


Figure 42 – Two UAVs are performing the patrolling in the scenario. When the vehicles are within the communication radius, the synchronization process starts to exchange the information stored in the current matrices (in grey). The information about the visited places of both matrices is updated (in black).

A matrix-based communication model can be employed to share and synchronize individual information stored in the internal matrices of the UAVs. In the simulation, the UAVs can verify the existence of other vehicles in the surrounding area, establishing a monitoring perimeter. In real flights, the vehicles can communicate with each other through wireless messages using XBee devices. These long-range devices can also communicate with microcontroller boards via a serial interface to process the exchanged information. Figure 42 illustrates the matrix-based communication model with two UAVs performing the patrolling. The UAVs detect each other presence within the communication radius and open a communication channel to trade information about the visited places and update their internal matrices. Algorithm 5 describes the Decentralized NC-Drone adopting the matrix-based communication model.

Algorithm 5 Decentralized NC-Drone

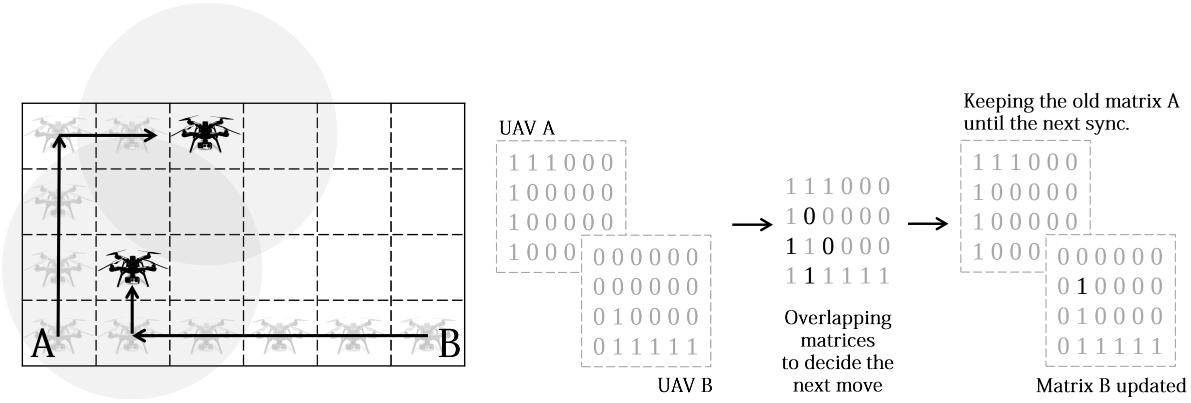
```
Input: A set of cells $\{c_1, ...c_p\}$, matrix
Output: A set of metric values $\{m_1, ...m_p\}$, matrix updated
repeat
    Read the u-value of the neighbor cells in the matrix
    Select the neighbor cell with lowest u-value
    if there is a tie and one of the cells is aligned with the sweep direction then
        Choose the aligned cell as the next place to be visited
    else
        Randomly choose one of the cells
    end if
    Update with +1 the u-value of the cell in the matrix
    Move to the selected neighbor cell
    if sync is on and there are UAVs within the radius then
        Open a communication channel
        Merge or combine the information of the matrices
        Update the matrices with the resulting values
        Close the communication channel
    end if
until Number of cycles is reached
```

The matrix-based communication model allows the UAVs to exchange information stored in the internal matrices. Synchronization methods can be used to merge information from two different matrices into a single one or combine multiple matrices to decide the next place to go. The proposed merging methods are NC-Drone MAX and NC-Drone AVG. The former one compares every position in both matrices and chooses the highest value, while the latter one calculates the mean between the original values in each position of both matrices and round the result to the nearest integer number. Both methods update the original matrix. Table 3 presents a brief description of the synchronization methods.

Table 3 – Synchronization methods for merging and combining matrix information.

Method	**Description**
NC-Drone MAX	Updates the matrix with the major value.
NC-Drone AVG	Computes the mean between two values and rounds the result to the nearest integer number.
NC-Drone MULTI	Superposes the matrices, but does not merge the information. UAVs can copy the matrices of other vehicles to its memory.

The NC-Drone MULTI combines multiple matrices when deciding each move. Using this approach, the UAVs copy the information from each other when they are within the synchronization perimeter. They can store multiple matrices from all vehicles in the scenario. When performing a move, the UAV superposes the stored content instead of fusing the information, summing the neighbor values of its current position. Then, it chooses the least visited location as the next place to be covered. UAV updates only its matrix during coverage. Other matrices internally stored are individually updated when there is a new synchronization. Figure 43 illustrates the NC-Drone MULTI.



(a) (b)

Figure 43 – Decentralized NC-Drone MULTI. (a) UAV A starts at the lower left corner moving forward and to the right, while the UAV B starts at the lower right corner moving to the left and forward, (b) When the UAVs are within the communication radius (current position in (a)), they can copy the matrix from each other and combine the information to make a decision.

4.3.4 Watershed Strategy

The Watershed is a mathematical morphology model widely used for segmentation in image processing field (BEUCHER; MEYER, 1992). The algorithm can also be used in patrolling missions involving UAVs to represent a matrix as a topographic relief with different elevations. Depression zones correspond to the minor values in the matrix, i.e., poorly or less frequently explored areas, while higher elevations represent the greater values. A global analysis of the map instead of using only local information (neighbor cells) may be useful to guide the vehicles to the needy areas, balancing the coverage.

Watershed Strategy (WS) analyzes internal matrices and computes the average value of its elements. This value is rounded down and used as a threshold by the WS, as indicated in line 1 of Algorithm 6. A $clusterList$ is created to store all the resulting clusters of the matrix (line 2). Then, the algorithm chooses a random position in the matrix M whose value is under the $threshold$ (line 3). This random cell is added to a new $cluster$ (line 5), which is a structure used to store all cells belonging to the same group. The same cell cannot be part of two different clusters. A flooding process starts

5	5	5	5	5	5	5	5	5	5	5	6
5	4	4	4	5	7	7	7	3	4	3	6
3	3	3	4	5	5	6	7	3	3	5	5
3	3	3	3	5	5	5	5	5	3	5	5
7	7	7	3	5	5	5	5	4	3	4	5
7	7	7	6	6	6	6	6	5	5	6	6

Figure 44 – Watershed Strategy forms two clusters with a group of cells whose u-value is beneath the $threshold$ = 5. The process starts in a random cell with u-value under the $threshold$ and spreads to the neighbors until there is no cell outside a cluster.

in the selected location, spreading to the neighboring cells (Moore neighborhood) whose values are beneath the $threshold$ (line 7 and 8). These neighbors are added to the current $cluster$ (line 9) and the propagation process repeats until there is no surrounding cell under the $threshold$. The resulting group of cells forms a $cluster$, which is added to a list of all clusters (line 12). This process forms several clusters and ends when there is no cell in the matrix under the threshold, as illustrated by Fig. 44.

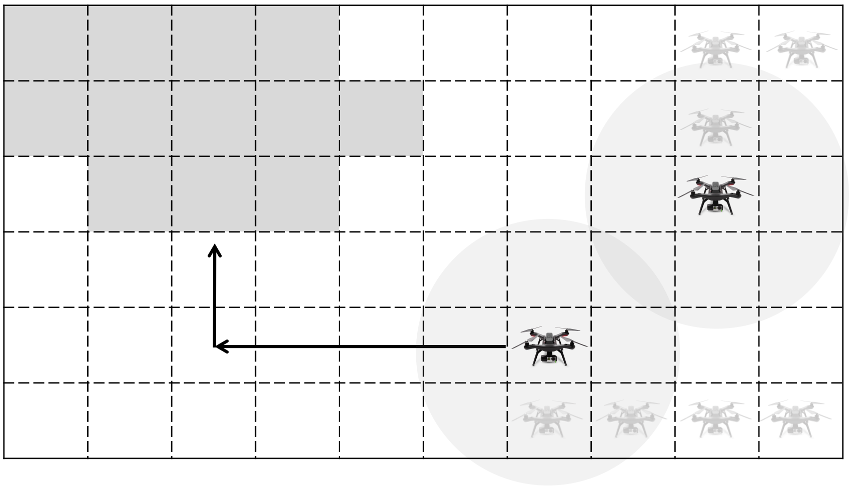


Figure 45 – Watershed Strategy computes the cluster (grey zone) when the two UAVs are within the communication radius. After synchronizing their matrices, the designated UAV follows a straight path with a turn to cover the selected zone.

The WS verifies the resulting clusters and selects the one with the larger number of cells (line 14). Then, the algorithm selects a random cell inside this cluster as the next location to be visited (line 15). The first UAV to finish the WS execution covers the cluster and sends a message to the other vehicles to prevent them from covering the same zone. A straight path with minimum turns is used by the UAV to reach the cluster zone. As can be seen in Fig. 45, the UAV in the bottom part of the scenario computes the cluster and goes toward it following a simple trajectory with a 90-degree turn. The

Algorithm 6 Watershed Strategy

```
Input: A matrix $M_{ij}$
Output: A destiny cell $c$
$threshold \leftarrow Floor(Threshold(M))$
Create empty $clusterList$
while cell $c \leftarrow$ Random ((cell $c$ in $M$ with u-value $< threshold$) and
    ($c \notin clusterList$)) do
    Create new $cluster$
    Insert $c$ in $cluster$
    for all cell $c$ in $cluster$ do
        Propagate $c$ to the $neighbors$
        for all (($neighbors$ of $c$ with u-value $< threshold$) and
            ($neighbors$ of $c \notin cluster$) and
            ($neighbors$ of $c \notin clusterList$)) do
            Add $neighbor$ to $cluster$
        end for
    end for
    Add $cluster$ to $clusterList$
end while
Choose the largest $cluster$ in $clusterList$
Returns a random cell $c$ in $cluster$
```

other UAV remains in the surrounding area visiting the closest neighbors. Once the UAV reaches the selected location, the tendency is to remain inside the cluster exploring the whole area.

4.3.5 Time-based Strategy

The previously proposed approaches use the number of visits to choose the next place to be visited by UAVs. When there are two or more cells with the same minimum value, the UAV chooses the one aligned with the sweeping direction. If there is no aligned cell, the vehicle randomly chooses one of the others. The Time-based Strategy (TS) considers the time when the last visit occurs to decide among the ties during the patrolling, selecting the cell visited less recently.

Time-based Strategy (TS) can also be explored combined with the Watershed Strategy (WS). Instead of using a threshold based on the average of visits to build the clusters, the approach can employ the average time of the last visit performed in all cells as a threshold. This strategy forms clusters with a set of cells covered less recently to guide the vehicles. Table 4 presents a summary of the time-based strategies.

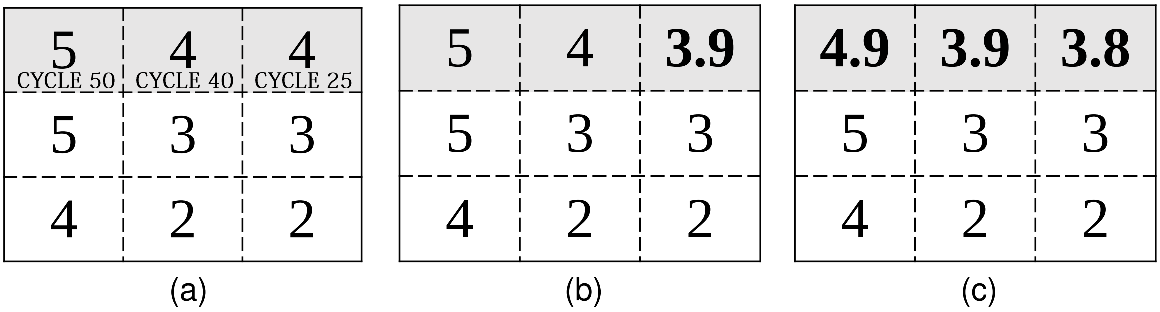
4.3.6 Evaporation Strategy

The uncertainty about information arises and increases over time due to the absence of visits in the scenario's locations, i.e., during the interval between two consecutive visits in a certain place. The Evaporation Strategy (ES) models the absence of visits by

Table 4 – Time-based strategies for NC-Drone.

Method	Description
NC-Drone TS	Selects the cell less recently visited when two or more cells have the same minimum value. It can be used with Centralized NC-Drone and Decentralized NC-Drone MAX, AVG, and MULTI.
NC-Drone WSTS	Combines the Watershed Strategy with the Time-based Strategy. It forms clusters with a set of cells less recently covered by adopting the average time of the last visit performed in all cells as a threshold.

reducing the amount of pheromone when a place is not visited for a while. This strategy works with an evaporation factor (EF) varying from $EF = 0.1$ to $EF = 1$ in steps of 0.1. By reducing the pheromone, the less recently visited areas become more attractive to the UAVs.



5 CYCLE 50	4 CYCLE 40	4 CYCLE 25
5	3	3
4	2	2

(a)

5	4	**3.9**
5	3	3
4	2	2

(b)

4.9	**3.9**	**3.8**
5	3	3
4	2	2

(c)

Figure 46 – Evaporation Strategy in three distinct cycles of simulation: (a) Cycle of simulation = 100, (b) Cycle of simulation = 125, (c) Cycle of simulation = 225.

The ES reduces the values stored in the matrix depending on the time of the last visit. Figure 46 illustrates the ES in three distinct cycles of simulation in a scenario represented by a matrix 3 x 3. Here we are considering only the first line highlighted in the matrix to exemplify. The last visit in each one of these places occurred at cycles 50, 40, and 25, respectively. The number of visits is 5, 4, and 4 in each position. These values are consistent in the current cycle of simulation = 100 in Figure 46a. In cycles 125 and 225, the third value is decreased by 0.1 two times, according to Figure 46b and Figure 46c. In the meantime, the first two values were also decreased at cycles 140 and 150, respectively. This example considers the pheromone dropping 0.1 at every 100 cycles in each unvisited location. The ES can explore different intervals to evaporate the pheromones, such as 50, 100, 250, 500, and 1k.

4.3.7 Communication-Frequency Strategy

The previously proposed approaches consider that the UAVs work with short-range communication to synchronize and exchange information. However, new technologies have been developed with increasing reach, such as XBee 3. This device works with different operating frequencies and adopts a wide variety of protocols, such as Zigbee, Thread, 802.15.4, DigiMesh, Wi-Fi, and Bluetooth. Moreover, it eliminates the need for an external microcontroller to process information. The range radius varies from 4000ft (1200m) to 2 miles (3200m) depending on the version, standard or PRO (DIGI, 2019a). This technology has been successfully integrated with UAVs, such as the Draganfly (DIGI, 2019b). The Communication-Frequency Strategy (CFS) explores the idea of a global range, where all UAVs are within the communication radius and can synchronize their matrices. The vehicles may explore different communication frequencies, reducing the number of exchanged messages compared with the centralized approach. Each UAV stores two matrices. The matrix A indicates the last synchronization, while the matrix B is a copy which is continuously updated when the vehicle moves. All vehicles compute the difference between the matrix B and A at every synchronization. The resulting matrix C corresponds to the visits performed by the UAV. The vehicles broadcast their resulting matrices, updating matrix A with the sum of all matrices C. These matrices correspond to the visits performed by the other UAVs. Then, matrix A is copied to matrix B, and the UAVs continue to count the visits in matrix B until the next synchronization.

4.3.8 Discussion

The NC-Drone is an energy-aware pheromone-based coverage path planning algorithm specially designed for the patrolling problem, which is a subtopic of CPP. The problem is related to monitoring and surveillance applications, where UAVs must visit and revisit every location of a grid-discretized scenario several times. We are seeking solutions with a uniform distribution of visits, both spatial and temporal, while keeping motion behavior unpredictable for an external observer or an intelligent target.

The algorithm explores a pheromone-scheme to represent the number of visits performed by all vehicles in each cell. The UAVs perform online coverage, interleaving between planning and execution. They decide the next location be visited based exclusively on the amount of pheromone. Thus, we adopted a simplified version of the energy model to introduce energy as a decision factor. NC-Drone modifies the original Node Counting heuristic when deciding between two cells with the same minimum-value. Our algorithm checks if one of these minimum-value cells maintains the UAV in the sweeping direction. If this is the case, the algorithm selects this cell as the next location; otherwise, it randomly chooses one of the other cells. In this way, we can prolong the straight segments of the path and minimize the number of turning maneuvers to reduce energy consumption.

A decentralized version of the NC-Drone adopts a matrix-based communication model, where each UAV stores internal matrices representing the area and computes the visits performed by itself. The vehicles can exchange information when they are within a synchronization radius, combining matrix data. Our goal is to replace the centralized scheme based on the ground control station (GCS). This system demands communication extensively, and it is not robust to failures, once every action of the UAV depends on the GCS.

Furthermore, we also propose other patrolling strategies, such as Watershed (WS), Time-based (TS), Evaporation (ES), and Communication-Frequency (CFS). These strategies present additional features and explore further information to guide the UAVs to the least visited areas and balance the patrolling. The WS uses the information of the entire matrix to form clusters of cells whose values are under a threshold. This strategy attracts the UAVs to those regions presenting low visitation. The TS helps to solve situations where two or more cells have the same minimum-value, choosing the cell visited less recently. The ES decreases the cells' values, raising the uncertainty of unvisited places. This strategy also attracts UAVs to locations that are unvisited for a while. The CFS explores the relationship between communication and performance. By adopting these strategies, we intend to improve the performance of the NC-Drone algorithm considering specific application requirements related to the uniform distribution of visits. As will be stated in Section 5.3, there is a trade-off between these application requirements and energy consumption.

The NC-Drone presents limitations considering the simulation environment. Our simulations do not consider issues as signal interference or loss of packages during the communication. Besides, we do not perform real flights to evaluate the applicability of this pheromone-based approach in real-world scenarios.

5 EXPERIMENTAL RESULTS

This chapter presents the experimental results obtained from simulations and real flights performed over regular and irregular-shaped areas of interest. Section 5.1 presents the comparison between the E-Spiral and the E-BF, an energy-aware back-and-forth approach proposed by DI FRANCO; BUTTAZZO (2016). The experiments consist of a wide range of simulations performed in 3750 polygonal areas of interest. Such areas present different characteristics, such as the number of vertices, the irregularity, and the size of the area. Real flights were also performed with a quadrotor with both algorithms in two different scenarios to validate the proposed approach and compare the energy spent during the missions with the one estimated by the energy model. Results showed that E-Spiral outperforms E-BF in all the cases, providing an effective energy saving even in the worst scenario (large areas and few vertices) with a percentage improvement of 10.37% up to the best case (high number of vertices) with 16.1% of improvement.

Next, Section 5.2 presents the comparison between the EG-CPP and the grid-based approach proposed by VALENTE et al. (2013). Real flights have been performed to validate the proposed approach and verify the accuracy of the energy model. In particular, the energy-cost function allowed saving 17% of the energy in comparison with the paths generated by the previous cost function. Additional simulation experiments were also performed in different irregular-shaped scenarios varying size and complexity. As a further contribution, we also apply two pruning techniques to the original algorithm, drastically reducing the computation time up to 99%.

Finally, Section 5.3 presents the comparison between the centralized NC-Drone and the four original RTSM. The decentralized variations are compared among each other to analyze the performance of the synchronization methods in the matrix-based communication protocol. Additional experiments show the improvements of the cooperative strategies (Watershed, Time-based, Evaporation, and Communication-Frequency) over centralized and decentralized NC-Drone. Finally, experiments combining all strategies are presented.

5.1 E-Spiral vs E-BF in Regular-shaped Areas

The proposed Energy-aware Spiral CPP (E-Spiral) algorithm described in Section 4.1 is compared to the E-BF algorithm proposed by DI FRANCO; BUTTAZZO (2016). Simulations performed over a broad set of regular-shaped areas of interest state the effectiveness of the proposed approach. Real flight results validate the simulation experiments and attest to the high accuracy of the improved energy model. Predicting the cost of a path is crucial to avoid unexpected crashes and make efficient use of the battery capacity.

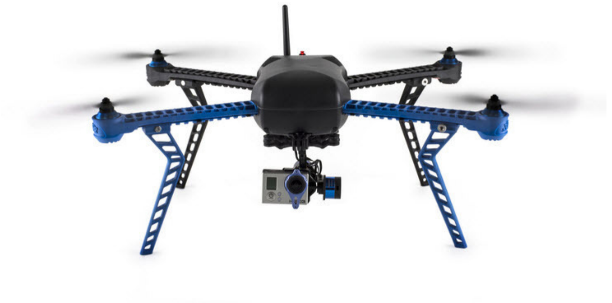


Figure 47 – IRIS quadcopter with a GoPro camera mounted on a Gimbal stabilizer.

An IRIS quadrotor executed the real flights with a GoPro camera mounted on a Gimbal stabilizer, as illustrated in Figure 47. The quadrotor weighs about 1.3 Kg and carries a LIPO 3S battery, while the autopilot is an Arducopter 3.2 on top of a PixHawk board. For each flight, the autopilot saves a log with all the useful information to analyze the experiment, such as GPS, speed, altitude, voltage, and current. The next subsections present the experimental results of the simulations and the real flights. The files used in the experiments are available on GitHub in `https://github.com/tauacabreira/ESpiral`.

5.1.1 MATLAB Simulations

A wide range of simulations was performed on MATLAB®. We explore a set of polygonal areas of interest with different characteristics, such as the number of vertices, the level of irregularity, and the size of the area. The number of vertices varies from 6 to 10. The level of irregularity varies from 0 to 1 and indicates the variance in the angular spacing of vertices. The average diameter varies from 200 to 600 meters and determines the size of the area. We generated fifty different areas using a MATLAB® function with all possible configurations, totalizing 3750 tested areas.

We ran the E-Spiral and the E-BF algorithms to generate the coverage paths for each area. Using a look-up table, we set the optimal speeds that minimize the amount of energy for every straight segment of the path. Then, we used the energy model improved in Section 4.1.3 to compute the energy spent by each approach.

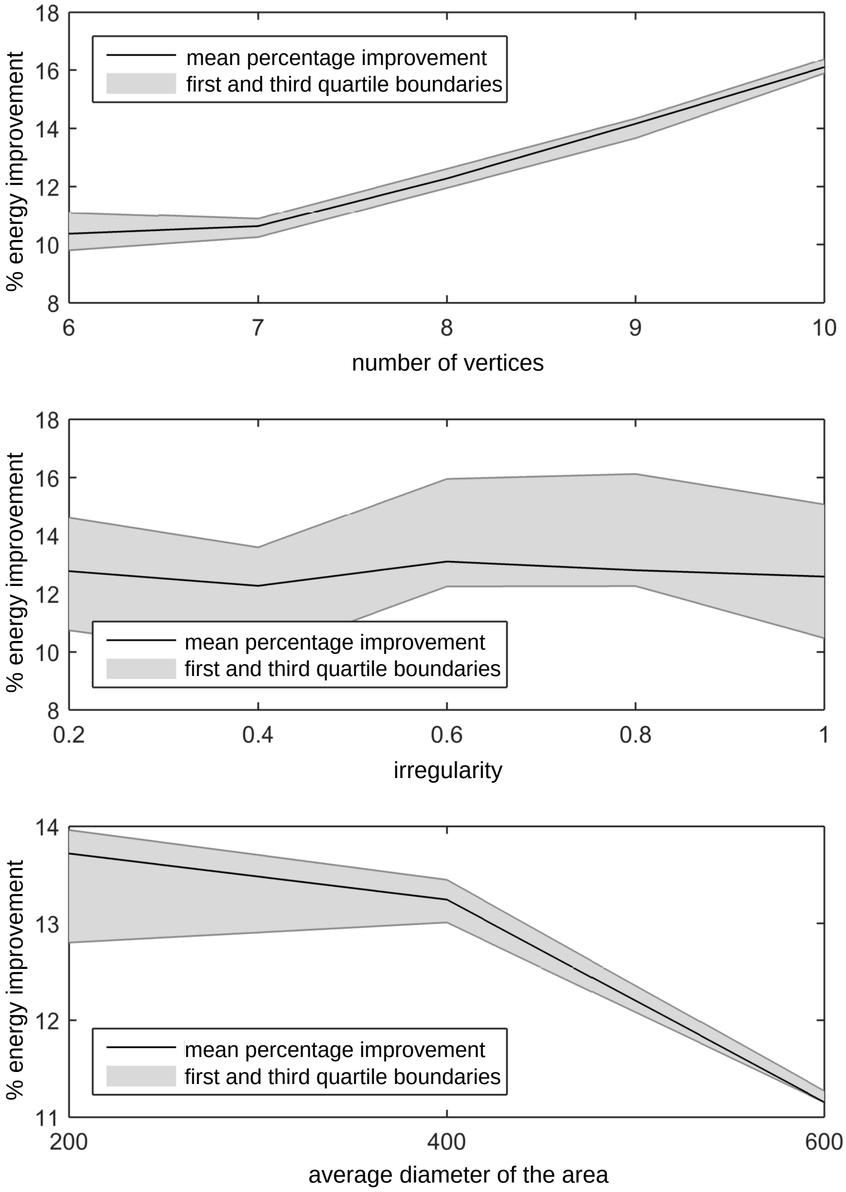


Figure 48 – Percentage of energy improvement of the E-Spiral over the E-BF varying: (a) the number of vertices, (b) level of irregularity, and (c) the size of the areas of interest. In all the cases E-Spiral outperforms E-BF.

The percentage of energy improvement of the E-Spiral over the E-BF algorithm is presented in Figure 48. The higher the number of vertices, the better is the performance of the E-Spiral, as illustrated in Figure 48a. The improvement reaches 16.1% in areas containing ten vertices.

E-Spiral takes advantage of smoother turning maneuvers in polygonal areas. The UAVs tend to keep the optimal speed for more extended periods in rounded-shaped areas containing wider angles, which minimizes the total energy consumption of the path. The levels of irregularity present a more significant difference between the first and the third quartiles. However, considering the mean value, the average percentage of improvement was around 13%. The difference between both algorithms remains constant from 0.2 to 1, as shown in Figure 48b. Finally, the performance of E-Spiral decreases as the area increases, but still overcomes the E-BF around 10%, as shown in Figure 48c. This performance drop is expected since the effect of wider turning maneuvers is less relevant in large areas. In such scenarios, the E-BF benefits from long straight distances, also keeping the optimal speed for prolonged periods.

5.1.2 Real Flights

The real flight experiments were performed in two different areas of interest, a polygon and a rectangle. These areas have been chosen to highlight the differences between both algorithms. We designed the areas using the waypoints as vertices in the Mission Planner software (OBORNE, 2017). Then, we import the waypoint file containing all points forming the area to MATLAB®. Next, we convert the Latitude/Longitude to Cartesian coordinates. We run both algorithms to generate the coverage paths in the resulting areas. In the end, a waypoint file containing all information about the coverage path is created. Such information includes the codes for specific commands (take-off, change speed, move to a waypoint), the latitude/longitude coordinates for each waypoint, the altitude, and the adopted speed. We transfer the waypoint file to the internal memory of the UAV and run a pre-programmed flight using the coverage paths generated by the algorithms.

In particular, the E-Spiral may be more effective when used in a polygon area while the E-BF should benefit from a rectangular area. Moreover, while the improved energy model will behave similarly to the original in the rectangular area due to 90° turns, a difference between the two should be evident when estimating the energy consumption of the polygon area paths. Figure 49 illustrates the areas of interest (red), the planned path (blue), the performed path during the real flights (white), the starting position (green “x"), and the final position (red “x").

The experimental results for the mission execution time and energy consumption can be seen in Figure 50a and Figure 50b, respectively. The green bars represent the real flights performed with the trajectories generated by E-Spiral and E-BF algorithms. The red and the yellow bars indicate the estimated values by the original and the improved version of the energy model, respectively. E-Spiral algorithm overcomes E-BF in both areas of interest. It reduces the mission execution time around 9% and the energy consumption around 7.7% in comparison with E-BF. The experimental results state the effectiveness of the proposed approach. Table 5 and Table 6 present detailed information about the time and the energy estimated by the original (OEM) and the improved energy model (IEM), along with the real flight results and accuracy rate.

The real flight results also stated the estimation performance improvement provided by the IEM over the OEM. The improved energy model increases the time and energy

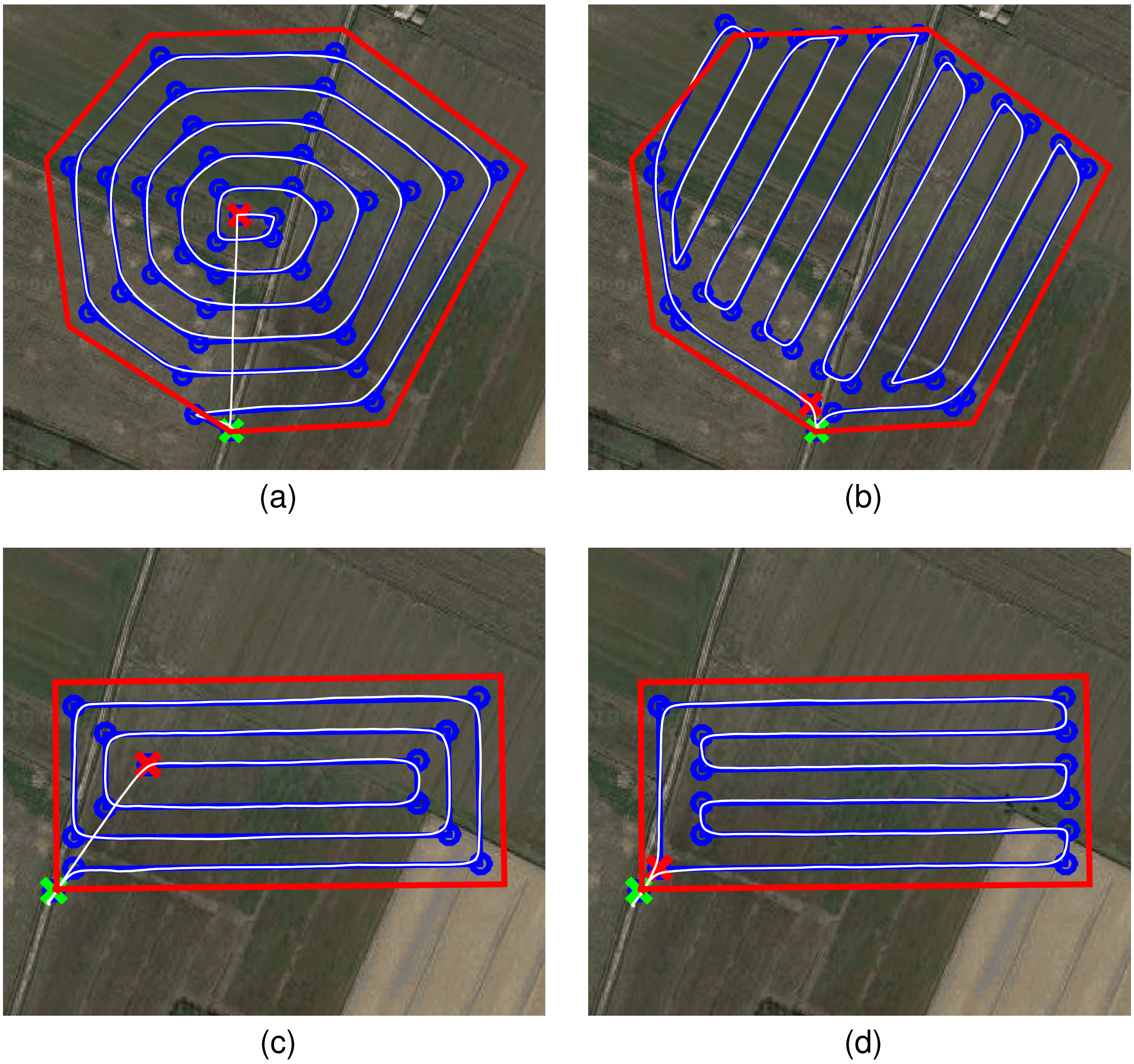


Figure 49 – Real flight paths generated with energy-aware algorithms: (a) E-Spiral algorithm in polygonal area, (b) E-BF algorithm in polygonal area, (c) E-Spiral algorithm in rectangular area, and (d) E-BF algorithm in rectangular area.

estimation precision in 13.24% and 13.41%, respectively. The OEM presented imprecise results in the polygonal area. It overestimates the mission execution time, causing errors in the energy consumption estimation of the E-Spiral. The imprecision is because of zero speed considered at the beginning and end of each straight segment of the path. The OEM considers that the UAV completely decelerates to perform every turning maneuver, regardless of the angle. After turning, the UAV accelerates from zero to the optimal speed. On the other hand, the IEM considers only a percentage of reduction in the speed at every turn. Using a function based on the entrance speed and the turning angle, the improved energy model can compute the resulting speed after the curve.

In this way, the IEM can be employed as an estimation energy tool for offline coverage path planning, since it presents a high accuracy rate in terms of time and energy. The accuracy between the simulation and the real flights regarding the mission execution

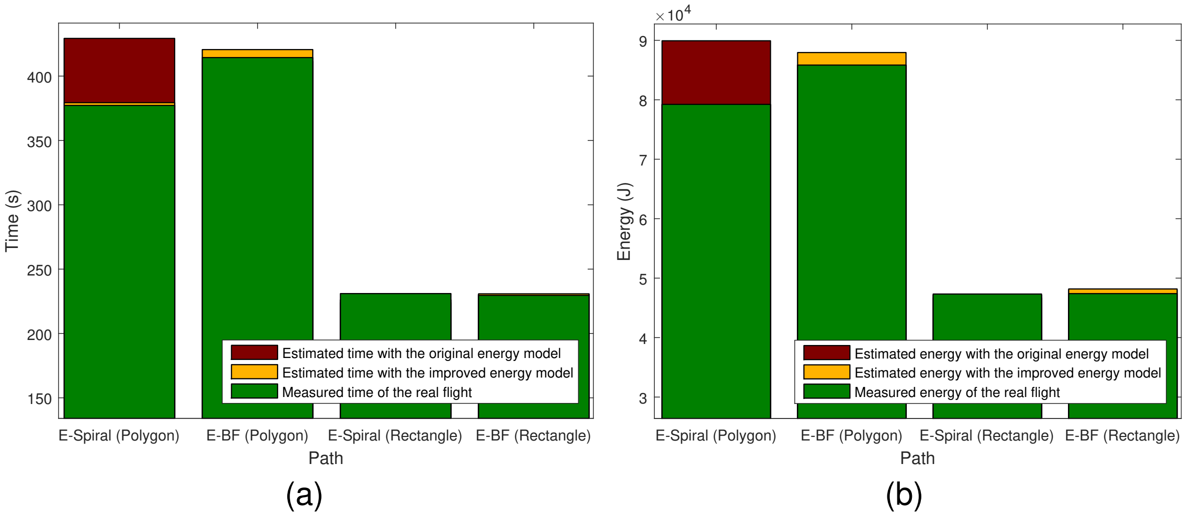


Figure 50 – Simulation and real flights results with E-Spiral and E-BF algorithms in rectangular and polygonal areas: (a) The accuracy of the mission time estimation varies from 96.93% to 99.47%, (b) The accuracy of the energy estimation varies from 97.54% to 99.91%.

Table 5 – Mission execution time in simulation and real flights with the E-Spiral and the E-BF in Polygonal (P) and Rectangular (R) Areas

Path/Area	**OEM**	**IEM**	**Real Flight**	**Accuracy**
E-Spiral (P)	$429.29s$	$379.35s$	$377.20s$	99.43%
E-BF (P)	$411.43s$	$420.60s$	$414.40s$	98.50%
E-Spiral (R)	$226.04s$	$223.91s$	$231.00s$	96.93%
E-BF (R)	$230.26s$	$230.81s$	$229.60s$	99.47%

time varies from 96.93% to 99.47%, while the energy accuracy varies from 97.54% to 99.91%, approximately. Using Equation (11), it is possible to split the trajectory into simple maneuvers, such as climbing, descending, turning, accelerating/decelerating, and flying at a constant speed. A correct estimation of the energy is crucial for any coverage path. It avoids battery underuse and prevents UAVs from crashing during flights due to battery-exhaustion.

5.2 EG-CPP vs Grid-based Method in Irregular-shaped Areas

The proposed Energy-aware Grid-based CPP (EG-CPP) algorithm described in Section 4.2 is compared to the original grid-based method proposed by VALENTE et al. (2013). Real energy measurements validate the benefits of the energy-aware cost function. These measurements are carried out during real flights using an IRIS quadrotor (see more details about the UAV model at the beginning of 5.1).

A set of simulations exploring irregular-shaped areas were also performed on

Table 6 – Energy consumption in simulation and real flights with the E-Spiral and the E-BF in Polygonal (P) and Rectangular (R) Areas

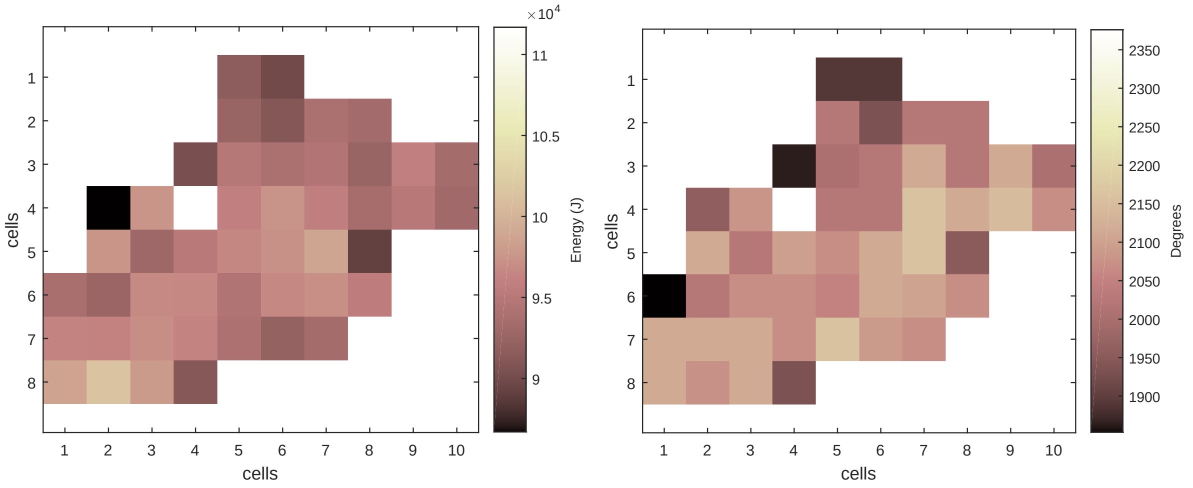
Path/Area	OEM	IEM	Real Flight	Accuracy
E-Spiral (P)	$89924J$	$79158J$	$79228J$	99.91%
E-BF (P)	$85845J$	$87945J$	$85837J$	97.54%
E-Spiral (R)	$47143J$	$46681J$	$47329J$	98.63%
E-BF (R)	$48030J$	$48182J$	$47401J$	98.35%

MATLAB® to state the energy savings in different scenarios. The simulation results are trustworthy. The energy model proved to be very accurate regarding the real energy measured and the estimated one. Further simulation experiments were also executed to analyze the computational time. An algorithm optimization using pruning techniques drastically reduces the computational time and improves the performance of the proposed method. All files and logs from the experiments are available on GitHub in `https://github.com/tauacabreira/EGCPP`.

5.2.1 Cost Function Analysis

The area of interest of the real flights consists of a concave polygon with an internal no-fly zone. The area is discretized into a grid of 8 x 10 cells according to our onboard camera characteristics. Coverage paths for photogrammetry sensing applications demand several additional elements to guarantee mission success. Thus, it is considered elements such as flight altitude (10m), camera resolution (2386 x 2386 pixels), the field of view (100 degrees), and overlapping rate (10%) to determine the size of the cells in the area. A flooding algorithm marks 47 valid cells of 80 during the offline search phase of the coverage path. Cells outside of the area or within the no-fly zone are not included.

The original cost function (O-F) of the grid-based algorithm is represented by Equation (12). The energy-aware cost function (E-F) of our EG-CPP is denoted by Equation (13) - see page 73 for the equations. Each algorithm generates 47 different coverage paths considering every valid cell of the area as a potential starting position. Using the correspondent cost function, it is possible to compute the cost of every path in simulation. Note that this test was possible only thanks to the addition of the pruning techniques in the algorithms. This modification sped up the execution time up to 99% as will be stated in Section 5.2.4, and allows us to run multiple instances of both algorithms. Otherwise, just a single run would take about 3 hours, making the following analysis challenging to achieve. Figure 51a and Figure 51b present the *colormaps* of the resulting coverage paths. Every colored cell represents the cost of the path with minimum-value starting in each one of the valid cells — the darker the color, the lower the cost, and the better the coverage path. The white cells represent the inner and outer no-fly zones. These cells



(a) (b)

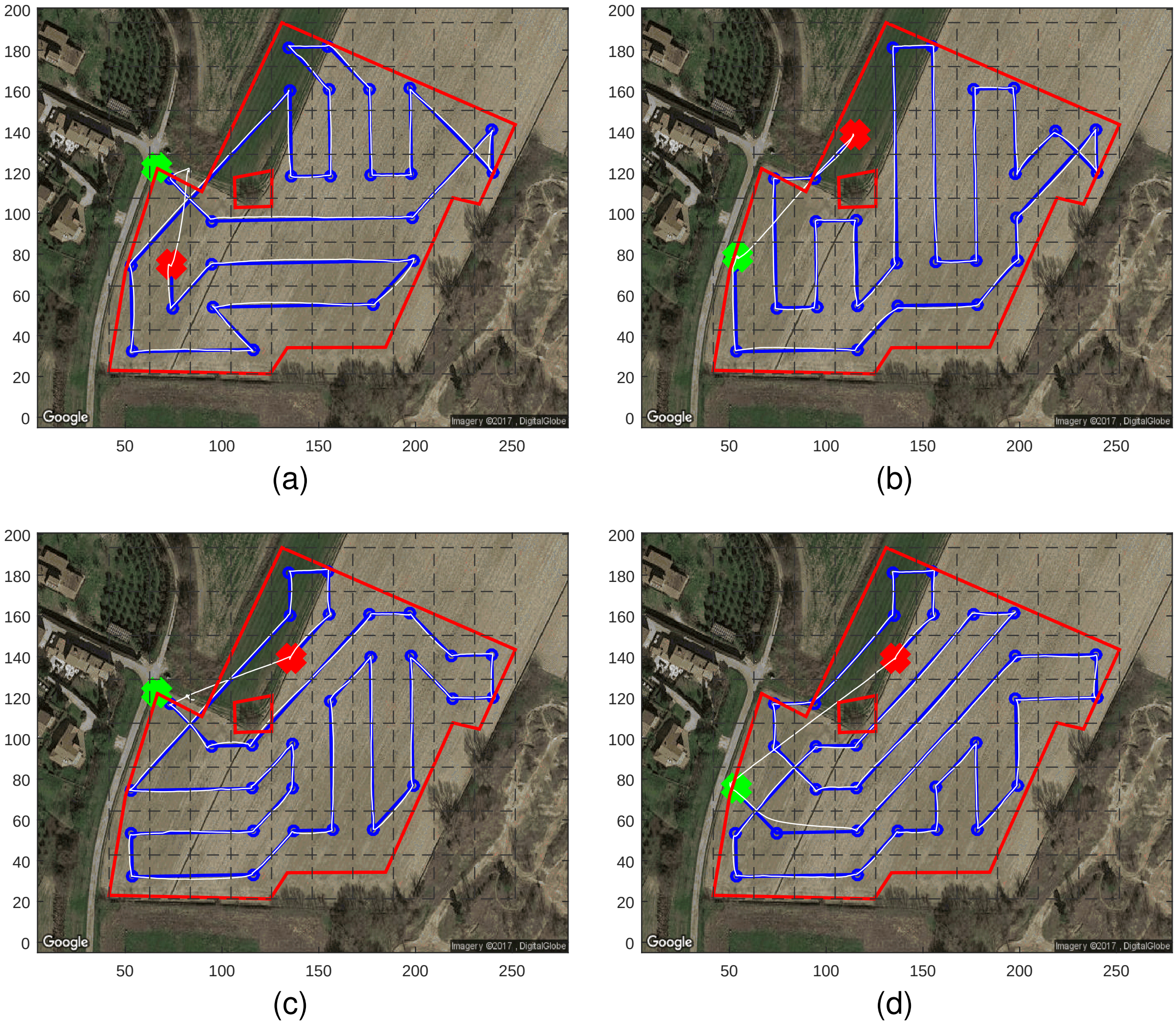
Figure 51 – Colormaps with the minimum-cost paths starting in each one of the 47 valid cells of the workspace: (a) E-F with the ideal starting position in cell (4,2), (b) O-F with the ideal starting position in cell (6,1).

are unreachable areas. They cannot be considered as a valid starting position or as an intermediate cell of the path during the searching process. As can be seen, there is a wide range of values, and the ideal solution depends on the initial point and the adopted cost function.

The minimum-cost path generated by the E-F (Figure 51a) starts at the cell (4,2), while the minimum-cost path generated by the O-F (Figure 51b) starts at the cell (6,1). Both cells are painted in black in the *colormap*. Although the two cost functions produce distinct behaviors, it is possible to identify some similarities in the results presented in the two figures. In both *colormaps*, the ideal starting positions are at one of the corners of the irregular-shaped area, not in the center of the scenario. The O-F computes the minimum-cost path based on the sum of the turning angles, while the E-F explores the energy consumption estimation directly. Both generate different coverage paths with different ideal starting positions. In this way, it is necessary to compare them in order to point out the most accurate cost function. Even if cost functions optimize different metrics, we believe that this is a fair comparison considering that the primary goal of the O-F is indirectly minimizing the energy.

5.2.2 Flight Results

Four real flights were performed to evaluate the energy consumption of the minimum-cost paths generated using O-F and E-F. All flights were executed in a short period (less than an hour), with a speed of $8m/s$ and in a pleasant weather condition with almost zero wind. Figure 52a and Figure 52b show the paths generated with E-F, while Figure 52c and Figure 52d present the paths generated with O-F, both starting from the cells (4,2) and (6,1), respectively. The area of interest and the no-fly zone are marked by the red line, the planned path by the blue line, the performed path during the real flights by the thin white line, the starting position by the green “x", and the final position by the red “x".



(a)

(b)

(c)

(d)

Figure 52 – Four real flights performed with two different starting positions according to the colormap: (a) Path generated with E-F and ideal starting position at Cell (4,2), (b) Path generated with E-F and normal starting position at Cell (6,1), (c) Path generated with O-F and normal starting position at Cell (4,2), (d) Path generated with O-F and ideal starting position at Cell (6,1).

The experimental results can be seen in Figure 53. The green bars represent the flight results, while the yellow bars illustrate the estimated values for the energy consumption obtained through the energy model. One can observe that the proposed E-F obtains the best results, showing an energy saving of 17% in real flights. The path generated with the E-F and starting from the ideal position at the cell (4,2) overcomes the path generated by the O-F using its ideal initial point at the cell (6,1) - the fourth bar vs. the first bar in Figure 53. Even not considering the best choice for the starting position - E-F with cell (6,1), the proposed solution overcomes the original approach -

the third bar vs. the first and second bars.

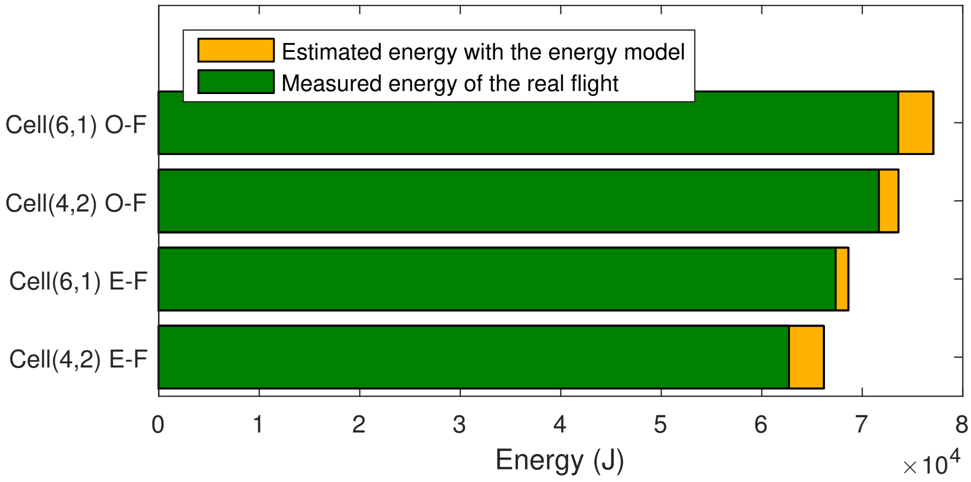


Figure 53 – Energy consumption measured during real flights (in green) and predicted in simulation (yellow) with the O-F and E-F considering two starting positions. The optimum found by O-F (cell (6,1)) behaves the worst and also consumes more than other discarded paths (cell (4,2)) showing that using the number of turns is not accurate enough. The E-F, instead, not only produces better paths given the same starting position (cell (6,1)) but also finds the best path with a resulting energy saving of 17%.

Table 7 – Energy consumption and mission execution time obtained in estimation and real flights with the original and the energy approaches

Path	O-F	E-F	Real Flight	Accuracy
Cell (6,1) O-F	N/A	359.15s	348.60s	97.06%
Cell (4,2) O-F	N/A	342.91s	340.00s	99.15%
Cell (6,1) E-F	N/A	319.97s	317.00s	99.07%
Cell (4,2) E-F	N/A	308.75s	306.00s	99.10%
Cell (6,1) O-F	1890°	$7.7053 \times 10^4 J$	$7.3583 \times 10^4 J$	95.49%
Cell (4,2) O-F	2205°	$7.3593 \times 10^4 J$	$7.1655 \times 10^4 J$	97.36%
Cell (6,1) E-F	1980°	$6.8607 \times 10^4 J$	$6.7354 \times 10^4 J$	98.17%
Cell (4,2) E-F	2025°	$6.6165 \times 10^4 J$	$6.2710 \times 10^4 J$	94.77%

Table 7 presents detailed information about time and energy results comparing original and energy-aware approaches using different starting positions obtained from the *colormaps*. The O-F is not able to estimate the time needed to perform a flight (N/A in Table 7), while the E-F can estimate energy and time. The O-F tries to save energy indirectly by minimizing the total sum of angles. However, while the O-F points out that the path starting from the cell (6,1) presents the minor cost value (1890°), the E-F states that actually, this path spends the higher amount of energy. In this way, the O-F is not trustworthy to indicate the less energy consuming coverage path. The real measurements confirm this assumption and validate our proposed approach with the E-F. The last column of Table 7 presents the accuracy rate of the E-F in estimating energy and time spent during the flights. The logs store information such as velocity, time, current, and voltage. The last two were used to compute real energy.

The energy-aware cost function (E-F) of the EG-CPP is based on the energy model, which was built upon the real measurements and considered external forces, such as the drag of the vehicle. Thus, the E-F can correctly estimate the energy necessary to perform a mission with a high accuracy rate varying from 94% to 98%, approximately. It is possible to estimate the energy needed to perform a given path splitting it into a set of straight lines composed by three phases (acceleration, flying at the constant speed, and deceleration), and a set of turning maneuvers (see Equation (13)). The results lead to interesting conclusions and possibilities. First, it is worth to notice that the *colormap* illustrated in Figure 51a is validated by the results obtained from the energy model. Second, it is possible to apply the offline feasibility test proposed by DI FRANCO; BUTTAZZO (2016) to know in advance if the energy stored in the battery is sufficient to perform a flight. An online feasibility test was also proposed by DI FRANCO; BUTTAZZO (2016) to continually check if the remaining energy in the battery is sufficient to bring the UAV back to the starting position. Therefore, the proposed solution can avoid crashes during flights due to battery-exhaustion.

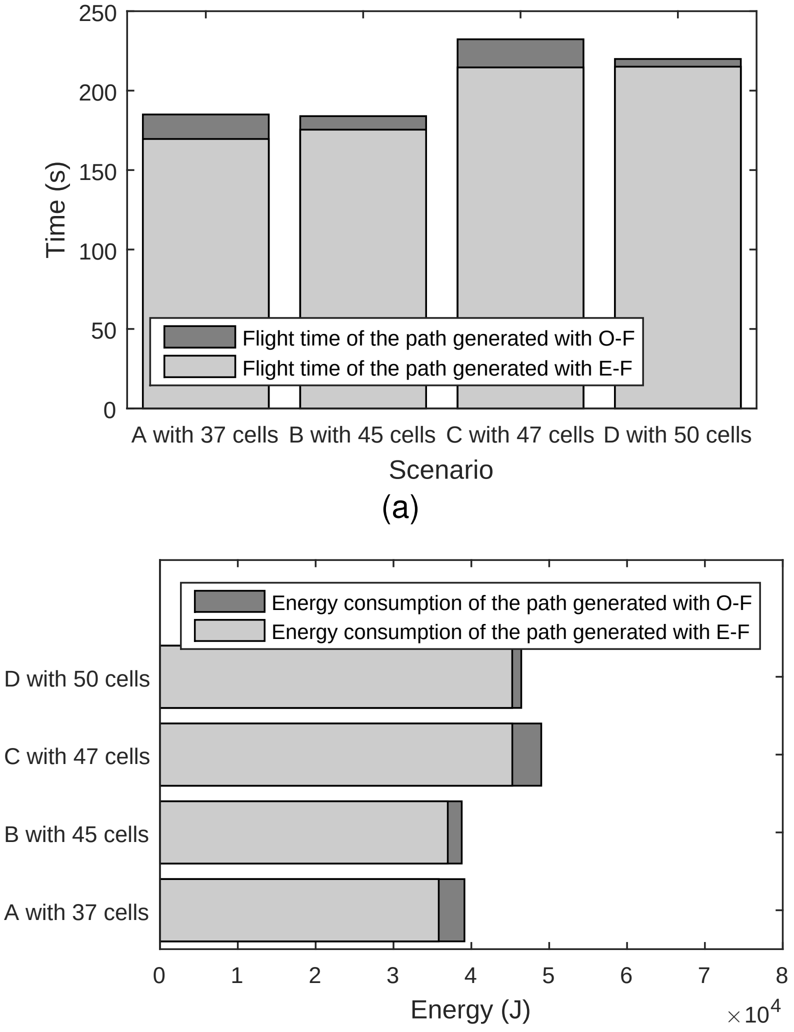
5.2.3 Simulation with Different Scenarios

Additional simulation experiments were performed with different irregular-shaped scenarios. We explored farm regions near the city of Pisa in Italy, importing such scenarios to MATLAB® to compare the O-F and the E-F. As stated in Section 5.2.2, the E-F overcame the O-F in a real situation considering the energy spent during the flight. Furthermore, the E-F presents a high accuracy regarding the estimated energy and the real measured one. In this way, we can trust in these additional simulations to state the effectiveness of the proposed approach.

Table 8 – Energy consumption estimation with the original and the energy approaches

Scenario	**Sum of Angles**		**Energy ($\times 10^4 J$)**		**Energy saving**
	O-F	E-F	O-F	E-F	
A (37 cells)	1764°	1871°	3.9114	3.5810	8.45%
B (45 cells)	1617°	1532°	3.8764	3.6972	4.62%
C (47 cells)	2114°	2205°	4.8980	4.5261	7.59%
D (50 cells)	2009°	1926°	4.6411	4.5249	2.50%

The experiments present four irregular-shaped scenarios with different size and complexity: (a) Scenario A with 37 cells; (b) Scenario B with 45 cells; (c) Scenario C with 47 cells containing a no-fly zone at the Cell (4,5) and (d) Scenario D with 50 cells containing a no-fly zone at the Cell (5,5). Table 8 presents the cost values of O-F and E-F for each scenario and the percentage of energy saving of the proposed approach



(a)

(b)

Figure 54 – Simulation experiment results with four different scenarios: (a) Flight time of the path generated with O-F (in dark grey) and E-F (in light grey), (b) Energy consumption of the path generated with O-F (in dark grey) and E-F (in light grey).

over the original one. Figure 54 presents the simulation experiment results considering the flight time and the energy consumption of the paths generated by the O-F and the E-F. Figure 55 and Figure 56 illustrates the four scenarios and the corresponding complete coverage paths generated by the cost functions.

The paths generated by the original approach are based on the O-F, which is the minimization of the total sum of the angles. For Scenario A containing 37 cells, the minimum-cost path using O-F is the path with 1764°. On the other hand, the paths generated by the proposed approach are based on the E-F, which computes the energy consumption necessary to perform the path. In this case, the minimum-cost path using E-F in Scenario A consumes $3.5810 \times 10^4 J$. One can use the O-F to compute the sum of the angles of the path generated by E-F (1871°), as well as one can use the E-F to estimate the energy spent by the path generated by O-F ($3.9114 \times 10^4 J$). In some cases,

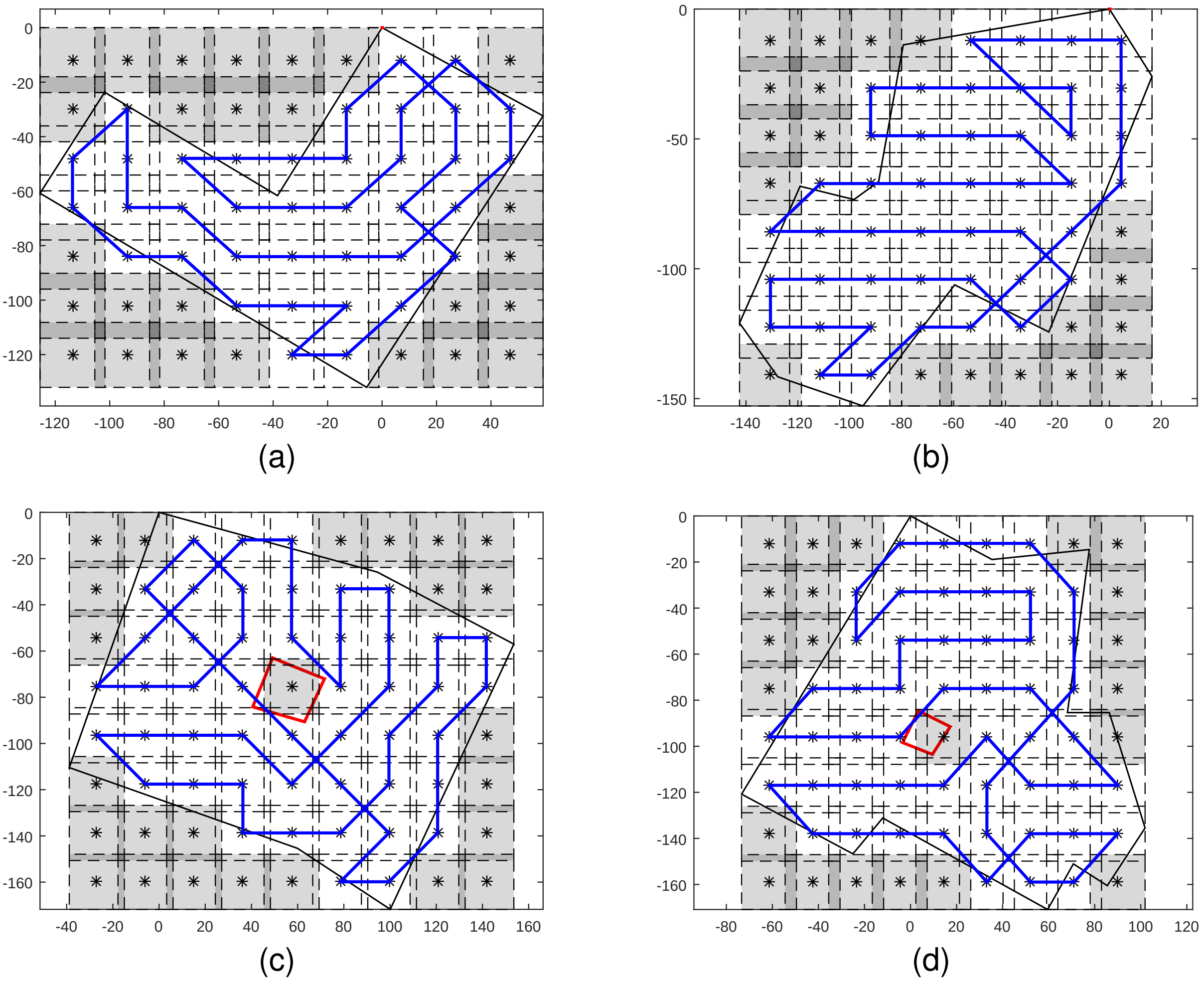


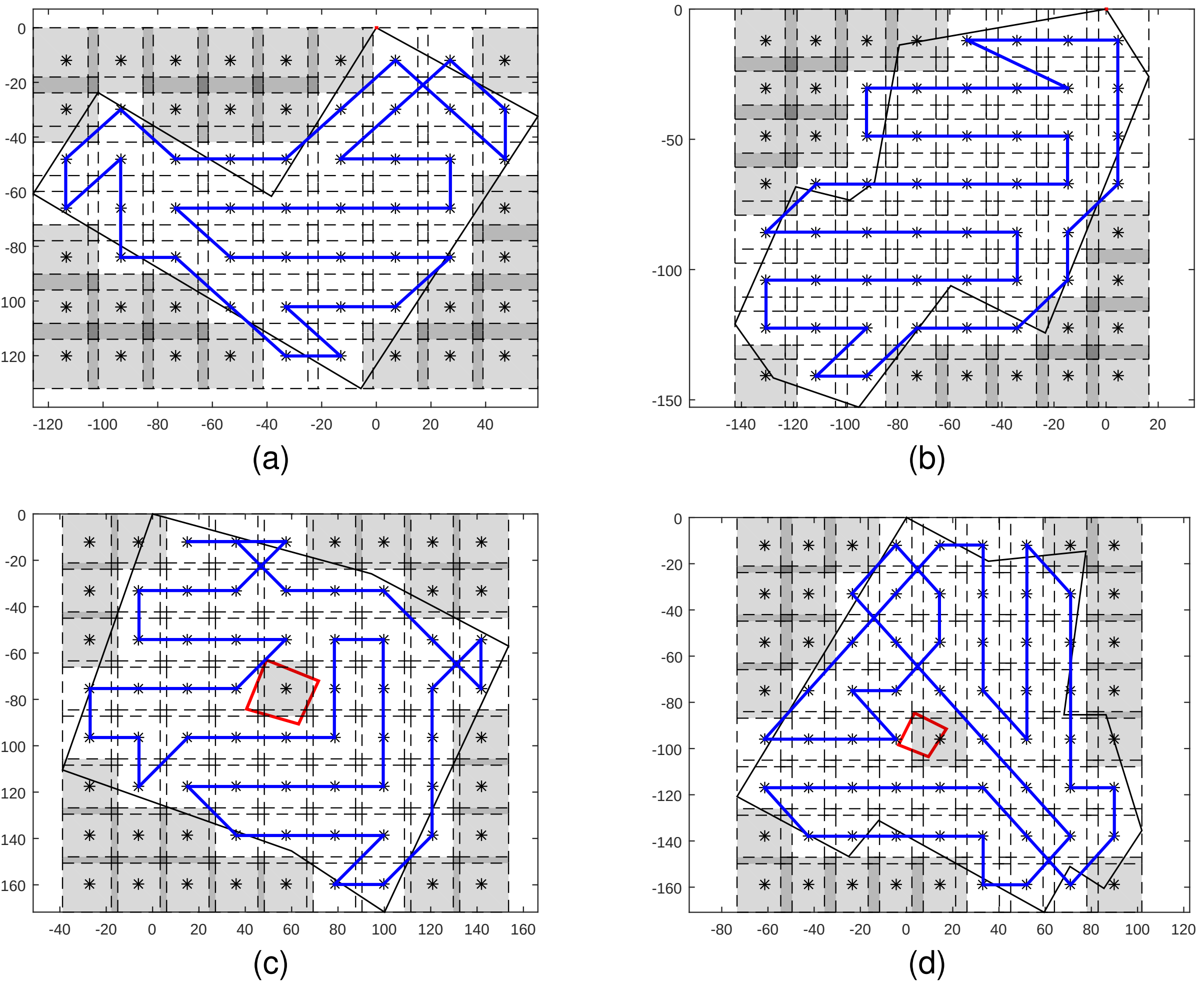
Figure 55 – Four different irregular-shaped scenarios based on farm regions near the city of Pisa in Italy. The coverage paths are generated by the O-F: (a) Scenario A with 37 cells, (b) Scenario B with 45 cells, (c) Scenario C with 47 cells containing a no-fly zone at the Cell (4,5), and (d) Scenario D with 50 cells containing a no-fly zone at the Cell (5,5).

as in the Scenario A and the Scenario C, the lower value for the sum of the angles does not mean the minimum energy consumption, which states that this is not an appropriate metric for generating energy-aware full coverage paths for UAVs. According to the results presented in Table 8, the four paths generated by E-F consume less energy than the paths generated by O-F with energy savings varying from 2.5% to 8.45%.

5.2.4 Computational Time Analysis

The impact of the two pruning techniques in the computational time of the algorithms was evaluated in a series of simulation experiments. The full coverage paths are generated during an offline planning phase. Thus, we are interested in reducing the computational time in order to speed up the total execution time of the mission.

A generic rectangular area composed of $n_c \times 4$ cells, where n_c is the number of



(a)

(b)

(c)

(d)

Figure 56 – Four different irregular-shaped scenarios based on farm regions near the city of Pisa in Italy. The coverage paths are generated by the E-F: (a) Scenario A with 37 cells, (b) Scenario B with 45 cells, (c) Scenario C with 47 cells containing a no-fly zone at the Cell (4,5), (d) Scenario D with 50 cells containing a no-fly zone at the Cell (5,5).

columns of the grid, was adopted to analyze the algorithms. The overall number of cells is increased by incrementing n_c. We are interested in measuring the computation time for a generic number of cells. Note that, we decided to use a rectangular area without obstacles because complex shaped areas, even with the same number of cells, may lead to a high difference in the computational cost.

For the sake of clarity, the grid-based approach proposed by VALENTE et al. (2013) is denoted as Alg. A, the grid-based with the original cost function, but modified by the pruning techniques as Alg. B, and the Energy-aware Grid-based CPP (EG-CPP) as Alg. C - with the pruning and the energy-aware cost function. The computational time of the coverage paths may change according to the adopted starting position. For this reason, we performed each experiment for all starting positions. Figure 57 reports the average of the computational time of Alg. A, Alg. B, and Alg. C in an area with cells varying from

16 to 48 with a step of 4.

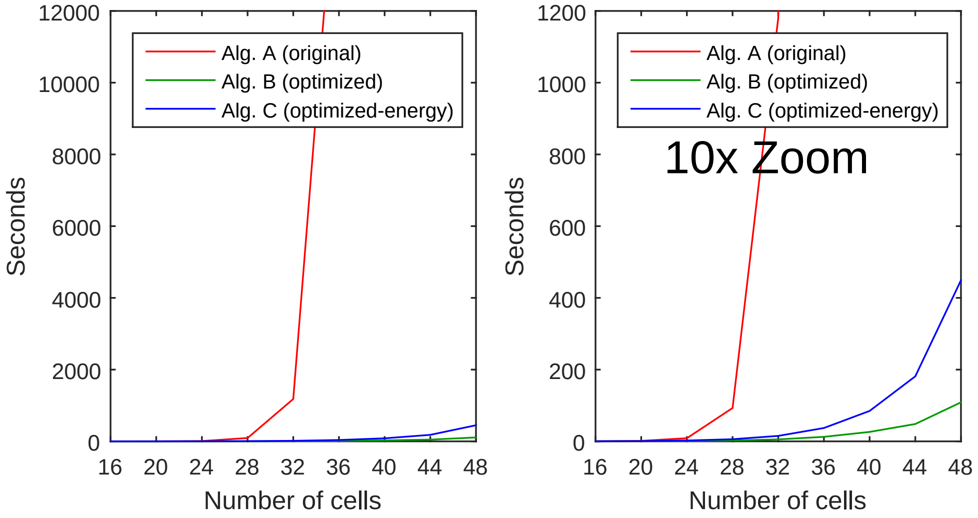


Figure 57 – Plot of the computational time from 16 to 48 cells comparing Alg. A (original), Alg. B (optimized), and Alg. C (optimized-energy). The 10x zoom ease to visualize the small difference between the Alg. B and the Alg. C, which is in order of seconds.

Experimental results show that Alg. A has a significant computational time during the offline planning phase. For instance, in a scenario with 36 cells, the algorithm takes more than three hours to run and find a path. In this way, the optimization proposed in Alg. B drastically reduces the computational time spent by the Alg. A, reaching a percentage of improvement that varies from 44% to 99.9% as the number of cell increases. Results show that the computational time is in the order of seconds instead of hours. Alg. C includes the energy-aware cost function and reaches a percentage of optimization over the Alg. A varying from 31% to 99%. The Alg. C is slightly more expensive than Alg. B but still takes only 35 seconds to be executed on 36 cells, stating the effectiveness of the proposed approach. The original cost function only considers the sum of angles of the path, while the proposed one considers more complex elements, such as acceleration, deceleration, and constant-speed phases.

Finally, we measured the difference in the total number of executions of the recursive function, considering the original and the modified approaches, which explains the improvement in the computation time of the algorithm. Figure 58 illustrates the histogram of the cost for all possible paths searched with Alg. A and B in a scenario with 24 cells. The first approach explores the entire sample field, presenting a wide range of values with a large number of executions for each one them. Several different paths present the same or similar costs. For example, more than 450 paths have the cost of 1400, which is the sum of angles in the original cost function. The optimized approach computes the cost of the paths during the search phase, and discards incomplete high-cost solutions, taking advantage of the pruning technique. In the example shown in Figure 58, the first path computed with the pruning technique had a cost of 1400. All the successive paths

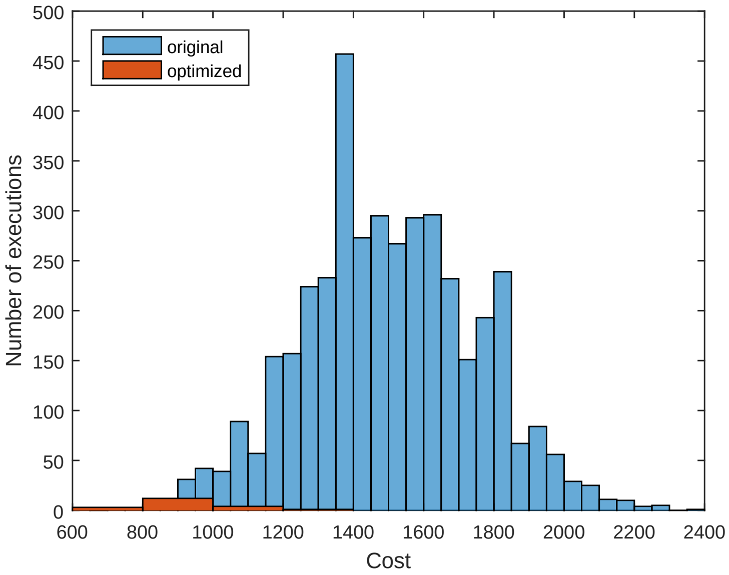


Figure 58 – Histogram of the cost of all possible paths searched with the Alg. A and the Alg. B in a workspace containing 24 cells. Using the optimized approach, the cost of the first complete path was 1400, and all the remaining searched paths, whose costs were higher than this value, were discarded without being fully explored.

(complete or incomplete) higher than this value were discarded. New paths with a lower value will further reduce the search cost. This procedure avoids an excessive number of executions of the recursive function, which is responsible for a considerable time in the algorithm execution.

5.3 NC-Drone Strategies in Areas with Partial Information

The proposed Energy-aware Pheromone-based NC-Drone method described in Section 4.3 is evaluated through several simulation experiments along with its all variations and strategies. The approaches were implemented in the NetLogo platform, a multi-agent programmable modeling environment. UAVs were modeled as agents and were able to navigate and interact with each other in a 50 x 50 grid scenario. Following the centralized approach, agents were able to read and write information in the cells of the scenario - the cells are also known as patches in the platform. In the decentralized variations, agents can exchange matrix information by communicating among themselves.

We employed four UAVs during the experiments. All vehicles start the patrolling task at the lower left corner of the area and move to a cell in a single cycle of simulation. Thirty simulations with 10.000, 15.000 and 20.000 cycles were executed for each approach. We compute the mean (M) and the standard deviation (SD) of the simulations and employ the Student's T-Test to check if there is a statistical difference among the approaches.

Three different performance metrics were adopted during the experiments, such as

Quadratic Mean of the Intervals (QMI), Standard Deviation of the Frequencies (SDF), and Number of 90° Turning Maneuvers (NTM). All metrics were previously described in Section 2.3.3. The performance metrics highlight different application requirements depending on the main goal of the task. We are seeking a uniform spatial and temporal distribution of visits while minimizing the number of turns. Furthermore, we need to keep the unpredictable behavior of the path for an external observer or intelligent target while minimizing energy consumption.

The energy estimation using an accurate energy model is the most reliable performance metric for coverage paths, as stated in Section 5.2 and Section 5.1. However, we understand that the most energy efficient coverage paths are the ones with prolonged straight segments and smoother turning maneuvers, which keeps the optimal speed for longer periods. In this way, we adopted a simplified version of the energy model to maintain the UAV following the sweeping direction as long as it is possible. Since we are dealing with an online coverage in a unknown environment, the UAVs must explore the scenario gathering information through their onboard sensors. During the patrolling, the aerial vehicles only consider the amount of pheromone in the surrounding neighbors to decide the next location to visit. When there is two or more cells with the same minimum-value, the UAV prefers the cell aligned with the sweeping direction. Therefore, the drastically reduction in the excessive number of turning maneuvers presents a real impact in the energy consumption.

One can download all algorithms implemented in NetLogo and the complete set of results of the simulations through the link `https://github.com/tauacabreira/NCDrone`.

5.3.1 Centralized NC-Drone vs Real-time Search Methods

The Centralized NC-Drone is compared with NC, LRTA*, TVUR, and WVUR. All approaches allow the vehicles to write virtual pheromones in the visited places. These digital marks can be read by all vehicles while they are moving around the area. The centralized process senses the neighbor cells, evaluates the number of visits on each one of them, chooses the least visited one and sends the motion coordinates to the vehicles. Table 9 and Table 10 present the mean (M) and the standard deviation (SD) results achieved by the approaches regarding QMI and SDF metrics, respectively. The best approaches and results are highlighted in bold for each metric.

The Centralized NC-Drone proves to be the best approach among the evaluated real-time search methods regarding QMI metric, overcoming all approaches with a difference that is considered to be extremely statistically significant, with 95% of confidence. In 20K cycles, the Centralized NC-Drone improves the QMI metric around 5% compared to the TVUR approach, which is the second-best approach in this metric, and 7% compared to the original NC. The SDF results point out that Centralized NC-Drone overcomes NC, LRTA* and TVUR in 10K and 20K cycles, while presents a result equivalent to the one

Table 9 – QMI results for Centralized NC-Drone and Real-time Search Methods.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	**768.55**	**28.29**	**770.34**	**16.24**	**773.48**	**9.26**
NC	806.62	9.07	826.70	8.29	829.54	6.47
LRTA*	805.73	8.23	820.97	7.94	828.07	5.87
TVUR	791.89	7.29	808.93	7.38	813.12	6.24
WVUR	816.55	9.28	831.88	8.84	840.48	6.82

achieved by WVUR. In 15K cycles, the Centralized NC-Drone is superior to NC and TVUR - around 11% compared to both approaches, and it is similar to the WVUR and LRTA* approaches.

Table 10 – SDF results for Centralized NC-Drone and Real-time Search Methods.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	**0.73**	**0.08**	**0.75**	**0.09**	**0.76**	**0.09**
NC	0.84	0.13	0.84	0.10	0.83	0.10
LRTA*	0.83	0.08	**0.78**	**0.08**	0.84	0.09
TVUR	0.87	0.10	0.84	0.10	0.84	0.09
WVUR	**0.77**	**0.08**	**0.76**	**0.08**	**0.76**	**0.09**

The vehicle may concentrate the navigation in specific locations for long periods. This behavior leads to unstable results, as documented in the literature. The Centralized NC-Drone presents a tendency to explore the scenario while avoids constant changes in the sweeping direction. These changes are provoked by the random process to decide between cells with the same value. In this way, the centralized algorithm obtains better results in the QMI and the SDF, once both performance metrics value the regularity. The WVUR approach presents good results only for the SDF metric and has the worse values for the QMI metric.

The NTM metric results are presented in Table 11. The Centralized NC-Drone overcomes all approaches in all simulations with values three or four times smaller than the other methods. Our proposed NC-Drone modifies the original algorithm by choosing the cell aligned with the sweeping direction in case of a tie between the cells with the lowest costs. It adopts a simplified version of the energy model, where we explore the prolongation of straight segments of the path and the minimization of the number of turning maneuvers to reduce the total energy consumption. Since we kept the speed

Table 11 – NTM results for Centralized NC-Drone and Real-time Search Methods.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	**5.8k**	**587.36**	**9.9k**	**653.55**	**14.5k**	**621.11**
NC	23.7k	78.87	35.6k	111.88	47.5k	150.01
LRTA*	24.2k	120.08	36.3k	141.23	48.4k	151.23
TVUR	24.0k	111.59	36.0k	108.93	48.1k	160.95
WVUR	23.8k	86.54	35.7k	110.46	47.6k	147.78

constant during the patrolling missions, we are not considering other relevant elements, such as the traveled distance and the optimal speed, to save energy.

5.3.2 Decentralized NC-Drone variations

In this section, we compared the proposed decentralized variations exploring the matrix-based communication protocol with different synchronization methods: Decentralized NC-Drone MULTI (D. MULTI), Decentralized NC-Drone MAX (D. MAX), and Decentralized NC-Drone AVG (D. AVG). The centralized version overcomes all decentralized variations once it has unlimited access to all information available in the environment. Thus, we added centralized results as an upper bound reference. The best decentralized-approach and its results are highlighted in bold for each metric.

Table 12 – QMI results for the Decentralized NC-Drone variations.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	768.55	28.29	770.34	16.24	773.48	9.26
D. MULTI	849.80	16.24	884.63	11.21	895.21	10.82
D. MAX	**797.58**	**10.97**	**816.81**	**8.54**	**825.43**	**6.86**
D. AVG	**795.57**	**9.60**	**813.00**	**5.80**	**823.67**	**7.36**

Considering only the decentralized versions, the D. MAX and the D. AVG present the best results in the QMI metric merging the information of the matrices every time the vehicles pass by each other. The D. MULTI presents the worse results only combining the information to make a decision every time, instead of merging the matrices. Ranking the methods according to this metric, we obtain C. NC-Drone, D. AVG, D. MAX, and D. MULTI. Table 12 summarizes the QMI metric results.

In the SDF results presented in Table 13, D. MULTI shows the best results and achieves the most uniform visitation among the decentralized variations. The results

Table 13 – SDF results for the Decentralized NC-Drone variations.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	0.73	0.08	0.75	0.09	0.76	0.09
D. MULTI	**1.14**	**0.13**	**1.14**	**0.11**	**1.16**	**0.15**
D. MAX	2.10	0.15	2.50	0.21	2.92	0.17
D. AVG	2.21	0.11	2.77	0.11	3.14	0.15

presented by the D. MULTI are stable during the three different cycles of simulation, while the values shown by D. MAX and D. AVG increases over time. The information merging techniques, D. MAX and D. AVG, are not able to equally distributed the visits between the cells as the D. MULTI. Ranking the methods according to this metric, we obtain C. NC-Drone, D. MULTI, D. MAX, and D. AVG.

Table 14 – NTM results for the Decentralized NC-Drone variations.

Approaches	10K		15K		20K	
	M	SD	M	SD	M	SD
C. NC-Drone	5.8k	587.36	9.9k	653.55	14.5k	621.11
D. MULTI	7.9k	521.24	13.8k	634.17	20.0k	674.05
D. MAX	**6.5k**	**282.11**	**11.1k**	**294.72**	**15.5k**	**316.20**
D. AVG	**6.4k**	**266.83**	**10.8k**	**415.43**	**15.8k**	**396.71**

The NTM results are summarized in Table 14. D. MAX and D. AVG present the best results in this metric, considering the decentralized approaches exclusively. According to the Student's T-test, the difference between the two decentralized methods is considered to be not statistically significant. Despite providing the best results in the SDF metric, D. MULTI presents a more elevate number of turns. The higher the number of cycles in the simulation, the higher the number of turning maneuvers. The percentage of increasing regarding the turns varies from 68.5% to 74.68% between the simulations with 10k and 15k, while the increasing between 15k and 20k is around 40~45%. Ranking the methods according to this metric, we obtain C. NC-Drone, D. MAX/AVG, and D. MULTI.

5.3.3 Watershed Strategy

In this section, experiments performed with the Watershed Strategy (WS) applied to the original approaches are presented. Table 15 presents the results for these strategies considering the QMI metric with the mean (M), standard deviation (SD). We also show the best and worst outcomes of each approach.

Table 15 – QMI results for the Watershed Strategy.

Approaches	M	SD	BEST	WORST
C. NC-Drone	772.33	33.24	737.95	860.15
Centralized-WS	**745.87**	**7.96**	**736.84**	**764.53**

The Centralized Watershed strategy outperforms the original centralized version with a difference considered to be statistically significant. Furthermore, the Watershed strategy drastically reduced the SD in 76%. With a low SD, we can provide a more stable behavior for the algorithm, as can be seen in the best and the worst results in the last two columns of the table. The decentralized variations (MULTI, MAX, and AVG) present no improvements by adopting the Watershed strategy, and its results have been omitted from the aforementioned table.

Moreover, the QMI improvement in the centralized algorithm provided by the novel strategy impacts neither negatively nor positively the SDF metric. In other words, there is no significant change in statistical terms in the results achieved by the centralized and distributed variations of the algorithm adopting the Watershed strategy. We also omitted here the results of the SDF metric for brevity.

Table 16 – NTM results for the Watershed Strategy.

Approaches	M	SD	BEST	WORST
C. NC-Drone	**5.7k**	**523.18**	**4.0k**	**6.5k**
Centralized-WS	7.4k	133.96	7.1k	7.7k

The NTM metric is depicted in Table 16. One can observe an increase in the value achieved in this metric considering the Watershed strategy, which states a correlation between the QMI improvement and the high number of turns in those algorithms.

5.3.4 Time-based Strategies

In this section, we show the experiments performed with the Time-based Strategies (TS and WSTS) applied to the original approaches. Table 17 presents the results for these strategies considering the QMI metric with the mean (M), standard deviation (SD), the best and the worst results.

Both TS and WSTS outperform the original centralized version, presenting a difference considered to be statistically significant. Furthermore, the Time-based Watershed (WSTS) strategy drastically reduced the SD in 88%. As we previously mentioned, with a low SD, we can provide a more stable behavior for the algorithm, as can be seen in

Table 17 – QMI results for the Time-based Strategy.

Approaches	M	SD	BEST	WORST
C. NC-Drone	772.33	33.24	737.95	860.15
Centralized-TS	752.39	29.74	718.46	840.64
Centralized-WSTS	**747.30**	**4.12**	**739.50**	**754.56**

Table 18 – NTM results for the Time-based Strategy.

Approaches	M	SD	BEST	WORST
C. NC-Drone	5.7k	523.18	4.0k	6.5k
Centralized-TS	**5.4k**	**449.35**	**4.2k**	**6.3k**
Centralized-WSTS	8.1k	92.18	7.9k	8.3k

the best and the worst results in the last two columns of the table. Once again, the decentralized variations (MULTI, MAX, and AVG) present no improvements by adopting the Time-based strategy and have no impact in the SDF metric, whose results were omitted from the table.

The NTM metric performance results are illustrated in Table 18. As one can see, Centralized-TS emerges as a promising strategy once it improves the QMI performance while keeping the SDF and the NTM metrics stable. In these two metrics, the results are similar to the ones in the original Centralized NC-Drone.

5.3.5 Evaporation Strategies

We have run four different experiments for each algorithm with intervals of 100, 250, 500, and 1k between each evaporation. Since the evaporation process adds a computational cost to the original algorithm every time it runs, we empirically tested four values to evaluate the performance of the ES and finds out the ideal interval. The evaporation factor (EF) is 0.1, which means that the amount of pheromone drops 0.1 at every x intervals. Table 19 presents the ES results considering the QMI metric in the centralized NC-Drone and its decentralized variations.

The ES improved the Centralized NC-Drone in the QMI metric in all intervals with a difference considered extremely significant in statistical terms. Centralized-ES-100 presented an improvement of around 10% over the original algorithm. Moreover, the ES reduced the standard deviation (SD) around 75% in all runs, making the algorithm more stable. The ES also improved the Decentralized NC-Drone MULTI (D. MULTI) in all runs and the Decentralized NC-Drone MAX (D. MAX) in the first interval (MAX-ES-100). Despite the improvement of around 5% in the D. MULTI, the D. MAX is still the

Table 19 – QMI results for the Evaporation Strategies.

Approaches	M	SD	BEST	WORST
C. NC-Drone	772.33	33.24	737.95	860.15
Centralized-ES-100	**700.97**	**8.59**	**683.4**8	**717.42**
Centralized-ES-250	702.49	8.22	683.69	723.09
Centralized-ES-500	706.60	8.66	689.79	726.23
Centralized-ES-1k	705.69	8.32	682.12	719.87
D. MULTI	849.80	16.24	816.43	877.47
MULTI-ES-100	806.84	14.18	782.82	844.45
MULTI-ES-250	807.71	15.23	767.34	833.86
MULTI-ES-500	809.62	12.45	790.97	836.98
MULTI-ES-1K	820.37	15.89	795.05	875.16
D. MAX	797.58	10.97	787.56	825.92
MAX-ES-100	**789.98**	**10.59**	**773.25**	**827.16**
MAX-ES-250	794.31	9.40	777.21	816.73
MAX-ES-500	794.13	8.87	773.47	810.93
MAX-ES-1K	793.98	7.98	782.45	811.41
D. AVG	795.57	9.60	780.80	828.88
AVG-ES-100	792.50	9.27	777.38	812.09
AVG-ES-250	793.26	9.13	773.47	810.61
AVG-ES-500	791.13	8.60	776.04	810.62
AVG-ES-1K	794.95	10.60	776.01	824.40

best algorithm among the decentralized variations in the QMI metric. There was no improvement in the Decentralized NC-Drone AVG (D. AVG).

The QMI results for the Centralized NC-Drone and D. MULTI worsened as the interval grows. The majority of cells do not stay without a visit for periods longer than 1k cycles of simulation. In this way, an evaporation process with this interval is rarely triggered and does not improve the algorithm's performance as it does with smaller intervals.

While this strategy improves both Centralized NC-Drone and D. MULTI in the QMI metric, the SDF metric performance drops in all runs by adopting the ES. On the other hand, there was a slight improvement in the D. MAX and D. AVG. Despite this improvement, the MULTI remains the best algorithm among the decentralized variations in the SDF metric. The improvements provided by ES in the QMI metric also present a negative impact on the NTM metric. The total number of turning maneuvers almost triplicate in the centralized version of the NC-Drone, drastically decreasing the performance in this metric. Once more, we can observe the trade-off and correlation between the QMI and NTM metrics. It is necessary to perform a large number of turns to obtain a more spatially distributed patrolling. The number of turns also increase in the D. MAX and D. AVG, but not so much as in the Centralized NC-Drone and D. MULTI. We have omitted here the results of both SDF and NTM metrics.

5.3.6 Communication-Frequency Strategies

We have run five different experiments for each algorithm with intervals of 50, 100, 250, 500, and 1000 cycles between every synchronization among the vehicles. Since communication presents an additional cost to the NC-Drone, we investigated the impact of different communication ranges empirically selected to evaluate the performance of the CFS. Figure 59 shows the results of the QMI metric considering the CFS.

The original Centralized NC-Drone behaves as a CFS-1, where all vehicles have access to all available information in the scenario at every step of the simulation. Thus, we can set Centralized NC-Drone as the upper bound solution for the experiments. CFS-50 presents the best results regarding the QMI metric considering this strategy. It obtains an outcome equivalent to the one presented by the Centralized NC-Drone, even sharing the matrices only at every 50 cycles. According to the Student's T-test, the difference between the two approaches is considered to be not statistically significant. However, as the interval between synchronizations doubles, metric performance falls between 7% and 8%. This behavior can be seen in all variations. The standard deviation also increases as the interval grows.

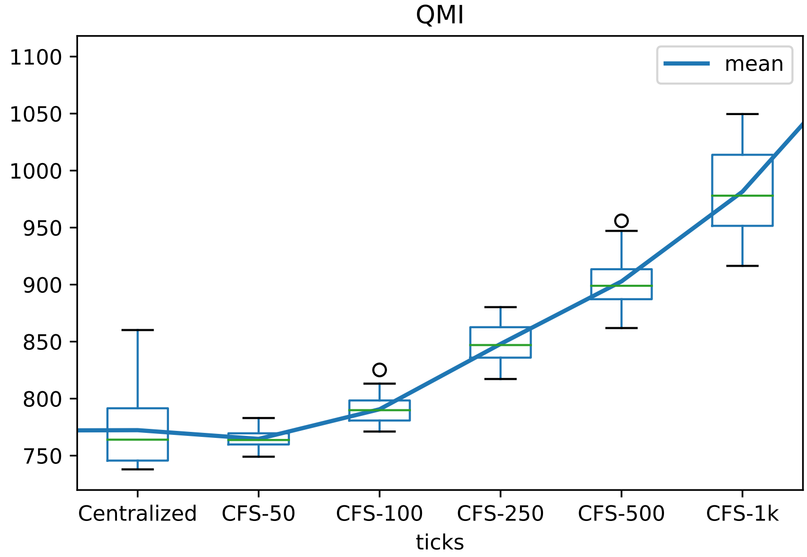


Figure 59 – QMI results for the Communication-Frequency Strategies.

The results of this strategy regarding the SDF metric are illustrated in Figure 60. In this metric, the performance drop of the CFS is not linear and increases more and more as the interval grows.

The NTM results for the experiment with the CFS strategy is presented in Figure 61.

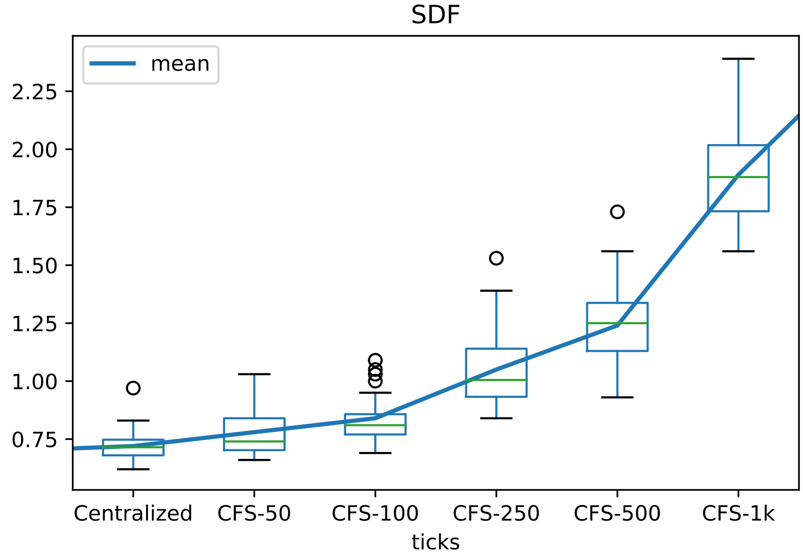


Figure 60 – SDF results for the Communication-Frequency Strategies.

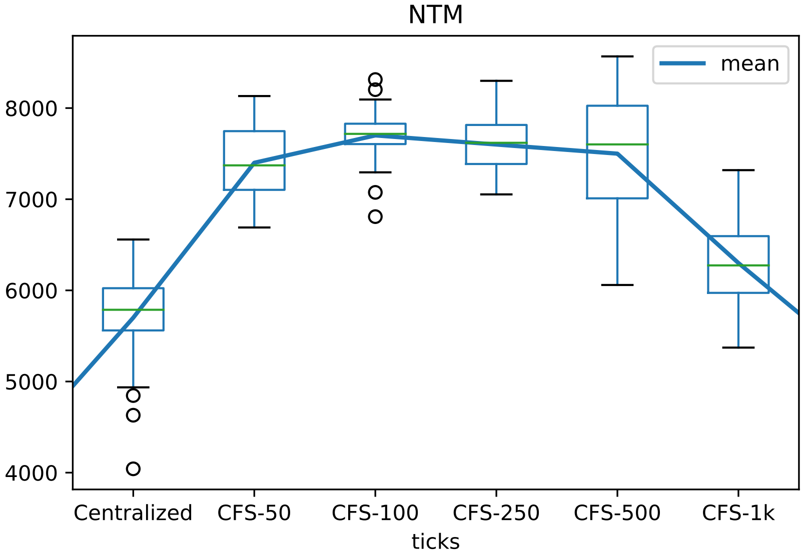


Figure 61 – NTM results for the Communication-Frequency Strategies.

In this strategy, the number of turning maneuvers increases compared to Centralized NC-Drone. Considering only the CFS strategies, the NTM continues to drop as the interval of synchronization grows from CFS-100 to CFS-1000. In these cases, the vehicles tend to follow straight paths for more extended periods and perform less turning maneuvers, since they have less information available about the visits. A UAV ends up visiting areas already covered by other vehicles, once in its internal map, these areas are not covered yet. This particular behavior leads to a performance drop of the QMI metric. Thus, it is necessary to perform more turns to obtain a more uniform distribution of visits.

Moreover, one can verify that there is a correlation between the presented metrics. The Pearson Correlation coefficient between QMI and NTM is -0.7713, which is a strong negative correlation, meaning that whenever QMI values grow, NTM tends to fall, and vice versa. The coefficient between SDF and NTM is -0.88963, and there is also a strong negative correlation. However, the coefficient between QMI and SDF is positive (0.9716), which means that the two metrics tend to grow together over time.

5.3.7 Combination of Strategies

Considering the improvements provided by the individual algorithms CFS, WS, TS, and ES in previous sections, we present here a few combinations and a final algorithm composed of all strategies together:

- CFS-WS-X explores global communication at regular intervals to exchange information and computes clusters of less visited cells to balance coverage;
- CFS-WS-ES-X combines the previous strategy with Evaporation Strategy reducing the amount of pheromone in certain places not visited for a while;
- CFS-WS-ES-Time-X includes tie-break between two places with the same minimum number of visits, selecting the less recently visited place;
- CFS-WSTS-ES-X adopts a Watershed Strategy based on the time of the last visit, instead of using the number of visits to form the clusters; and
- CFS-WSTS-ES-Time-X combines the last two strategies with the Evaporation Strategy. In all strategies, the evaporation factor drops 0.1 at every 100 cycles of simulation.

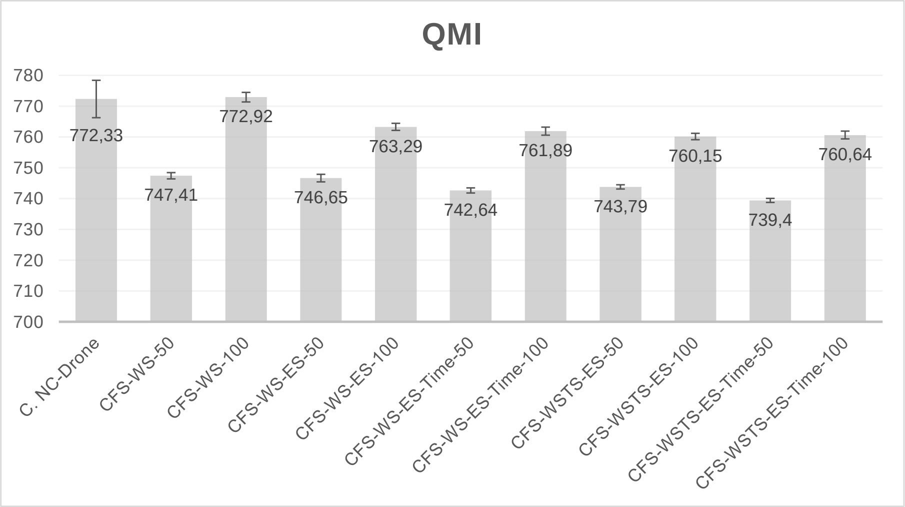


Figure 62 – Results of QMI metric with plot bars for the Combination of Strategies.

Table 20 – QMI results for the Combination of Strategies.

Approaches	M	SD	BEST	WORST
C. NC-Drone	772.33	33.24	737.95	860.15
CFS-WS-50	747.41	5.58	736.92	756.99
CFS-WS-100	772.92	8.50	756.09	797.12
CFS-WS-ES-50	746.65	6.74	730.91	758.80
CFS-WS-ES-100	763.29	6.30	750.81	782.08
CFS-WS-ES-Time-50	742.64	4.56	736.29	751.75
CFS-WS-ES-Time-100	761.89	7.17	750.84	781.67
CFS-WSTS-ES-50	743.79	3.77	734.30	750.69
CFS-WSTS-ES-100	760.15	5.69	748.47	772.23
CFS-WSTS-ES-Time-50	**739.40**	**3.62**	**731.20**	**747.21**
CFS-WSTS-ES-Time-100	760.64	6.95	745.13	784.05

The results of the QMI metric for the combination of strategies are presented in Table 20 and Figure 62. The table presents the mean (M), standard deviation (SD), the best, and the worst result for each strategy, while the figure presents the mean and the standard error. We performed experiments with communication intervals of 50 and 100 cycles for each combination. The strategies with an interval of 50 cycles overcome the ones with an interval of 100.

The achieved results are slightly improved as more combinations of strategies are adopted in both cases (intervals of 50 and 100). The standard deviation also drops as the combinations are formed, making the solution more stable. Combinations containing the tie-break based on time present the best results in the QMI metric, CFS-WS-ES-Time-X, and CFS-WSTS-ES-Time-X. These solutions presented an improvement of 4% and 4.5% over the Centralized NC-Drone, despite sharing matrix information only at every 50 cycles of simulation. The combined solution with a larger interval of 100 cycles also overcomes the centralized algorithm.

Table 21 and Figure 63 show the results of the SDF performance metric considering the combination of strategies. The results obtained by the CFS-WS-50 are similar to the ones obtained by the Centralized NC-Drone in the SDF metric. However, as the combination of strategies improves QMI results, SDF scores become slightly worse. The SDF metric varies from 0.73 in the CFS-WS-50 to 1.09 in the CFS-WSTS-ES-Time-100. Since QMI and SDF are independent metrics from each other, it is possible to choose the most appropriate combination of strategies depending on the application requirement.

The NTM metric also presents a performance drop as more strategies are combined,

Table 21 – SDF results for the Combination of Strategies.

Approaches	M	SD	BEST	WORST
C. NC-Drone	**0.72**	**0.07**	**0.62**	**0.97**
CFS-WS-50	0.73	0.05	0.63	0.85
CFS-WS-100	0.79	0.06	0.68	0.92
CFS-WS-ES-50	0.84	0.05	0.76	1.02
CFS-WS-ES-100	0.95	0.08	0.85	1.20
CFS-WS-ES-Time-50	0.85	0.05	0.75	0.98
CFS-WS-ES-Time-100	0.92	0.07	0.81	1.11
CFS-WSTS-ES-50	0.93	0.06	0.83	1.05
CFS-WSTS-ES-100	0.97	0.08	0.84	1.15
CFS-WSTS-ES-Time-50	1.02	0.10	0.86	1.30
CFS-WSTS-ES-Time-100	1.09	0.10	0.91	1.38

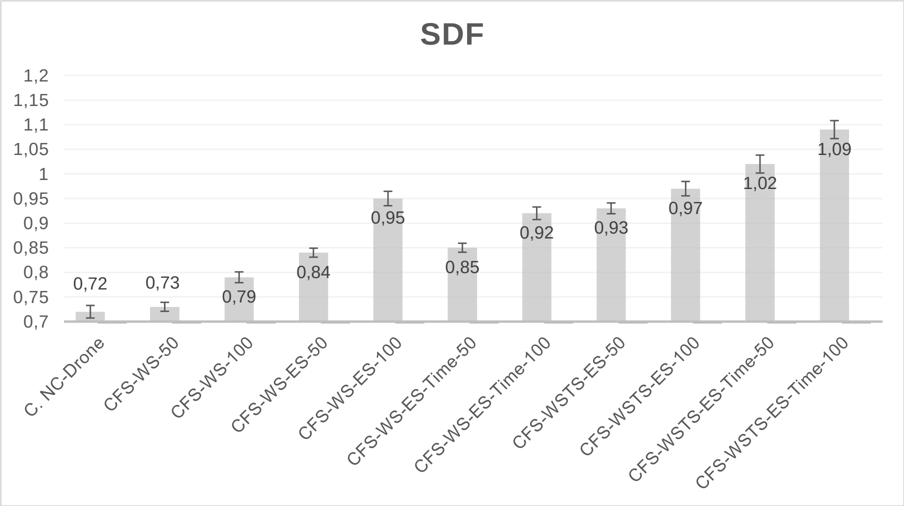


Figure 63 – Results of SDF metric with plot bars for the Combination of Strategies.

varying from 7.7k in the CFS-WS-50 to 9.8k in the CFS-WSTS-ES-Time-100. Once more we can verify the correlation between the QMI and the NTM performance metric. A larger number of turning maneuvers is necessary to obtain a uniform patrolling. Table 22 and Figure 64 show the NTM results considering the combination of strategies.

5.4 Discussion

This chapter presented the experiments performed to evaluate our proposed approaches with state-of-the-art coverage strategies. In Section 5.1, we compared our E-Spiral flight pattern with the E-BF, an energy-aware back-and-forth algorithm proposed

Table 22 – NTM results for the Combination of Strategies.

Approaches	M	SD	BEST	WORST
C. NC-Drone	**5.7k**	**523.18**	**4.3k**	**6.6k**
CFS-WS-50	7.7k	188.25	7.3k	8.0k
CFS-WS-100	7.9k	243.28	7.4k	8.4k
CFS-WS-ES-50	9.1k	199.17	8.7k	9.5k
CFS-WS-ES-100	9.4k	235.31	9.0k	9.9k
CFS-WS-ES-Time-50	8.9k	149.82	8.5k	9.1k
CFS-WS-ES-Time-100	9.3k	236.48	8.8k	9.7k
CFS-WSTS-ES-50	9.5k	187.33	9.0k	9.9k
CFS-WSTS-ES-100	9.7k	235.46	9.3k	10.1k
CFS-WSTS-ES-Time-50	9.6k	231.12	9.2k	10.2k
CFS-WSTS-ES-Time-100	9.8k	239.99	9.3k	10.4k

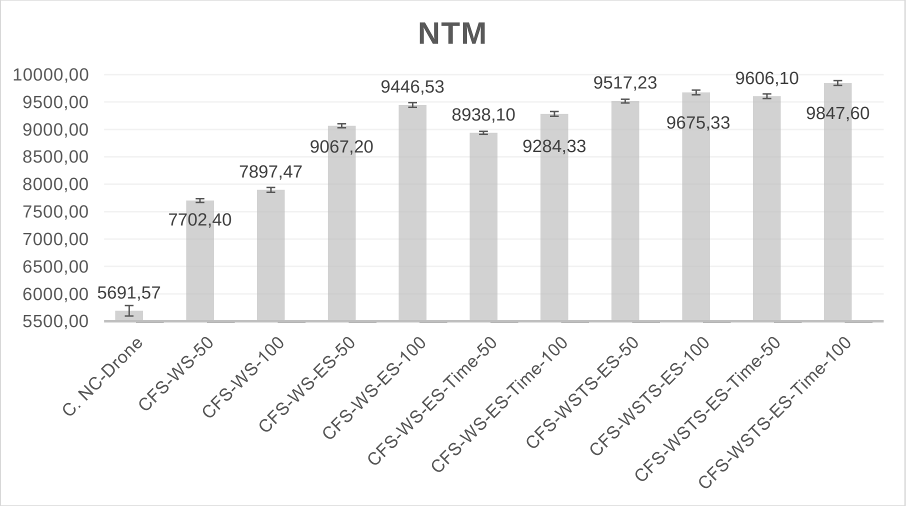


Figure 64 – Results of NTM metric with plot bars for the Combination of Strategies.

by DI FRANCO; BUTTAZZO (2016). We performed a massive number of simulations on MATLAB®, exploring 3750 different areas of interest. These areas presented a variety of characteristics, such as the number of vertices (from 6 to 10), the levels of irregularity (from 0 to 1), and the size of the area (diameter from 200 to 600). The E-Spiral overcomes the E-BF in all cases. Our algorithm performs better in rounded-shaped areas containing a more significant number of vertices, reaching a percentage of improvement of around 16.1%. In such scenarios, the UAVs can maintain the optimal speed for more extended periods and perform smoother turning maneuvers. The different levels of irregularity maintain the improvement of the E-Spiral over the E-BF stable at around 13%. In larger areas with a diameter of 600m, the percentage of improvement of the E-Spiral drops to around 10%, but our approach still overcomes the E-BF. The real flights performed in two different areas of interest also state the effectiveness of our proposed approach and the accuracy of the improved energy model. The E-Spiral reduced the mission execution time and energy consumption by 9% and 7.7% in the polygonal area, respectively. The accuracy rate of the energy model in estimating the energy was computed comparing the energy obtained through the equations of the energy model on MATLAB® with the log information obtained from the real flights.

In Section 5.2, we presented a comparison between the EG-CPP with the energy-aware cost function (E-F) and the grid-based algorithm with the sum of the angles cost function (O-F) proposed by VALENTE et al. (2013). We explored an irregular-shaped area containing 47 valid cells and one internal no-fly zone. We run both algorithms, starting in each one of the 47 valid cells of the scenario. Each algorithm pointed out a minimum-cost path starting at a different position. The EG-CPP indicated the cell (4,2), while the grid-based indicated the cell (6,1). The real flights performed over the irregular-shaped area stated the reliability of our proposed approach. The path generated with the E-F consumed less energy than the path computed by the O-F (around 17% of energy reduction) - even with the O-F indicating a path with a smaller sum of angles. In this way, the O-F is not trustworthy to minimize energy. Additional experiments were also performed in four different areas: scenario A with 37 cells, scenario B with 45 cells, scenario C with 47 cells and a no-fly zone, and scenario D with 50 cells and a no-fly zone. These areas are farm regions of Pisa, Italy, and were imported to MATLAB®. We run both algorithms considering the same starting position. Once more, our EG-CPP overcame the original grid-based method with a percentage of improvement varying from 2.5% to 8.45%. We can trust in these simulation results considering the high accuracy of the energy model in estimating energy - over 95% of precision. We also evaluated the impact of adopting two pruning techniques to reduce the computational time of the algorithm. Our pruning techniques allow us to compute the cost of the paths step by step during the search phase. We only computed the cost introduced every time a new neighbor is explored. Then, we stored the minimum-cost path and discarded partial paths whose costs are already higher than the minimum path. Therefore, we can drastically reduce the computational time of the algorithm.

In Section 5.3, we presented comparisons with different variations of the NC-Drone. The centralized NC-Drone was compared with real-time search heuristics and overcame all approaches in the three performance metrics, QMI, SDF, and NTM. Our approach was able to reduce the QMI and SDF, performing a more uniform coverage. At the same time, the NC-Drone drastically reduced the NTM exploring the simplified version of the energy model. Our algorithm prolongs the straight segments of the path. It also minimizes the number of turning maneuvers by choosing the cell aligned with the sweeping direction when there are two or more cells with the same minimum-value. We also compared our decentralized variations NC-Drone MAX, NC-Drone AVG, and

NC-Drone MULTI among each other. NC-Drone MAX and NC-Drone AVG performed better in the QMI and the NTM metric. These approaches were able to merge the information from matrices when the vehicles were within the synchronization radius, providing more uniform coverage in terms of the interval between the visits. NC-Drone MULTI outperformed in the SDF metric. This approach only combined the information from different synchronized matrices without fusing the information into a single matrix. The approach was able to obtain a more uniform coverage considering the frequency of visits. In this way, one may consider the use of different strategies depending on the specific application requirements.

We also performed experiments comparing the cooperative strategies with centralized and decentralized NC-Drone. The cooperative strategies were Watershed (WS), Time-based (TS), Evaporation (ES), and Communication-Frequency (CFS). The WS enhanced the QMI performance metric compared to the centralized NC-Drone. The lower the QMI, the better the solution. Furthermore, the WS decreased the standard deviation in 76%, becoming the algorithm more stable. The TS and WSTS (a combination of WS and TS) also reduced the QMI metric and standard deviation. However, the WS, TS, and WSTS presented no improvements in the decentralized variations nor the SDF metric. The ES presented the best results compared with the other cooperative strategies, improving the centralized and the decentralized variations of the NC-Drone. The ES executed in intervals of 100 cycles reduced the QMI metric by 10% compared to the original centralized algorithm. Moreover, the ES also drastically reduced the standard deviation of the algorithm by 75%. The CFS with a communication interval of 50 cycles was able to overcome the centralized NC-Drone in the QMI metric. However, as the interval between synchronizations doubles, metric performance falls between 7% and 8%. The NTM increases compared to centralized NC-Drone. Considering only the CFS strategies, the NTM continues to drop as the interval of synchronization grows from CFS-100 to CFS-1000. There is a correlation between the QMI and the NTM, meaning that whenever QMI values grow, NTM tends to fall, and vice versa. Finally, we combined the CFS with the previously presented cooperative strategies. The combination of all strategies together (CFS-WSTS-ES-Time-50) presented better results with a percentage of improvement of 4.5% over the centralized NC-Drone.

6 CONCLUSION

This work proposed energy-aware coverage path planning solutions for unmanned aerial vehicles. All approaches explored an improved version of the energy model previously proposed by DI FRANCO; BUTTAZZO (2016) to generate energy-efficient coverage trajectories. The original energy model (OEM) was obtained through empirical experiments. The authors performed real flights under different circumstances (climbing, descending, rotating, flying at constant speed) to analyze the energy consumption in the function of the distance and the speed. Thanks to the real flights, the authors were able to incorporate the drag force in the model. They split every straight segment of the path in three phases - acceleration, constant speed, and deceleration. For each turning maneuver, OEM considered that the UAV decelerates from the current speed to zero, performs the turn, and then accelerates from zero to the next speed. In this way, the OEM was able to find the minimum amount of energy for different straight segments of the path by adopting optimal speeds. However, the OEM is suitable only for the E-BF.

In this work, we improved the accuracy of the OEM to deal with different coverage paths, such as the ones generated by E-Spiral. We performed several real flight experiments varying the flight speed and the turning angles. Thanks to these experiments, we discovered that the UAV reduces only a percentage of its flight speed depending on the entrance speed and the turning angle. Therefore, the improved energy model (IEM) is capable of correctly estimate the energy for any coverage path. Furthermore, it can also be employed as a cost function to generate energy-aware coverage paths.

The proposed energy-aware coverage path planning approaches include a flight pattern (E-Spiral), a complete algorithm (EG-CPP), and a pheromone-based heuristic (NC-Drone). Our solutions deal with regular and irregular-shaped areas of interest containing full and partial information. We compared the proposed algorithms with state-of-the-art strategies through simulations on MATLAB® and real flights using an IRIS quadcopter. The simulations are trustworthy. The improved energy model presented high accuracy in estimating the energy spent by the UAV during the missions. We stated the simulation precision by comparing the energy estimated by the energy model with the log information obtained from real flights.

First, we proposed the E-Spiral, an energy-aware spiral coverage path planning algorithm. It performs missions over regular-shaped areas consisting of concave and convex polygons. The flight pattern starts at the vertices of the polygon. Then, it slowly decreases its radius towards the center of the area. The approach considers specific requirement applications, such as overlapping and image resolution, to guarantee a complete area mapping in photogrammetric sensing applications.

We presented a comparison between our E-Spiral and the E-BF, an energy-aware back-and-forth algorithm proposed by DI FRANCO; BUTTAZZO (2016). We performed a wide range of simulations exploring 3750 polygonal areas with different characteristics, such as the number of vertices, irregularity, and size of the area. Results point out that E-Spiral outperforms E-BF in all cases. The larger the number of vertices, the better the performance improvement of E-Spiral over E-BF. Rounded-shaped areas with smoother angles favor our E-Spiral algorithm. In such areas, the percentage of improvement was about 16.1%. Our solution was able to maintain the optimal speed for more extended periods, minimizing the total energy consumption. The different levels of irregularity maintain the percentage of improvement of E-Spiral over E-BF stable in around 13%. Considering the size of the area, the higher the radius, the lower is the percentage of improvement. In larger areas, the E-BF was also able to maintain the optimal speed for more extended periods thanks to the long straight segments. Nevertheless, our algorithm still presented an improvement of 10.37% in such scenarios. We also performed real flights with both algorithms in two different areas of interest, a polygonal and a rectangular one. Both strategies presented similar results in the rectangular area, but our E-Spiral still slightly overcomes the E-BF. In the polygonal area, our algorithm reduced the mission execution time and the total energy consumption by 9% and 7.7%, respectively.

Next, we proposed an energy-aware grid-based coverage path planning algorithm called EG-CPP. The complete algorithm generates coverage paths for mapping missions over irregular-shaped areas containing no-fly zones. First, a Wavefront algorithm marks the neighborhood adjacency of the cells. Then, a DLS is performed over a grid-discretized area through approximate cellular decomposition. Our solution improves an existing grid-based method by replacing its original cost function based on the sum of angles to an energy-aware cost function based on the IEM. As a further contribution, two pruning techniques improved the performance of the algorithm. This improvement speeds up the computational time of the algorithm up to 99%, allowing to explore all the different starting positions in the workspace and saving even more energy.

We presented a comparison between the EG-CPP and the original grid-based approach to validate the proposed method and verify the accuracy of the improved energy model. We considered a real irregular-shaped area discretized into an 8 x 10 grid. The area contains 47 valid cells and an inner no-fly zone. We run both algorithms considering every valid cell as a potential starting position for the coverage path. Then, we selected the minimum-cost path generated by each algorithm. The EG-CPP using the energy-aware cost function (E-F) considers the cell (4,2) as the ideal initial position. The grid-based approach with the sum of the angles cost function (O-F) assumes the cell (6,1) as the best choice for starting the path. Therefore, we performed four real flights with the coverage paths generated by both algorithms with the different starting positions. In particular, the E-F allowed saving 17% of the energy in comparison with the paths generated by the O-F. The minimization of the number of turning maneuvers is adopted by several authors in the literature seeking to minimize the total energy consumption indirectly. Thanks to our experiments, we were able to show that the sum of angles is not a reliable cost function. It is possible to generate more energy-efficient coverage paths exploring an accurate energy model. The energy model considers different relevant aspects of energy consumption of a UAV, such as vehicle dynamics, kinematic energy, and optimal speed.

Additional simulation experiments were also performed on MATLAB®. We import four different areas of interest based on farm regions of the city of Pisa, Italy. They consist of irregular-shaped scenarios with varying size and complexity. Once more, the proposed EG-CPP overcomes the original grid-based method, presenting an energy improvement percentage from 2.5% to 8.45%, depending on the adopted area. Finally, we analyzed the impact of the pruning techniques in the computational time of the algorithms. We explored a simple rectangular area varying from 16 to 48 cells in steps of 4. We run three versions of the algorithm: the original grid-based method (Alg. A), the original grid-based method, but with the two pruning techniques (Alg. B), and the EG-CPP with the novel energy-aware cost function and the two pruning techniques (Alg. C). The optimization process drastically reduced computational time. Alg. B and Alg. C obtain a percentage of improvement over Alg. A varying from 44% to 99.9% and from 31% to 99% as the number of cells increases, respectively. Alg. C is slightly more expensive than Alg. B due to the complexity of the energy-aware cost function. However, Alg. C still takes only 35 seconds to generate a complete coverage path in a scenario with 36 cells, versus 3 hours of Alg. A. The simulation and the real flights stated the effectiveness of the proposed approach in saving energy. Furthermore, the pruning techniques were able to speed up the computational time of the algorithm without compromising the quality of the final solution.

Finally, we proposed an energy-aware pheromone-based solution for patrolling missions called NC-Drone. Our solution extends the original Node Counting (NC) algorithm. It adopts a simplified version of the energy model, where we prolong straight segments of the path and minimize the number of turning maneuvers to save energy. NC-Drone explores the amount of pheromone in the neighbor cells to decide the next move every step. When two or more cells have the same minimum-value, our algorithm checks if one of these cells is aligned with the sweeping direction. In case affirmative, the UAV moves to this cell, maintaining the same direction. Otherwise, it randomly selects another cell and performs a turning maneuver. The main goal here is to keep the UAV following a straight path as long as it is possible and perform unpredictable turns once and a while. This unexpected behavior is desirable for UAVs performing surveillance and patrolling missions. Since we kept the speed constant during the missions, we do not consider other relevant elements, such as the traveled distance and the optimal speed, to save energy.

We developed two types of NC-Drone, centralized and decentralized. The centralized algorithm explores the reading and writing pheromone-scheme in the cells. It is usually adopted with the assistance of a ground control station in real missions. We compared the centralized NC-Drone with previously proposed real-time search methods, such as Node Counting (NC), Learning Real-Time A* (LRTA*), Thrun's Value-Update Rule (TVUR), and Wagner's Value-Update Rule. Our centralized version overcomes all heuristics in the three performance metrics, Quadratic Mean of the Intervals (QMI), Standard Deviation of the Frequencies (SDF), and the Number of 90^o Turning Maneuvers (NTM). NC-Drone reduces the QMI and the SDF metrics, which leads to a more uniform distribution of visits, both spatial and temporal. At the same time, our algorithm also drastically reduced the NTM almost four times compared to the original heuristics. The decentralized algorithm explores a matrix-representation of the scenario to store the visited cells in the vehicle's memory. When two UAVs are within a communication radius, they can share the matrices and synchronize the information. We explore three types of synchronization schemes, NC-Drone MAX, NC-Drone AVG, and NC-Drone MULTI. The first two approaches overcome the NC-Drone MULTI in the QMI and the NTM metric, while the NC-Drone MULTI presents the best results in the SDF metric.

Moreover, we also proposed cooperative strategies to improve the performance of the NC-Drone algorithm, such as Watershed (WS), Time-based (TS), Evaporation (ES), and Communication-Frequency (CFS). The WS forms clusters of less-visited cells. The ES decreases the amount of pheromone of unvisited cells. Both strategies aim at attracting UAVs to these needed areas to balance the coverage. The TS helps to decide between cells with the same minimum-value, opting by the cell less recently visited. The CFS explores the trade-off between communication and performance, where UAVs have a full-range communication and exchange information from time to time. We performed additional experiments to highlight the improvements of the cooperative strategies in the centralized and decentralized NC-Drone. WS, TS, and the combination of both strategies were able to enhance the performance of the QMI metric in the centralized NC-Drone. ES presented the best results considering the cooperative strategies. We tested different intervals to execute the evaporation, such as 100, 250, 500, 1k cycles. The Centralized-ES-100 improves the QMI of the original centralized approach by 10%. ES was also able to improve the QMI performance of NC-Drone MULTI and NC-Drone MAX. Considering the CFS, we tested different intervals to execute the communication between the UAVs, such as 50, 100, 250, 500, and 1k. Since communication presents an additional cost to the NC-Drone, we investigated the impact of different communication ranges empirically selected to evaluate the performance of the CFS. CFS-50 presents the best results regarding the QMI metric considering this strategy. It obtains an outcome equivalent to the one presented by the original centralized, even sharing the matrices only at every 50 cycles. Finally, we combine the CFS with all cooperative strategies, where the CFS-WSTS-ES-Time-50 presented the best results in the QMI metric. Our proposed strategies were able to improve only the performance of the QMI metric, providing a more uniform distribution in the interval of visits. We stated that there is a correlation between the QMI and the NTM metric, where it is necessary to perform a higher number of turning maneuvers to achieve more uniform coverage. Therefore, to attend the application requirement of the patrolling problem, we need to spend more energy to perform the mission.

Table 23 – Proposed Energy-aware Coverage Path Planning Approaches for Unmanned Aerial Vehicles

	E-Spiral	**EG-CPP**	**NC-Drone**
Algorithm	Flight Pattern	Complete Algorithm	Pheromone-based
Area of Interest	Regular-shaped	Irregular-shaped	Regular-shaped
Area Format	Squares, Rectangles, and Convex Polygons	Concave and Convex Polygons with NFZ	Grid-discretized Scenario
Decomposition	No decomposition	Approximate Cellular Decomposition	Approximate Cellular Decomposition
Information Availability	Full Information	Full Information	Partial Information
Type of Coverage	Offline	Offline	Online

Table 23 presents a summarization of the three energy-aware proposed approaches. The core of the proposed approaches consists of generating coverage paths with prolonged straight segments combined with wider angles to keep the optimal speed as long as it is possible. In this way, it is possible to minimize the total energy consumption. Different application requirements may also influence energy consumption, such as flight altitude, overlapping, and the resolution of taken pictures. For instance, in photogrammetric sensing applications, it may be necessary to perform flights at lower altitudes to obtain high-quality pictures. These flights prolong the path length and, consequently, the energy spent. Both E-Spiral and EG-CPP deal with these application requirements and are suitable for this type of mission. Another application requirement is the uniform visitation in patrolling missions. In such scenarios, the UAVs perform online coverage, gathering information while exploring the area. Our NC-Drone can handle this type of mission, providing a balanced distribution in the frequency and the interval of visits. Furthermore, our heuristic adopts a simplified version of the energy model to save energy.

The major limitation of adopting this energy model for the coverage strategies is the necessary initial data gathering through real flights to work with different types of UAVs and controllers. It is possible to use the same polynomial equations proposed in the energy model. However, it is necessary to identify the optimal speeds and the different periods of acceleration and deceleration for the UAV. Furthermore, we only performed flights with pleasant weather without external disturbances like wind gusts.

As future work, there are several possibilities to be explored. First, a generic energy model for any UAV can be developed based on the parameters and characteristics of the aerial vehicle, such as propellers, rotors, and payload. The energy can be calculated using a closed-form mathematical energy model based on the dynamic of the movement, the principles of superposition, and energy conservation. Moreover, one can consider the use of the closed-form energy model to generate energy-efficient coverage planning strategies for different types of UAVs.

Second, the coverage path planning problem can be explored in larger areas of interest by using multiple aerial vehicles. The area of interest may be split into subareas through cellular decomposition. The resulting subareas can be assigned to each UAV to perform the coverage. Different coverage strategies may be employed, depending on the format of the subarea. A process of coordination between the fleet of UAVs can also be adopted to improve the performance and avoid collisions between the UAVs.

Finally, the coverage path planning can be performed in areas of interest containing fixed and mobile recharging stations. In such scenarios, the UAVs perform the coverage and return to the nearest recharging station from time to time to recharge or replace the battery. The main contribution is the prolongation of the mission execution time due to multiple recharges, preventing the vehicle from battery exhaustion and unexpected crashes. The UAV must continuously monitor the energy spent during the mission and verify the remaining energy in the battery. At the same time, the UAV must estimate the necessary energy to return to one of the recharging stations safely. When the UAV detects that the remaining energy is at a critical level, it must interrupt the coverage task and come back to one of the stations with a safety margin. The battery must remain with at least 30% of energy at the end of each mission. Solutions for this problem include several features, such as the adoption of an accurate energy model, the generation of coverage trajectories, and the localization of recharging stations.

REFERENCES

ACEVEDO, J. J.; ARRUE, B. C.; MAZA, I.; OLLERO, A. Cooperative Large Area Surveillance with a Team of Aerial Mobile Robots for Long Endurance Missions. **Journal of Intelligent & Robotic Systems**, Netherlands, v.70, n.1-4, p.329–345, 2013.

ACEVEDO, J. J.; ARRUE, B. C.; MAZA, I.; OLLERO, A. Distributed Approach for Coverage and Patrolling Missions with a Team of Heterogeneous Aerial Robots under Communication Constraints. **International Journal of Advanced Robotic Systems**, [S.l.], v.10, n.1, p.28, 2013.

ACEVEDO, J. J. et al. One-to-one coordination algorithm for decentralized area partition in surveillance missions with a team of aerial robots. **Journal of Intelligent & Robotic Systems**, Netherlands, v.74, n.1-2, p.269–285, 2014.

ADAMS, D. **Best drone 2019**: the top drones for stunning 4K video, pin-sharp photos and aerial selfies. (accessed on 19 June 2019), Available online: `https://www.t3.com/features/best-drone`.

KHATIB, O.; KUMAR, V.; RUS, D. (Ed.). **Experimental Robotics**: The 10th International Symposium on Experimental Robotics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p.491–500.

ALBANI, D.; NARDI, D.; TRIANNI, V. Field coverage and weed mapping by UAV swarms. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2017., 2017, Vancouver, BC, Canada. **Anais. . .** IEEE, 2017. p.4319–4325.

ALVARENGA, J.; VITZILAIOS, N. I.; VALAVANIS, K. P.; RUTHERFORD, M. J. Survey of unmanned helicopter model-based navigation and control techniques. **Journal of Intelligent & Robotic Systems**, Netherlands, v.80, n.1, p.87–138, 2015.

ANDERSEN, H. L. **Path Planning for Search and Rescue Mission using Multicopters**. 2014. Dissertação (Mestrado em Ciência da Computação) — , Norway.

ARAUJO, J.; SUJIT, P.; SOUSA, J. B. Multiple UAV area decomposition and coverage. In: IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR SECURITY AND DEFENSE APPLICATIONS (CISDA), 2013., 2013, Singapore. **Anais. . .** IEEE, 2013. p.30–37.

ARDUPILOT. **ArduPilot**. (accessed on 20 March 2018), Available online: `http://ardupilot.org`.

ARTEMENKO, O.; DOMINIC, O. J.; ANDRYEYEV, O.; MITSCHELE-THIEL, A. Energy-aware trajectory planning for the localization of mobile devices using an unmanned aerial vehicle. In: INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND NETWORKS (ICCCN), 2016., 2016, Waikoloa, HI, USA. **Anais. . .** IEEE, 2016. p.1–9.

BALAMPANIS, F.; MAZA, I.; OLLERO, A. Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2016., 2016, Arlington, VA, USA. **Anais. . .** IEEE, 2016. p.275–283.

BALAMPANIS, F.; MAZA, I.; OLLERO, A. Area partition for coastal regions with multiple UAS. **Journal of Intelligent & Robotic Systems**, Netherlands, v.88, n.2-4, p.751–766, 2017.

BALAMPANIS, F.; MAZA, I.; OLLERO, A. Coastal areas division and coverage with multiple UAVs for remote sensing. **Sensors**, [S.l.], v.17, n.4, p.808, 2017.

BALAMPANIS, F.; MAZA, I.; OLLERO, A. Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2017., 2017, Miami, FL, USA. **Anais. . .** IEEE, 2017. p.617–624.

BANDEIRA, T. W.; COUTINHO, W. P.; BRITO, A. V.; SUBRAMANIAN, A. Analysis of path planning algorithms based on travelling salesman problem embedded in UAVs. In: BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC), 2015., 2015, Foz do Iguaçu, Brazil. **Anais. . .** IEEE, 2015. p.70–75.

BARRIENTOS, A. et al. Aerial Remote Sensing in Agriculture: A Practical Approach to Area Coverage and Path Planning for Fleets of Mini Aerial Robots. **J. Field Robot.**, [S.l.], v.28, n.5, p.667–689, 9 2011.

BASILICO, N.; CARPIN, S. Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2015., 2015, Hamburg, Germany. **Anais. . .** IEEE, 2015. p.610–615.

BEUCHER, S.; MEYER, F. The morphological approach to segmentation: the watershed transformation. **Optical Engineering-New York-Marcel Dekker Incorporated-**, [S.l.], v.34, p.433–433, 1992.

BOUZID, Y.; BESTAOUI, Y.; SIGUERDIDJANE, H. Quadrotor-UAV optimal coverage path planning in cluttered environment with a limited onboard energy. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2017., 2017, Vancouver, BC, Canada. **Anais. . .** IEEE, 2017. p.979–984.

CABREIRA, T. M.; DI FRANCO, C.; FERREIRA, P. R.; BUTTAZZO, G. C. Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications. **IEEE Robotics and Automation Letters**, [S.l.], v.3, n.4, p.3662–3668, 2018.

CABREIRA, T. M.; FERREIRA, P. R. Terrain Coverage with UAVs: Real-Time Search and Geometric Approaches Applied to an Abstract Model of Random Events. In: XIII LATIN AMERICAN ROBOTICS SYMPOSIUM AND IV BRAZILIAN ROBOTICS SYMPOSIUM (LARS/SBR), 2016., 2016, Recife, Brazil. **Anais. . .** IEEE, 2016. p.151–156.

CABREIRA, T. M.; KAPPEL, K. S.; FERREIRA, P. R.; BRISOLARA, L. B. de. An Energy-Aware Real-Time Search Approach for Cooperative Patrolling Missions with Multi-UAVs. In: LATIN AMERICAN ROBOTIC SYMPOSIUM, 2018 BRAZILIAN SYMPOSIUM ON ROBOTICS (SBR) AND 2018 WORKSHOP ON ROBOTICS IN EDUCATION (WRE), 2018., 2018b, João Pessoa, Brazil. **Anais. . .** IEEE, 2018b. p.254–259.

CANNATA, G.; SGORBISSA, A. A minimalist algorithm for multirobot continuous coverage. **IEEE Transactions on Robotics**, [S.l.], v.27, n.2, p.297–312, 2011.

CESETTI, A. et al. A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications. **Journal of Intelligent & Robotic Systems**, Netherlands, v.61, n.1-4, p.157–168, 2011.

CHANG, W. et al. Development of a power line inspection robot with hybrid operation modes. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2017., 2017, Vancouver, BC, Canada. **Anais. . .** IEEE, 2017. p.973–978.

CHENG, C.-T.; FALLAHI, K.; LEUNG, H.; CHI, K. T. Cooperative path planner for UAVs using ACO algorithm with Gaussian distribution functions. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 2009., 2009, Taipei, Taiwan. **Anais. . .** IEEE, 2009. p.173–176.

CHEVALEYRE, Y. Theoretical analysis of the multi-agent patrolling problem. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON INTELLIGENT AGENT TECH-

NOLOGY, 2004.(IAT 2004)., 2004, Beijing, China. **Proceedings...** IEEE, 2004. p.302–308.

CHOSET, H. Coverage for robotics–A survey of recent results. **Annals of mathematics and artificial intelligence**, [S.l.], v.31, n.1-4, p.113–126, 2001.

CHOSET, H.; PIGNON, P. Coverage path planning: The boustrophedon cellular decomposition. In: FIELD AND SERVICE ROBOTICS, 1998, London, United Kingdom. **Anais...** Springer, 1998. p.203–209.

COOMBES, M.; CHEN, W.-H.; LIU, C. Boustrophedon coverage path planning for UAV aerial surveys in wind. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2017., 2017, Miami, FL, USA. **Anais...** IEEE, 2017. p.1563–1571.

COOMBES, M.; FLETCHER, T.; CHEN, W.-H.; LIU, C. Optimal polygon decomposition for UAV survey coverage path planning in wind. **Sensors**, [S.l.], v.18, n.7, p.2132, 2018.

DARRAH, M.; TRUJILLO, M. M.; SPERANSKY, K.; WATHEN, M. Optimized 3D mapping of a large area with structures using multiple multirotors. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2017., 2017, Miami, FL, USA. **Anais...** IEEE, 2017. p.716–722.

DI FRANCO, C.; BUTTAZZO, G. Energy-aware coverage path planning of UAVs. In: IEEE INTERNATIONAL CONFERENCE ON AUTONOMOUS ROBOT SYSTEMS AND COMPETITIONS, 2015., 2015, Vila Real, Portugal. **Anais...** IEEE, 2015. p.111–117.

DI FRANCO, C.; BUTTAZZO, G. Coverage Path Planning for UAVs Photogrammetry with Energy and Resolution Constraints. **Journal of Intelligent & Robotic Systems**, Netherlands, p.1–18, 2016.

DIGI. **Digi XBee3 Zigbee 3**. EUA: Digi International Inc., 2019. [Online; accessed April-02-2019], `https://www.digi.com/products/embedded-systems/rf-modules/2-4-ghz-modules/xbee3-zigbee-3`.

DIGI. **Drone Technologies Disrupting Industries, Saving Lives**. EUA: Digi International Inc., 2019. [Online; accessed April-02-2019], `https://www.digi.com/customersuccesses/draganfly`.

FAMOURI, M.; HAMZEH, A. A new complete heuristic approach for ant rendezvous problem. In: INTERNATIONAL CONFERENCE ON COMPUTER AND KNOWLEDGE ENGINEERING (ICCKE), 2014., 2014, Mashhad, Iran. **Anais...** IEEE, 2014. p.364–369.

FLOREANO, D.; MATTIUSSI, C. **Bio-Inspired Artificial Intelligence**: Theories, Methods, and Technologies. USA: MIT Press, 2008.

FORSMO, E. J.; GRØTLI, E. I.; FOSSEN, T. I.; JOHANSEN, T. A. Optimal search mission with unmanned aerial vehicles using mixed integer linear programming. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2013., 2013, Atlanta, GA, USA. **Anais...** IEEE, 2013. p.253–259.

GALCERAN, E.; CARRERAS, M. A Survey on Coverage Path Planning for Robotics. **Robot. Auton. Syst.**, Amsterdam, Holland, v.61, n.12, p.1258–1276, 12 2013.

GUERRERO, J. A.; BESTAOUI, Y. UAV path planning for structure inspection in windy environments. **Journal of Intelligent & Robotic Systems**, Netherlands, v.69, n.1-4, p.297–311, 2013.

HAYAT, S.; YANMAZ, E.; BROWN, T. X.; BETTSTETTER, C. Multi-objective UAV path planning for search and rescue. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2017., 2017, Singapore. **Anais...** IEEE, 2017. p.5569–5574.

HUANG, H.; HOFFMANN, G. M.; WASLANDER, S. L.; TOMLIN, C. J. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2009., 2009, Kobe, Japan. **Anais...** IEEE, 2009. p.3277–3282.

HUANG, W. H. Optimal line-sweep-based decompositions for coverage algorithms. In: ICRA. IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001., 2001, Seoul, South Korea. **Proceedings...** IEEE, 2001. v.1, p.27–32.

JIAO, Y.-S.; WANG, X.-M.; CHEN, H.; LI, Y. Research on the coverage path planning of UAVs for polygon areas. In: IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS, 2010., 2010, Taichung, Taiwan. **Anais...** IEEE, 2010. p.1467–1472.

JOHNSON, G. L. **Wind Energy Systems**. [S.l.]: Prentice-Hall Englewood Cliffs (NJ), 1985.

JOHNSON, W. **Rotorcraft aeromechanics**. United Kingdom: Cambridge University Press, 2013. v.36.

KAKAES, K. et al. Drones and Aerial Observation: New Technologies for Property Rights, Human Rights, and Global Development: A Primer. **New America**, Washington, DC, USA, p.6–103, 2015.

KALLMANN, M.; BIERI, H.; THALMANN, D. Fully dynamic constrained delaunay triangulations. In: **Geometric modeling for scientific visualization**. Berlin, Germany: Springer, 2004. p.241–257.

KANELLAKIS, C.; NIKOLAKOPOULOS, G. Survey on computer vision for UAVs: Current developments and trends. **Journal of Intelligent & Robotic Systems**, Netherlands, v.87, n.1, p.141–168, 2017.

KHAN, A.; YANMAZ, E.; RINNER, B. Information merging in multi-UAV cooperative search. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2014., 2014, Hong Kong, China. **Anais. . .** IEEE, 2014. p.3122–3129.

KOENIG, S.; LIU, Y. Terrain Coverage with Ant Robots: A Simulation Study. In: FIFTH INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2001, Montreal, Quebec, Canada. **Proceedings. . .** ACM, 2001. p.600–607.

KOENIG, S.; SIMMONS, R. G. Easy and hard testbeds for real-time search algorithms. In: AAAI/IAAI, VOL. 1, 1996, Portland, Oregon, USA. **Anais. . .** ACM, 1996. p.279–285.

KORF, R. E. Real-time heuristic search. **Artificial intelligence**, [S.l.], v.42, n.2-3, p.189–211, 1990.

KUIPER, E.; NADJM-TEHRANI, S. Mobility models for UAV group reconnaissance applications. In: INTERNATIONAL CONFERENCE ON WIRELESS AND MOBILE COMMUNICATIONS (ICWMC'06), 2006., 2006, Bucharest, Romania. **Anais. . .** IEEE, 2006. p.33–33.

LAVALLE, S. M. **Planning algorithms**. United Kingdom: Cambridge University Press, 2006.

LEVCOPOULOS, C.; KRZNARIC, D. Quasi-Greedy Triangulations Approximating the Minimum Weight Triangulation. In: SODA, 1996, Atlanta, GA, USA. **Anais. . .** ACM, 1996. p.392–401.

LI, D.; WANG, X.; SUN, T. Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. **Electronics Letters**, [S.l.], v.52, n.9, p.699–701, 2016.

LI, Y.; CHEN, H.; ER, M. J.; WANG, X. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. **Mechatronics**, [S.l.], v.21, n.5, p.876 – 885, 2011. Special Issue on Development of Autonomous Unmanned Aerial Vehicles.

LIM, S.; BANG, H. Waypoint guidance of cooperative UAVs for intelligence, surveillance, and reconnaissance. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL AND

AUTOMATION, 2009., 2009, Christchurch, New Zealand. **Anais...** IEEE, 2009. p.291–296.

LIM, S.; BANG, H. Waypoint planning algorithm using cost functions for surveillance. **International Journal Aeronautical and Space Sciences**, [S.l.], v.11, n.2, p.136–144, 2010.

LOTTES, P. et al. UAV-based crop and weed classification for smart farming. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2017., 2017, Singapore, Singapore. **Anais...** 2017, 2017. p.3024–3031.

MAZA, I. et al. Experimental results in multi-UAV coordination for disaster management and civil security applications. **Journal of Intelligent & Robotic Systems**, Netherlands, v.61, n.1-4, p.563–585, 2011.

MAZA, I.; OLLERO, A. Multiples UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: DISTRIBUTED AUTONOMOUS ROBOTIC SYSTEMS 6, 2007. **Anais...** Springer Japan, 2007. p.221–230.

NAM, L.; HUANG, L.; LI, X. J.; XU, J. An approach for coverage path planning for UAVs. In: IEEE 14TH INTERNATIONAL WORKSHOP ON ADVANCED MOTION CONTROL (AMC), 2016., 2016, Auckland, New Zealand. **Anais...** IEEE, 2016. p.411–416.

NATTERO, C.; RECCHIUTO, C.; SGORBISSA, A.; WANDERLINGH, F. Coverage algorithms for search and rescue with uav drones. In: WORKSHOP OF THE XIII AIIA SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 2014, Pisa, Italy. **Anais...** Italian Association for Artificial Intelligence, 2014.

OBORNE, M. **Mission Planner - Ground Station**. (accessed on 15 June 2017), Available online: `http://planner.ardupilot.com`.

ÖST, G. **Search path generation with UAV applications using approximate convex decomposition**. 2012. Dissertação (Mestrado em Ciência da Computação) — , Sweden.

PARADZIK, M.; İNCE, G. Multi-agent search strategy based on digital pheromones for UAVs. In: SIGNAL PROCESSING AND COMMUNICATION APPLICATION CONFERENCE (SIU), 2016., 2016, Zonguldak, Turkey. **Anais...** IEEE, 2016. p.233–236.

PFEIFER, R.; LUNGARELLA, M.; IIDA, F. Self-organization, embodiment, and biologically inspired robotics. **science**, [S.l.], v.318, n.5853, p.1088–1093, 2007.

PHAM, H. X.; LA, H. M.; FEIL-SEIFER, D.; DEANS, M. A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2017., 2017, Vancouver, BC, Canada. **Anais. . .** IEEE, 2017. p.6648–6653.

PIAGGIO, M.; SGORBISSA, A.; ZACCARIA, R. Programming real-time distributed multiple robotic systems. In: ROBOT SOCCER WORLD CUP, 1999. **Anais. . .** Springer, 1999. p.412–423.

PIRZADEH, A.; SNYDER, W. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1990, Cincinnati, OH, USA. **Proceedings. . .** IEEE, 1990. p.2113–2119.

POPOVIĆ, M. et al. Online informative path planning for active classification using UAVs. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2017., 2017, Singapore. **Anais. . .** IEEE, 2017. p.5753–5758.

POPOVIĆ, M. et al. Multiresolution mapping and informative path planning for UAV-based terrain monitoring. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2017., 2017b, Vancouver, Canada. **Anais. . .** IEEE, 2017b. p.1382–1388.

QUIGLEY, M. et al. ROS: an open-source Robot Operating System. In: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE, 2009, Kobe, Japan. **Anais. . .** IEEE, 2009. v.3, p.5.

RAMASAMY, M.; GHOSE, D. Learning-based preferential surveillance algorithm for persistent surveillance by unmanned aerial vehicles. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2016., 2016, Arlington, VA, USA. **Anais. . .** IEEE, 2016. p.1032–1040.

RAMASAMY, M.; GHOSE, D. A heuristic learning algorithm for preferential area surveillance by unmanned aerial vehicles. **Journal of Intelligent & Robotic Systems**, Netherlands, v.88, n.2-4, p.655–681, 2017.

RENZAGLIA, A.; REYMANN, C.; LACROIX, S. Monitoring the evolution of clouds with UAVs. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2016., 2016, Stockholm, Sweden. **Anais. . .** IEEE, 2016. p.278–283.

RINEAU, L. **Cgal - 2d conforming triangulations and meshes - 2.5 optimization of meshes with lloyd.** `https://doc.cgal.org/latest/Mesh{_}2/index.html{#}secMesh{_}2{_}optimization`.

ROHMER, E.; SINGH, S. P.; FREESE, M. V-REP: A versatile and scalable robot simulation framework. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013., 2013, Tokyo, Japan. **Anais. . .** IEEE, 2013. p.1321–1326.

ROSALIE, M.; DANOY, G.; CHAUMETTE, S.; BOUVRY, P. From random process to chaotic behavior in swarms of UAVs. In: ACM SYMPOSIUM ON DEVELOPMENT AND ANALYSIS OF INTELLIGENT VEHICULAR NETWORKS AND APPLICATIONS, 6., 2016, Malta, Malta. **Proceedings. . .** ACM, 2016. p.9–15.

ROSALIE, M. et al. Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2017., 2017, Miami, FL, USA. **Anais. . .** IEEE, 2017. p.1392–1397.

SADAT, S. A.; WAWERLA, J.; VAUGHAN, R. Fractal trajectories for online non-uniform aerial coverage. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2015., 2015, Seattle, WA, USA. **Anais. . .** IEEE, 2015. p.2971–2976.

SADAT, S. A.; WAWERLA, J.; VAUGHAN, R. T. Recursive non-uniform coverage of unknown terrains for uavs. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2014., 2014, Chicago, Illinois, USA. **Anais. . .** IEEE, 2014. p.1742–1747.

SAEED, A. S. et al. A review on the platform design, dynamic modeling and control of hybrid UAVs. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2015., 2015, Denver, CO, USA. **Anais. . .** IEEE, 2015. p.806–815.

SAMPAIO, P.; SOUSA, R.; ROCHA, A. New Patrolling Strategies with Short-Range Perception. In: XIII LATIN AMERICAN ROBOTICS SYMPOSIUM AND IV BRAZILIAN ROBOTICS SYMPOSIUM, 2016., 2016, Recife, Brazil. **Proceedings. . .** IEEE, 2016.

SANTAMARIA, E.; SEGOR, F.; TCHOUCHENKOV, I. Rapid aerial mapping with multiple heterogeneous unmanned vehicles. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 10., 2013, Baden-Baden, Germany. **Proceedings. . .** ISCRAM, 2013.

SEGOR, F.; BÜRKLE, A.; PARTMANN, T.; SCHÖNBEIN, R. Mobile ground control station for local surveillance. In: FIFTH INTERNATIONAL CONFERENCE ON SYSTEMS, 2010., 2010, Menuires, France. **Anais. . .** IEEE, 2010. p.152–157.

STALMAKOU, A. **UAV/UAS path planning for ice management information gathering**. 2011. Dissertação (Mestrado em Ciência da Computação) — , Norway.

TORRES, M.; PELTA, D. A.; VERDEGAY, J. L.; TORRES, J. C. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. **Expert Systems with Applications**, [S.l.], v.55, p.441 – 451, 2016.

TRUJILLO, M. M. et al. Optimized flight path for 3D mapping of an area with structures using a multirotor. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2016., 2016, Washington, USA. **Anais...** IEEE, 2016. p.905–910.

VALENTE, J.; DEL CERRO, J.; BARRIENTOS, A.; SANZ, D. Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. **Computers and electronics in agriculture**, [S.l.], v.99, p.153–159, 2013.

VALENTE, J. et al. Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. **Precision Agriculture**, USA, v.14, n.1, p.115–132, 2013.

VINCENT, P.; RUBIN, I. A Framework and Analysis for Cooperative Search Using UAV Swarms. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2004., 2004, New York, NY, USA. **Proceedings...** ACM, 2004. p.79–86. (SAC '04).

WARE, J.; ROY, N. An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2016., 2016, Stockholm, Sweden. **Anais...** IEEE, 2016. p.1507–1514.

WATTS, A. C.; AMBROSIA, V. G.; HINKLEY, E. A. Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use. **Remote Sensing**, Switzerland, v.4, n.6, p.1671–1692, Jun 2012.

XU, A.; VIRIYASUTHEE, C.; REKLEITIS, I. Optimal complete terrain coverage using an unmanned aerial vehicle. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON, 2011, Shanghai, China. **Anais...** IEEE, 2011. p.2513–2519.

XU, A.; VIRIYASUTHEE, C.; REKLEITIS, I. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. **Autonomous Robots**, USA, v.36, n.4, p.365–381, 2014.

ZELENKA, J.; KASANICKÝ, T. Insect Pheromone Strategy for the Robots Coordination. In: INDUSTRIAL AND SERVICE ROBOTICS, 2014. **Anais...** Trans Tech Publications Ltd, 2014. p.163–171. (Applied Mechanics and Materials, v.613).

ZELENKA, J.; KASANICKY, T. Outdoor UAV control and coordination system supported by biological inspired method. In: ROBOTICS IN ALPE-ADRIA-DANUBE REGION

(RAAD), 2014 23RD INTERNATIONAL CONFERENCE ON, 2014, Smolenice, Slovakia. **Anais. . .** IEEE, 2014. p.1–7.

ZELENKA, J.; KASANICKY, T. Insect pheromone strategy for the robots coordination: Reaction on loss communication. In: COMPUTATIONAL INTELLIGENCE AND INFORMATICS (CINTI), 2014 IEEE 15TH INTERNATIONAL SYMPOSIUM ON, 2014, Budapest, Hungary. **Anais. . .** IEEE, 2014. p.79–83.

APPENDIX A SUMMARIZATION OF RELATED WORKS

The revised coverage path planning approaches are summarized in the following tables. Table 24 summarizes the approaches considering areas of interest with no cellular decomposition technique. The table presents the CPP approach, corresponding reference, shape of the area of interest, adopted performance metrics to evaluate coverage pattern, indication of single or multiple UAVs used in the coverage, and the type of UAV: rotary-wing (RW), fixed-wing (FW), or both. The approaches considering areas of interest discretized through the exact cellular decomposition technique are summarized in Table 25. The online/offline column in the table refers to how the coverage path is obtained. In offline cases, the whole path is computed before being performed, while in online cases the path can be computed or modified during coverage.

Finally, Table 26 summarizes the CPP approaches revised considering areas of interest discretized through the approximate cellular decomposition technique considering full and partial information. While the CPP approaches with no decomposition or combined with exact cellular decomposition can be executed by rotary-wing, fixed-wing, or both types of vehicles, the CPP approaches using approximate cellular decomposition almost exclusively adopt rotary-wing UAVs. This is because the rotary-wings present maneuverability advantages when making turns in scenarios discretized into a grid. The fixed-wing has maneuverability restrictions, demanding a large space to make turns.

Table 24 – Coverage path planning approaches in areas of interest with no decomposition technique

Approach	Ref.	Shape of the area	Performance metrics	Single/ Multiple	Type
Back-and-Forth, Square, Sector Search, Barrier Patrol	ANDERSEN (2014)	Rectangular	Fixed and mobile target detection; Coverage rate	Single	RW
Back-and-Forth	COOMBES; CHEN; LIU (2017)	Polygonal	Flight time	Single	FW
Energy-aware Back-and-Forth	DI FRANCO; BUTTAZZO (2016)	Polygonal	Energy consumption	Single	RW
Three-stage Energy-aware	LI; WANG; SUN (2016)	3D Topology	Energy consumption	Single	RW
Smoothing algorithms: E-MoTA e I-MoTA	ARTEMENKO et al. (2016)	Regular Grid	Energy consumption; Mission time; Level of localization accuracy	Single	Both
Mixed Integer Linear Programming (MILP)	FORSMO et al. (2013)	Rectangular	Flight time	Multiple	FW
Circular	AHMADZADEH et al. (2008)	Rectangular	Coverage rate; time	Multiple	FW

Table 25 – Coverage path planning approaches in areas of interest with exact cellular decomposition

Approach	Ref.	Online/ Offline	Shape of the area	Performance metrics	Single/ Multiple	Type
Back-and-Forth	JIAO et al. (2010); LI et al. (2011)	Offline	Polygonal	Number of turning maneuvers	Single	Both
Back-and-Forth	TORRES et al. (2016)	Offline	Polygonal	Number of turning maneuvers; Path length	Single	RW
Back-and-Forth	XU; VIRIYASUTHEE; REKLEITIS (2011, 2014)	Offline/ Online	Irregular	Path length; Coverage time	Single	FW
Back-and-Forth and Spiral	ÖST (2012)	Offline	Polygonal	Path length	Single	FW
Back-and-Forth	MAZA; OLLERO (2007)	Offline	Polygonal	Number of turning maneuvers	Multiple	RW
Spiral	BALAMPANIS; MAZA; OLLERO (2016, 2017a,b,c)	Offline	Polygonal	Path length	Multiple	FW
Back-and-Forth (Line Formation)	VINCENT; RUBIN (2004)	Offline	Rectangular	Target detection; Search time, Number of UAVs; Information exchange	Multiple	RW
One-to-one coordination (Decentralized Technique)	ACEVEDO et al. (2014, 2013a,b)	Online	Irregular	Interval of visits; Information latency	Multiple	Both
Back-and-Forth/Zamboni (Local Priority)	ARAUJO; SUJIT; SOUSA (2013)	Offline	Polygonal	Number of turning maneuvers; uncertainty	Multiple	FW

Table 26 – Coverage path planning approaches in areas of interest with approximate cellular decomposition

Approach	Ref.	Online/ Offline	Shape of the area	Performance metrics	Single/ Multiple	Type
Gradient-based approach	VALENTE et al. (2013)	Offline	Irregular/ Regular Grid	Coverage time	Single	RW
Wavefront Algorithm and Cubic Interpolation	NAM et al. (2016)	Offline	Irregular/ Regular Grid	Path length; Number of turning maneuvers	Single	RW
Multi-RTT* Fixed Node (RRT*FN) and Genetic Algorithm (GA)	BOUZID; BESTAOUI; SIGUERDIDJANE (2017)	Offline	Regular Grid	Path length	Single	RW
Wavefront Algorithm	BARRIENTOS et al. (2011)	Offline	Irregular/ Regular Grid	Position and altitude errors; Wind disturbances; Path length; Mission, flight and configuration times	Multiple	RW
Harmony Search	VALENTE et al. (2013)	Offline	Irregular/ Regular Grid	Number of turning maneuvers	Multiple	RW
Breadth-First, Depth-First, and Shortcut Heuristic	SADAT; WAWERLA; VAUGHAN (2014)	Online	Square	Total distance of coverage	Single	RW
Hilbert space-filling curves	SADAT; WAWERLA; VAUGHAN (2015)	Online	Square	Total distance of coverage	Single	RW
Long straight-lines algorithm	SANTAMARIA; SEGOR; TCHOUCHENKOV (2013)	Offline	Irregular/ Grid related to the sensor	Total distance; Number of turns; Number of jumps between cells	Multiple	RW
Edge Counting and PatrolGRAPH*	NATTERO et al. (2014)	Online	Graph Grid	Path length; Robots distance average	Multiple	RW
Reinforced Random Walk	ALBANI; NARDI; TRIANNI (2017)	Online	Rectangular	Coverage time; Global detection efficiency	Multiple	RW
Cellular Automata	ZELENKA; KASANICKÝ (2014); ZELENKA; KASANICKY (2014a,b)	Online	Regular Grid	Exploration time with/without barriers	Multiple	RW
Waypoint planning with uncertainty	LIM; BANG (2010)	Online	Rectangular	Certainty of information points	Multiple	Both
Information merging for cooperative search	KHAN; YANMAZ; RINNER (2014)	Online	Rectangular	Target localization	Multiple	RW
Fixed-horizon with CMA-ES	POPOVIĆ et al. (2017,b)	Online	Rectangular	Entropy; Classification rate	Multiple	RW
Learning-based Preferential Surveillance Algorithm (LPSA)	RAMASAMY; GHOSE (2016, 2017)	Online	Regular Grid	Distribution of visits; Target localization; Threat avoidance	Single	Both
Back-and-Forth	PARADZIK; İNCE (2016)	Online	Regular Grid	Total distance; Coverage rate; Redundancy rate	Multiple	RW
Genetic Algorithm (GA)	TRUJILLO et al. (2016)	Offline/ Online	Polygonal/ Regular Grid	Path length	Single	RW
GA with flood fill algorithm	DARRAH et al. (2017)	Offline/ Online	Polygonal/ Regular Grid	Path length	Multiple	RW
Multi-Objective Path Planning with GA	HAYAT et al. (2017)	Offline/ Online	Rectangular	Mission Completion Time	Multiple	RW

Chaotic Ant Colony Optiomization to Coverage	ROSALIE et al. (2016, 2017)	Offline/ Online	Regular Grid	Coverage rate; Recent coverage ratio;
				Fairness (coverage distribution);
				Connectivity (UAVs distribution)
					Multiple	RW

ACO with Gaussian distribution functions	CHENG et al. (2009)	Online	3D Regular Grid	Path length and rotation angle; Inclination and area overlapping rate	Multiple	Both