Pontifícia Universidade Católica
do Rio de Janeiro

Bruno José Olivieri de Souza

Flight Coordination Approaches of UAV Squads for WSN Data Collection

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática.

Advisor: Prof. Markus Endler

Rio de Janeiro
March 2019

Pontifícia Universidade Católica
do Rio de Janeiro

Bruno José Olivieri de Souza

Flight Coordination Approaches of UAV Squads for WSN Data Collection

Thesis presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática. Approved by the undersigned Examination Committee.

Prof. Markus Endler
Advisor
Departamento de Informática – PUC-Rio

Prof.ª Noemi de La Rocque Rodriguez
Departamento de Informática – PUC-Rio

Prof. Sérgio Colcher
Departamento de Informática – PUC-Rio

Prof.ª Adria Ramos de Lyra
Departamento de Ciência da Computação – UFRRJ

Prof. Francisco José da Silva e Silva
Departamento de Informática – UFMA

Rio de Janeiro, March the 21st, 2019

Bruno José Olivieri de Souza

The author received his Bachelor degree in Computer Science from the Instituto de Computação of Universidade Federal Fluminense (UFF) in 2003. He received his Master degree in Computer Science from PUC-Rio in 2015. During his career, he participated in several projects mainly in private companies in Brazil, Portugal, and London. During his Master degree at PUC-Rio, he received the prestigious FAPERJ "Nota 10" scholarship. Currently, he works at Tribunal Superior do Trabalho.

Bibliographic data

Olivieri de Souza, Bruno José

Flight Coordination Approaches of UAV Squads for WSN Data Collection / Bruno José Olivieri de Souza; advisor: Markus Endler. – Rio de janeiro: PUC-Rio, Departamento de Informática, 2019.

v., 111 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Informática – Teses. 2. Veículos Aéreos Não Tripulados. 3. Coleta de Dados. 4. Redes de Sensores Sem Fio. 5. Algorítmos Distribuídos. 6. Enxames. I. Endler, Markus. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Dedicated to the family, in all its senses. The family is the natural and mighty reason for the prosperity of humankind.

Acknowledgments

Verba volant, scripta manent. First of all, I would like to thank my family, who always supported, helped and inspired me: my wife Carol, my parents, Ribamar and Márcia and my sisters Fernanda and Madalena, who have always wished me the best of everything. Thank you, my sister, Fernanda, who was responsible for my post-graduation enrollment decision. My dear daughter O^2, who was born at the end of this endeavor, brought more light on my path.

I would like to thank my advisor, Markus Endler, for being a patient guide and reader of my work. I would also like to thank everyone from LAC, especially Igor Vasconcelos and Marcos Roriz.

I would like to thank all professors and staff from PUC-Rio, and in particular Regina Zanon and professor Noemi Rodrigues, both of whom taught and helped me extensively during the PhD. I would also like to thank the professor, Adria Lyra, who from before the first day of the master's degree program to the last day of the doctorate was available and inspiring.

I would like to thank my personal friends who helped me either directly or indirectly through various and lonely moments regarding NP hard problems.

I would like to thank my partners in fellowship on this path, with whom I shared many long hours of studying and social life renunciation, especially Rafael Oliveira. He was my study partner for many weekends and long-distance discussions even across continents. We gave each other six years of mutual support in this journey. "*Made with ssl*!"

Abstract

Olivieri de Souza, Bruno José; Endler, Markus (Advisor). **Flight Coordination Approaches of UAV Squads for WSN Data Collection**. Rio de Janeiro, 2019. 111p. Tese de doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Wireless sensor networks (WSNs) are an important means of collecting data in a variety of situations, such as monitoring large or hazardous areas. The retrieval of WSN data can yield better results with the use of unmanned aerial vehicles (UAVs), for example, concerning the increase in the amount of collected data and decrease in the time between the collection and use of the data. In particular, disaster areas may be left without communication resources and with great residual risk to humans, at which point a WSN can be quickly launched by air to collect relevant data until other measures can be put in place. Some studies present approaches to the use of UAVs for the collection of WSN data, focusing mainly on optimizing the path to be covered by a single UAV and relying on long-range communication that is always available; these studies do not explore the possibility of using several UAVs or the limitations on the range of communication. This work describes DADCA, a distributed scalable approach capable of coordinating groups of UAVs in WSN data collection with restricted communication range and without the use of optimization techniques. The results show that the amount of data collected by DADCA is similar or superior, by up to 1%, to path optimization approaches. In the proposed approach, the delay in receiving sensor messages is up to 46% shorter than in other approaches, and the required processing onboard UAVs can reach less than 75% of those using optimization-based algorithms. The results indicate that the DADCA can match and even surpass other approaches presented, while also adding the advantages of a distributed approach.

Keywords

Unmanned Aereal Vehicles; Data Collection; Wireles Sensors Networks; Distributed Algorithms; Swarm.

Resumo

Olivieri de Souza, Bruno José; Endler, Markus. **Abordagens de Coordenação de Voo para Grupos de VANT em Coleta de Dados de WSN**. Rio de Janeiro, 2019. 111p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Redes de sensores sem fio (WSN) são uma importante alternativa na coleta de dados em diversas situações, tais como no monitoramento de grandes áreas ou áreas que apresentem perigo. A recuperação de dados de WSNs é uma importante atividade que pode obter melhores resultados com o uso de veículos aéreos não tripulados (UAV) como, por exemplo, em relação ao aumento da quantidade de dados coletados e diminuição do tempo entre a coleta dos dados e seu uso. Em particular, áreas tomadas por desastres podem ficar sem recursos de comunicação e com grande risco residual para humanos, momento no qual uma WSN pode ser rapidamente lançada por via aérea e atuar na coleta de dados relevantes até que medidas pertinentes e dedicadas possam ser colocadas em ação. Estudos apresentam abordagens no uso de UAVs para coleta dos dados de WSN, focando principalmente na otimização do caminho a ser percorrido por um único UAV e se baseando em uma comunicação de longo alcance sempre disponível, não explorando a possibilidade da utilização de diversos UAVs ou à limitação do alcance da comunicação. Neste trabalho apresentamos o DADCA, uma abordagem distribuída escalável capaz de coordenadar grupos de UAVs na coleta de dados de WSN sob restrições de alcance de comunicação, sem fazer uso de técnicas de otimização. Resultados indicam que a quantidade de dados coletados pelo DADCA é semelhante ou superior, em até 1%, a abordagens de otimização de caminhos percorridos por UAVs. O atraso no recebimento de mensagens de sensores é até 46% menor do que outras abordagens e o processamento necessário a bordo de UAVs é no mínimo menor do que 75% do que aqueles que utilizam algoritmos baseados em otimização. Os resultados apresentados indicam que o DADCA é capaz de igualar e até superar outras abordagens apresentadas, agregando vantagens de uma abordagem distribuída.

Palavras-chave

Veículos Aéreos Não Tripulados; Coleta de Dados; Redes de Sensores Sem Fio; Algorítmos Distribuídos; Enxames.

Table of contents

List of figures

List of tables

List of Abreviations

D_n^T – Mean message delay time of received data in GS during the period T

S_{left} – Letf side of K of a given UAV

S_{right} – Right side of K of a given UAV

TD^T – Tax of collected data (collected divided by total generated)

ρ – Radio transmittion rate

b – Theoretical intersection between $S_{left} \cap S_{right}$

Ad hoc – A decentralised type of wireless network

B – UAV flight time, as known as flight endurance

CH – Cluster Head

DADCA – Distributed Algorithm for Aerial Data Collection

FOV – Field of view

FT – UAV Fail Tax

GS – Ground Station

K – Original tour length

LKH – Lin-Kernighan-Helsgaun Heuristic

LSN – Linear sensors network

MTBF – Mean Time Between Failures

mTSP – Variation of TSP with multiple salesmen

R – Radio range

ROS – Robot Operating System

SE – DADCA's Shadow Effect

SNIR – Signal-to-interference-plus-noise ratio

T – Period of data collection operation

TSP – Travelling Salesman Problem

TSPN – Variation of TSP with neighbourhoods

UAV – Unmanned Aerial Vehicle

V2I – Communication Vehicle to Infrastructure

V2V – Communication Vehicle to Vehicle

W – Period that a UAV needs to cover the entire tour K once.

WM – Wasted moviment

WSN – Wireless Sensors Network

p – Number of Cluster Heads

u – Number of UAVs

1 Introduction

This thesis investigates the possibilities and limitations of a distributed approach to coordinate multiple unmanned aerial vehicles (UAVs) for data collection from wireless sensor networks (WSNs). WSNs are deployed to collect environmental data for various applications (e.g. seismic monitoring, wildlife tracking, soil and air condition control, etc.). Data collection and transmission are the fundamental operations of WSNs [1]. However, in the scenarios that have no infrastructure or regions that lack communication, the data collection from sensors of the WSN faces serious difficulties[2]. Extensive research has been conducted on several kinds of mobile agents serving as data collectors for WSNs [3]. The use of UAVs as such mobile agents can enhance WSN data collection in areas that are difficult to reach due to hazardous conditions or communication limitations. Thus, in many applications UAVs are well suited as mobile agents for collecting data from a WSN, increasing the network's capacity and flexibility [4].

Coordination of UAVs in a distributed manner is a complex task [5] because there is no central node that is able to have a consistent view of the state of all UAVs (i.e. the instantaneous position, velocity, residual battery level, etc.); furthermore, the ad-hoc communication topology also limits consensus and task sharing. Therefore, such coordination requires effective algorithms to overcome these constraints. Providing a guarantee of effectiveness may threaten the efficiency regarding the amount of collected data or concerning the delay to collect data from WSNs by UAVs. UAVs have shown tremendous growth, both in research and applied use [6]. Concerning swarm of UAVs, the current UAV simple model uses a single ground controller to control one or more UAVs [7]. Regarding the control of UAVs, the first issue to resolve is the path plan. For this purpose, the travelling salesman problem (TSP) is widely considered [3]. Most of these studies focus on the use of a single UAV in an optimized tour or split the role problem into a smaller set of the same problem to be solved in the same way. It remains challenging to develop cooperative UAVs on area coverage tasks and energy-efficient UAV communication technology [8]. Moreover, as we aim to use a fully distributed approach, path planning should be computed in the UAVs, regarding its computational constraints.

To address these issues, this thesis investigates whether it is possible to implement a distributed algorithm to coordinate several fully autonomous UAVs (i.e. non human-controlled) collecting data from a WSN without centralized control or knowledge of internal UAVs states and relying only on ad-hoc communication. In response to these questions, we propose DADCA, a distributed algorithm that combines a well-known algorithms of path planning with a cooperative and oscillatory behavior that relies only on the exchange of all payload data from pairs of UAVs (whenever they approach) decision-making.

The remainder of this chapter outlines the motivation of this thesis and explains the main challenges of current methods for WSN data collection by UAVs. It also presents the main research questions, goals and contributions of the thesis. The chapter concludes by detailing the organization of subsequent chapters.

1.1 Motivation

The use of unmanned aerial vehicles (UAV), otherwise known as *drones*, for several purposes, has significantly increased in the last decade [9]. UAVs offer agile and cost-effective solutions for many demanding civilian applications [10], and have drawn significant research interest in recent years due to their wide range of applications, including surveillance and monitoring, footage in movies, sport events, inventory verification and inspection, cargo delivery, communication platforms, rural environment inspection, and disaster response and emergency relief [11][12]. Such wide-spread applicability is mainly due to the enormous capabilities of aerial vehicles in terms of mobility, autonomy, communication, and processing capabilities, in addition to its relatively low cost[9].

The emergence of WSNs has become an active research area in a broad range of critical applications [13]. However, in many cases it is difficult or even impossible to connect WSNs directly to their data destination. Typical UAV applications may involve the relaying time-critical data generated from WSNs on the ground to remote ground stations connected to the Internet [14]. UAVs can act as mobile data collectors, such as connection nodes in WSNs [15]. In such cases, UAVs can provide these connections by visiting the WSNs periodically and relaying or carrying the data to the proper destination. Recently, there has been growing interest in data collection through groups of collaborative UAVs [16][17].

One important application of UAVs is in emergency and rescue (such as earthquakes or nuclear explosions), in which the emergency team can benefit from a fast and comprehensive monitoring solution as soon as possible and until proper operation be organized. To support such monitoring, a WSN can be spread across the incident area and have its data collected by a single UAV. Using multiple UAVs to fulfill a mission can be considered advantageous compared to using one UAV [18] due to the typical large size of the geographic area and the demand to haul sensors data in a timely way. However, while using several UAVs can improve the data collection rate, multiple-UAV operations also introduce several challenges[19]. The approaches most often applied in these cases are derivatives of the Traveling Salesman Problem (TSP) and Vehicular Routing Problem (VRP).

At the same time, significant advances have been made for off-line path (tour) planning regarding the addressing the problem of distributed task allocation for multiple UAVs [20]. UAV coordination and their decentralized control are therefore timely topics [17]. Besides, communication is one of the most significant challenges in designing systems with multiple flying vehicles (a.k.a swarms) and also a crucial aspect of cooperation and collaboration [21]. In order to coordinate UAVs in distributed tasks, a inter-UAV communication approach must be effective.

A reliable communication infrastructure among UAVs in collaboration is critical in maintaining this connected network for data relaying tasks [14]. For this reason, reliable communication requires a radio base station close to the site. However, in some cases implementing this infrastructure to provide long-range UAV communication is not possible—for example, in emergency response and relief situations. Some emergencies, such as earthquakes, could destroy any existing communication infrastructure. In such cases, the location of the incident may be uncertain, and obtaining temporary infrastructure equipment (such as a mobile cellphone radio station) may not be immediately possible. These reasons make it impossible to use vehicle-to-infrastructure (V2I) solutions. In these cases, ad-hoc communication plays an important role.

With the goal of controlling several UAVs in a distributed manner, relying only on vehicle-to-vehicle (V2V) communication, we propose an approach that can be also use as a first-time emergency response approach for repetitive visitation of WSN sensors during a T period. In a early phase of this thesis work [22], we introduced an early version of the Distributed Algorithm for Aerial Data Collection (DADCA), a distributed algorithm that coordinates groups of UAVs collecting data from WSNs. DADCA yields better results than the TSP in specific scenarios. It does not rely on central control and runs in a distributed manner, with all processing done only in the UAVs. This algorithm takes into consideration the fact that the number of working UAVs can change during the cyclic visitation period T due to UAV malfunctions or UAV replacements, or expansion the set of UAVs during its execution without synchronization.

In this thesis, we extend this idea, and propose new distributed approaches based on DADCA for aerial data collection. These new approaches focus on the use of different path planning techniques which aiming to bring better results in distinct metrics. We also evaluate these new metrics and implement other authors' most recent approaches to provide better comparisons. We also analyze the trade-offs between distinct approaches to planning the path to collect data from WSNs.

Disasters can occur over large areas, as did the large floods throughout Japan in 2018 [23] and major earthquake in Italy in 2016 [24]. These incidents can occur in unpredictable or inaccessible areas, which makes it difficult to rescues or maintain surveillance. Some recurring natural events like tsunamis [25] and hurricanes [26], although they can be predicted in advance [27][28], can be devastating to hundreds or even thousands of people [29].

Regardless of the nature of the disaster, search and rescue activities begin as soon as possible after the event [30]. Many of the impacted areas may be under harsh conditions, remote, and isolated due to a lack of communication infrastructure. Building a permanent communication infrastructure may be considered too expensive too slow and difficult to maintain. WSN can play an pivotal role in such situations by providing extensive data (e.g. about the people, the infrastructure, the rescue teams and the environmental condition) to improve search and rescue operations. Seismic, nuclear, humidity, noise, and heat are examples of data from WSNs that can provide critical data to help SAR teams locate missing people or determine emerging hazards.

For example, we present the following scenario: after a nuclear accident at a nuclear plant, some rescue tasks need to be executed as soon as possible. For a defined period after the accident—72 hours, for instance—it is crucial to maintain continuous awareness about radiation reads throughout a vast and affected are by the radiation. For example, several radiation sensors may be placed around the plant, while UAVs can be used to collect data straight from the sensors or cluster heads (CHs).

When deploying a WSN in such large-scale settings or in remote regions, or both, the use of mobile data collectors (mobile sink nodes) can be an attractive option. This method can overcome possible WSN partitions that prevents or reduces WSN routing; and provides move adaptivity of the itinerary and the residual energy at each WSN node, as by others[31][32][33]. Additionally, mobile sink nodes—especially with UAVs—is the most convenient method for accessing each sensor node in a large-scale WSN [17].

Some authors have proposed the use of a single UAV as a data collector [34][35][36][37]. However, a single UAV acting as a mobile sink node has some drawbacks. For example, it represents a single point of failure, that may compromise the WSN operation as a whole. Another issue is the latency due to the time that a single UAV would require to visit all the nodes.

The use of multiple UAVs can enhance tolerance agains malfunctions, because the dynamic group of UAVs can stand in for for some UAVs malfunctioning in the group. A group of UAVs shall also reduce the overall latency through a divide and conquer approach, i.e. splitting the set of all points to visit among the operating UAVs. A divide and conquer approach can reduce the individual task of a UAV, enabling faster delivery of the collected data.

Moreover, the use of UAVs as mobile sink nodes requires long-range enabled radios with some kilometers of range, which is more suitable for applications employing a single UAV. Shorter range radios with uses less energy and tend to be lighter, thereby indirectly requiring less UAV energy expenditure to maintain the flight. Moreover, if the UAV flies closer to the CH, the uplink transmission energy of the CH required to send the target data can be further reduced [38] which augments the CH battery life time.

1.2 Problem statements

1.2.1 Communication

The problems motivating this work relate to incidents that have destroyed the pre-existing communication infrastructures, e.g. earthquakes knocking down radio base station towers. Thus, this work relies on a scenario with no pre-existing communication infrastructures.

The use of long-range communication simplifies WSN data collection. If a UAV has a radio with 5km of radio range, it could cover an area of 12,5km^2 delivering its collected data straight for a base station up to 5km of distance. However, it is expected that long-range radios would consume more battery power and therefore would rapidly drain the energy of the entire network. In general, energy conversion is most efficient only for short ranges[39].

The use of short-range radios entails ad hoc communication. Compared to traditional ad hoc networks, the mobile ad hoc networks established by

UAVs are more efficient for completing complex tasks in harsh environments [40].

Therefore, the first problem addressed in this work can be stated as follows:

> **Problem statement I:**
> For WSNs deployed in a remote and difficult-to-reach region and without coverage of a wide-area communication infrastructure it is very difficult to collect the corresponding sensor data in a timely way.

1.2.2 Coordination

Needless to say, coordinating multiple UAVs for collecting WSN data requires an approach to coordinate their movements (tours). In order to control multiple UAVs in any activity, centralized control of the UAVs is usually performed as in [41], [42] and [43].

The use of centralized control facilitates the allocation of UAV activities but requires communication between each of the UAVs and a control point as a ground station. Without such communication it is necessary to use a decentralized approach in which UAVs can cooperate with each other to achieve the overall goal just as swarms of UAVs have been targeted at other collective tasks and applications, such as UAVs in repeating patterns [44] and UAVs on area Coverage [18].

Therefore, the second problem addressed in this work can be stated as follows:

> **Problem statement II:**
> Decentralized Movement coordination in swarms of UAVs is a complex endeavour, as decisions have to be made in real-time and collectively, and the outcome must be coherent with the common task.

1.2.3 Path-planning

Determining the optimal order to visit sensors or CHs (i.e., path or tour) in a WSN is a well-known NP-hard problem [45]. It is reasonable to expect that the computational capacity of a computer embedded in a UAV is limited and that State of The Art in path-planning optimization algorithms may not be the most appropriate for this scenario. Non-optimized forms of path-planning processing may be the best choice for decentralized movement coordination along a common route..

Therefore, the third problem addressed in this work can be stated as follows:

Problem statement III:
As the processing and memory resources on each UAV are scarce, traditional optimality seeking path-planning algorithms for the computation of the CH visitation tour cannot be executed there. Thus, the challenge is on how to compute sub-optimal tours that still render an efficient collective and coherent tours for a set of UAVs.

1.3 Goals and research questions

To address the problems described above, the main general goal (G) of this work is:

G: To propose and study approaches for the collection of WSN data by UAVs based only on distributed coordination and ad hoc communication.

This general goal is intended to be achieved by answering the following research questions (RQs):

RQ1: What would be an effective approach for UAVs to collect data from a WSN regarding mean delay time? This RQ is related to Problem I. It aims to investigate how the proposed approaches can be compared concerning the delay of collected messages.

RQ2: How much more efficient it is to use a distributed approach to WSN data collection by multiple UAVs in comparison to direct application of known approaches for single UAVs? This RQ is related to Problem II and aims to investigate how the proposed approaches can be compared in terms of the amount of data collected.

RQ3: How do the proposed strategies behave in terms of UAV hardware resource comsumption? This RQ is related to Problem III. Its purpose is to investigate how the proposed approaches can perform on devices to the UAVs' embedded hardware.

By answering these research questions, we envisage three sub-goals (SGs) to be achieved in our work:

SG1: Propose a approach to route planning and data collection in WSNs by groups of UAVs.

SG2: Define the relevant comparison criteria for such WSN data collection approaches.

SG3: Identify the limits of the proposed approaches and trade-offs relevant to the choices between them based on the established criteria.

1.4 Contributions

The main contributions of this thesis are:

- Proposal of a distributed approach that coordinates the flight movement of UAVs for the collection of WSN data assuming that the position of the CHs to be visited are known in advance;
- An extensive discussion of the experimental results and analysis of trade-offs based on the selected criteria and other approaches presented.

1.5 Organization

This thesis is organized as follows. In the next chapter, Chapter 2, we present and discuss the main related works that address the problems described above. Chapter 3 further elaborates the underlying problem and presents criteria for comparison of the approaches: collected data tax, message delays and processing time in UAVs. Chapter 4 presents the proposed algorithm, DADCA, and related work used as a benchmark. This chapter also discusses the algorithm steps, the algorithm variation, and the limitations. Chapter 5 presents the evaluation model, simulating two real scenarios with possible nuclear issues. Chapter 6 presents the simulation results and comparisons between the proposed approaches. Chapter 7 presents the concluding remarks concerning the proposed approach and presents suggestions for future work.

2
Related work

This chapter presents several recent approaches to WSN data collection by UAVs. Overall, they can be classified according to their main investigation focus: UAV and WSN interaction, UAV tour planning for data collection and communication concerns. Sections 2.1 to 2.3, respectively, briefly discuss the these studies. Finally, Section 2.4 presents these works in a comparison table and Section 2.5 summarizes the discussion.

2.1
UAV and WSN interaction

In WSNs data can be gathered using: (1) the classical approach including a network with infrastructure; (2) mobile elements, such as ground robots or UAVs [46]. We focus on studies regarding aerial data collection from WSNs with UAVs as the mobile elements.

Regarding data collection from WSNs, it is usual to cluster their sensors in order to save energy. The low energy adaptive clustering hierarchy (LEACH) is commonly used to select cluster heads (CHs) for the purpose of conserving energy [47]. UAVs can collect data from CHs in the same way that a data collector can navigate from sensor to sensor in a WSN. Both scenarios require a tour plan. The study by Dios et al. is one of the seminal works on WSN data collection with UAVs [48]. The authors propose using UAVs to collect data from several areas with multiple sensors. Their work includes no inter-UAV cooperation different from the present research. Instead, each UAV relies on the same TSP tour to arrange their designed WSN areas in order of visitation.

Some related works, such as [49] and [50], divide the whole set of CHs by the number of available UAVs and send a single UAV for the subset of CHs. Their approach may decrease the delay in the UAVs' data collection, but their work does not address the central point of failure problem presented by the use of a single UAV for each cluster, since a single UAV is dispatched to distinct areas. The use of several UAVs for each set of CHs could address the failure issue. Regarding the use of distinct UAVs for subsets of CHs, the present research addresses the problem of data collection from the perspective of a single set of CHs. We aim to investigate the aerial data collection from the perspective of a non-divisible set.

Regarding the use of multiple UAVs working in the same context, there exist frameworks that aim to enable multi-UAV control [51][52][53]; these rely on hardware enablers through the Robot Operating System (ROS) [54]. However, these frameworks provide no distributed coordination investigations. Mazayev et al. have conducted research that takes multiple-UAV data collection into consideration [46] but focuses on a heuristic to optimize the UAV buffer size and speed constraints with a centralized approach. Several UAVs working within a WSN are taken into consideration by Sharma et al. [55] also study multiple UAVs working with a WSN, but their work focuses on data dissemination and energy efficiency inside WSNs.

Indeed, the study by Qadori et al. [13] indicates a lack of research on collaboration approaches among agents for collecting and summarizing data. The studies that they cite investigate multiple agents working side by side independently (as the UAVs in [48]), rather than in collaboration [13]. This study draws an analogy with agents navigating through a WSN and UAVs navigating among CHs in terms of collaboration.

Jawhar et al. present another framework, which uses several UAVs to collect data from Linear Sensor Networks (LSNs) [56]. Similar to our work, their study involves multiple UAVs collecting data from a WSN. However, the framework developed by Jawhar et al focuses on the linear and hierarchical model of LSNs. In this context, their work explores the nuances of constant and variable speeds of UAVs [56]. In contrast to their research [56], this work includes no assumptions or constraints regarding how the sensors are spread over an area.

2.2 UAV tour planning

Research on sensor data collection with UAVs has developed various tour-planning techniques. Some works suggest the straight use of conventional techniques based on the TSP or VRP variations [5][57][58]. Other works focus on optimization to solve the TSP by applying genetic algorithm (GA) optimization, as in [59], [60], and [50]. Yet others employ particle swarm optimization (PSO), as in [61] and [62]. Considering the radio range as neighborhoods[63], some works discuss TSP-like heuristics, such as [16] and [61]. However, all of these works consider only a single UAV collecting data and focus on optimizing the tour. This study uses well-established tour planning techniques and does not aim to contribute to research on optimization techniques.

In contrast to the TSP and VRP optimization approaches, Wang et al. [50] study a scenario with a large set of sensors, which is hard to solve with the aforementioned optimization approaches. They suggest a sort-based tour-planning process, enabling a vast number of CHs to collect data. The research by Wang et al. [50] is one of the benchmarks to which this work's proposed algorithms are compared.

2.3 Communication approaches

Data collection problems in sensor network nodes can take many different approaches. Similar to the Google project Loon [64], Facebook project Aquila [65], and the project Absolute [66], the work of Zhang et al. [67] focuses on enabling ground communication through the use of several UAVs. In these works, the UAVs act as mobile sink nodes with large throughput and long-range radios. Specifically, in [67] the proposed solution focuses on coverage modeled as a mixed integer nonconvex optimization problem. These works use UAVs as a mobile base station for enabling aerial data collection as in [68], [69] and [70] for UAVs acting as flying 5G base stations. The authors in [71] discuss how UAVs are constrained by limited resources including communication. Our work differs from these in that our approach does not rely on long-range radios in the UAVs.

Yanmaz et al. [72] suggest a high-level architectural design for a collaborative aerial system, which consists of UAVs with onboard sensors and embedded processing, coordination, and networking capabilities. Their work makes use of the ROS mentioned above [54] and relies on IEEE 802.11s mesh communication to provide proof of work. The research by Yanmaz et al. is similar to the present work in that it relies only on ad-hoc communications. However, in [72], the data collection is related not to WSN but to UAV on-board censoring and the distributed decision-making structure focuses only on UAV collisions [72].

The survey by Gupta et al. [73] brings together several studies regarding UAV networks. The most frequently cited type of network relies on a backbone providing long-range communication as presented in [74] for cloud-assisted data collection. However, in data collection and search-and-rescue scenarios it is reasonable to use flying ad-hoc networks (FANET) [73], a specific type of mobile ad-hoc network (MANET)[75].

There are other lines of inquiry on data collection with UAVs beyond tour-planning techniques. Some works focus on opportunistic sensor data collection with UAVs, as discussed in [76], but still rely on a single UAV.

Regardless of the UAV tour planning, some works investigate a low-level communication approach, as described in [77], which provides medium access control (MAC) protocols dedicated to UAVs collecting data from WSNs [78]. Other works focus on UAV or WSN energy efficiency, as discussed in [79], [8], [80] and [81]. Berrahal et al. describe another approach in which multiple sensors comprise the structure for a WSN specializing in border surveillance [82], but the UAVs have no collaboration or knowledge of others UAVs.

2.4
Comparison overview

The Table 2.1 presents an overview view of major related works. The column *WSN relation* presents directly related works regarding data collection and some works presenting highly similar studies. The column *Work & tour approach* presents the main challenge and the tour-planning approach of each work. The column *Comm. Approach* indicates how the UAVs communicate, including V2I (vehicle to infrastructure) in cases where UAV long-range radios or other infrastructure is in place, or ad-hoc in cases where each UAV relies only on short-range radios. In ad-hoc communication scenarios, UAV just exchange data upon some kind of meeting if the other communication part is another UAV or a base station. The column *Multi-UAV* indicates whether the related work takes into consideration more than one UAV operating at the same time. The column *Coordination* describes how the UAVs are controlled.

Table 2.1: Related Works

Related Work	WSN relation	Work & Tour approach	Comm. approach	Multi-UAV	UAV Coordination
This work	Data collection	Polynomial path planning	ad hoc	Yes	Distributed
Dios et al.[48]	Data collection	TSP straight use	V2I	No	Centralized
Several researches [5][57][58][59] [60][61][62]	Data collection	TSP optimization	V2I	No	Centralized
Jin Wang et al. [49]	Data collection	Choice of CHs by UAV tour	V2I	Yes, in isolation	Centralized
Wang et al. [50]	Data collection	FPPWR	V2I	No	Centralized
Some researches [68],[69] & [70]	Data collection	Optimization for area Coverage and relay through 5G	V2I	No	Centralized
Mazayev et al.[46]	Data collection with buffer and TTL constraints	Tour optimization upon constraints	V2I	Yes, in isolation	Centralized
Ma et al. [76]	Opportunistic data collection from mobile sensors	Predefined static route	ad hoc	No	Centralized
Berrahal et al. [83][82]	Data collection borders sensoring & video surveillance	Predefined static route	V2I	Yes, in isolation	Centralized
Jawhar et al.[56]	Data collection, but in specific sensor distribution (LSN)	Predefined static route	Hierarchical V2I	Yes, in isolation	Centralized
Yanmaz et al.[72]	Direct sensoring	mTSP straight use	V2I & ad hoc	Yes, in isolation	Centralized
Zhang et al.[67]	Data dissemination w/ constraints	Optimization for area Coverage	V2I	Yes, but static	Centralized
Sharma et al. [55]	Data dissemination	Workload area divided by the number of UAVs	V2I & ad hoc	Yes	Centralized
Thammawichai et al.[71]	Pursue a single target or survey an area of interest for data relay	Constraints optimization	Hierarchical V2I	Yes	Centralized

2.5 Summary

The works cited above do not provide approaches for multi-UAV distributed coordination in WSN data collection. Instead, they either do not use multiple UAVs or use them in isolation. As discussed in [13], further research is required on multi-collector approaches, particularly regarding cooperation among them.

To the best of our knowledge, most related works in this area of research only consider data collection in a single moment, meaning that at some point a single UAV passes through every cluster to collect data once[34][84]. However, as discussed above, certain situations may require data to be collected for a period T, such as search-and-rescue missions or monitoring.

This work explores collaboration among UAVs performing data collection and compares our approach and algorithm to the aforementioned studies in the case of multiple UAVs. The collaboration among UAVs involves dynamically resizing each UAV tour upon a UAV malfunction or reinforcement. This resizing process prevents uncovered CHs after a UAV leaves the data collection system. It also prevents variable delivery delays. The collaboration is also responsible for forwarding messages between UAVs to reach the ground station (GS) without the necessity of a UAV displacement to the GS to deliver its collected data.

3
Aerial data collection

We investigate how to use several UAVs for such data collection by testing different strategies to collect data from the from a ground-deployed WSN using multiple UAVs. Chapter 4 then presents two strategies to collect data from the WSN with UAVs on the basis of related works and proposes a distributed approach to meet this objective. In this section, we define the application architecture and metrics used to compare approaches with distinct strategies in such data collection.

There are different types of sensors which read the data that will be consumed in different manners. For example, there are sensors as cameras that make sense consume its data as FIFO (first in first out) perspective in a sense to follow their images evolution. In another perspective, there are radiation sensors which will make sense to have the newer sensor data as fast as possible, so a LIFO (last in first out) data collection will be making more sense.

In both cases, have more sensor data will be more useful to follow the remote state but not the collection order. Further, in this work, we present our evaluations in a LIFO data collection perspective. Our work does not aim to be the best fit for both scenarios (LIFO and FIFO) because it is an application-driven choice.

3.1
Scenario overview

A possible scenario in which collecting WSN data with UAVs might be useful is an unforeseen problem or accident in an wireless networks spread over an uncovered area. As an example, we present the Fukushima accident [29], in which a nuclear accident led to the evacuation of an area totaling $160km^2$. This area needed to remain under monitoring for nuclear hazards and other possible accidents due to the tsunami for an initial period of T until proper countermeasures were put into place. To perform this monitoring, we propose the use of a WSN and UAVs. A WSN provides remote sensing, and UAVs provide access to the site avoiding human contact with hazardous conditions. More examples are listed in [48].

The problem presented can be characterized as a coverage problem but without centralized control or processing. We aim to use autonomous UAVs without central control or supervision. Central processing point can bring single points of failure or even processing limitation regarding the single point capabilities while a fully distributed system can work without a single point of failure or limitations regarding the number of UAVs.The UAV's mobile capabilities are equivalent to a VTOL UAV. A VTOL (vertical take-off and landing) aircraft is one that can hover, take off, and land vertically. A VTOL aircraft can act as a quadcopter and a fixed wing aircraft. Our scenarios assumes that each UAV is capable of avoiding collisions.

A simple technique to prevent collision would be to fly over possible obstacles and use slightly different altitudes for each UAV. As an alternative, Mazayev et al. propose more sophisticated techniques using visual processing onboard UAVs[46]. The former technique has no processing requirements, while the latter types require significantly more onboard processing power. We apply slightly different altitudes for each UAV in our approach.

3.2 System overview

Our model assumes that each UAV, sensor, and CH, as well as mobile GS, has a short-range radio with R range for communication. Communication occurs only when two nodes are within range R. We assume that the sensors and CHs have a static position on a $\mathbb{R}^3$ surface and their number is always larger than the set of UAVs.

We divide the solution to the role problem of interest into two phases: a preparation phase and a collection phase. This work focuses on the second phase, but for background information, we briefly describe the preparation phase in Figure 3.1. As an additional note, we consider the area of interest to be unknown until the accident occurs, so no WSN is put into place.

When an accident occurs (Figure 3.1.a), the sensors are spread over the area by air using a larger UAV or manned aircraft (Figure 3.1.b). As noted in Figure 3.1.c, these sensors are then able to form into clusters and will be accessible by the CH. In order to identify the CH coordinates, a UAV can be launched to fly over the entire area in a Zamboni pattern [85] and detect the coordinates of each CH. For more simple sensors that do not have GPS, a heuristic such as that presented by Mazayev et al. [46] may be applied.

As indicated in Figure 3.2.e above, the UAVs keep on collecting data for a period of time T. We illustrate a snapshot the entire scenario in Figure 3.3, which presents an area of interest after an accident.

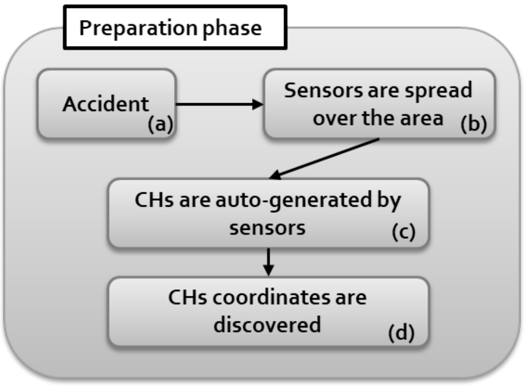

Preparation phase
Accident (a)
Sensors are spread over the area (b)
CHs are auto-generated by sensors (c)
CHs coordinates are discovered (d)

Figure 3.1: Figure illustrating actions from the accident to the CH coordinates discover.

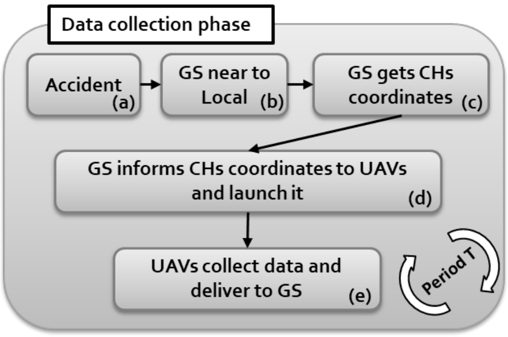

Data collection phase
Accident (a)
GS near to Local (b)
GS gets CHs coordinates (c)
GS informs CHs coordinates to UAVs and launch it (d)
UAVs collect data and deliver to GS (e)
Period T

Figure 3.2: Figure illustrating the problem from the GS perspective. Figure 3.1.d information is used here from item (c). Item (e) presents the cyclic data collection.

A CH remains in passive mode waiting for a UAV flyover. Figure 3.4 illustrates the dynamics of UAV-CH message exchange. Each UAV flies with a constant velocity V and altitude H. Each UAV, after reaching a distance of less than 1.1*R to the CH locations, begins sending a *field of view* message (FOV). When a CH receives a FOV, it prepares the exchange data transfer process with the UAV. That role situation remain in place until the CH stops receiving FOVs. Figure 3.4 illustrates a UAV collecting data from a CH during its stay within a distance S, with $S < R$. In Figure 3.5, we explain the reason that distance S is smaller than radio range R. Due to the UAV flight altitude H, it cannot travel beyond a distance of 2*R and continue exchanging data with the CH as ilustraed in Figure . As indicated, the choice of H directly impacts the period of time available for a UAV to collect data from a single CH.

To organize the set of UAVs collecting data from CHs and delivering data to the GS, a set of rules must be followed, which we consider a strategy. This work presents various strategies for organizing the data collection, all of which incorporate certain assumptions: (1) All communication is executed through

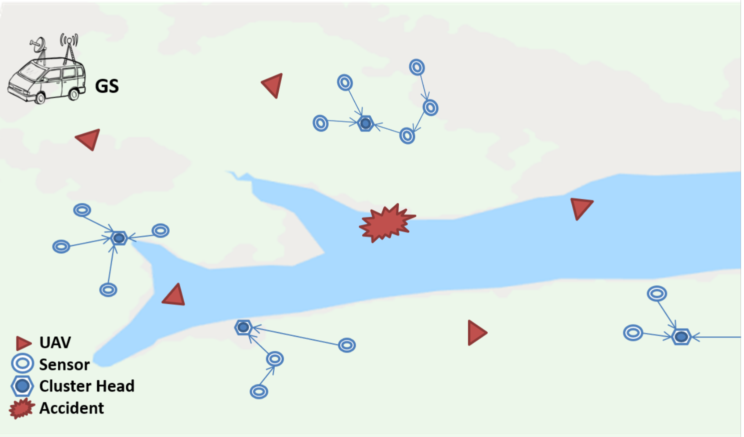


Figure 3.3: Figure presenting the interest area with sensors, CH, GS and UAVs flying around.

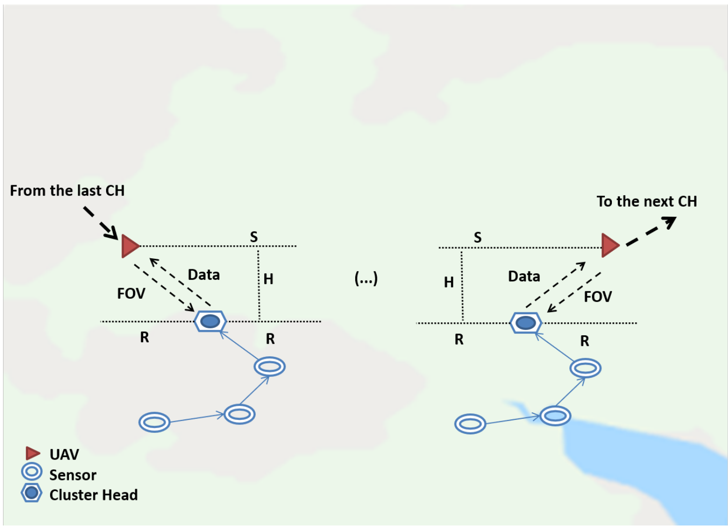


Figure 3.4: The Figure presents a UAV before and after the feasible exchange horizontal distance S with the constrain of $S \leq 2\sqrt{H^2 - R^2}$. R is measured on the ground, H is upright from ground and S is parallel to R.

short-range radios in ad-hoc mode; (2) the GS does not act as a controller node, only as a relay node; (3) CH coordinates are available from the step presented in Figure 3.1.d.

During the data collection period T, some UAVs can fail temporarily or permanently without notice. For example, a UAV's batteries can be depleted. In such cases, the UAV returns to the GS to recharge and then resumes collecting data under the rules set in place. This communication model is not perfect and the radio range R is not constant and messages can fail, which is presented in evaluation section.

Some issues, such as security or Byzantine fault tolerance in the coordination of UAVs, fall outside the scope of this research. We also do not intend to analyze lower-level radio communication issues in movement. This research

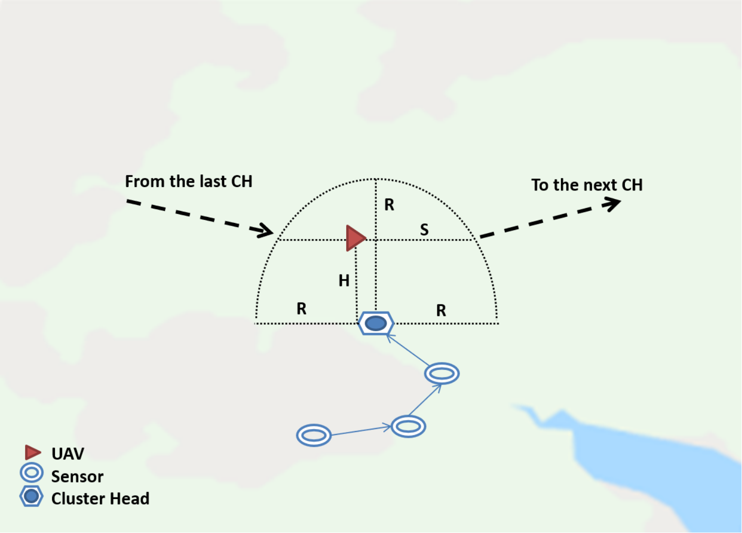


Figure 3.5: A UAV inside the range of message exchange with a CH. The images simplify the radio range as a uniform disc, but our simulation does not as present in next chapters.

presumes that such communication is successful, and the only radio issues addressed are those listed in Chapter 5.

3.3 Metrics formalization

We aim to compare our fully distributed approach with state-of-the-art approaches which are centralized. To compare strategies for collecting data from a WSN using UAVs, it is necessary to select certain metrics. All metrics and comparisons are related to the period of interest T. Three metrics were chosen in this study for the following aspects: the amount of data collected, the average data transfer message delay, and the UAV embedded processing time.

The first metric is the total amount of sensors' collected data, initially measured in kilobytes, then presented as a rate-related amount of data that arrives at the GS.

During period T, the set of p CHs generates an amount of data presented in (1) and each single CH i has its own data rate generation C_i. Thus, C_i^T represents the amount of data generated after a T period by a i-nth CH. So, the C_p^T is the total amount of generated data from all CH.

$$C_p^T = \int_0^T \sum_0^p C_i dt \quad (1)$$

For each instant t of T that a single CH i is exchanging data with a UAV, a value $\overline{C}_i^T$ is accumulated in the C_i rate as in (2). A transmission rate ρ is multiplied by $\overline{C}_i^T$, indicating the total amount of transmitted data by a CH i during a period T. Only data that arrives at the GS is taken into consideration in $\overline{C}_i^T$. The rate ρ is the radio transmission rate in bytes per second.

$$\overline{C}_i^T = \int_0^T \sum_0^p C_i * \rho \quad dt \qquad (2)$$

Furthermore, when T is reached, the rate of collected data TD^T (3) is obtained by dividing the collected data $\overline{C}_i^T$ by the sum of the generated data C_i^T of all CHs p. TD^T represents the percentage of collected data in the evaluation graphs.

$$TD^T = \sum_0^p \frac{\overline{C}_i^T}{C_i^T} \qquad (3)$$

The second metric is the data delivery delay. It is the time between a message be sent by a CH and received at GS. For each message m_i sent by a CH to a UAV, the UAV receives a timestamp with its payload. The set M_i^T represents all delays of n messages sent by a CH that reach the GS during the period T, as illustrated in (4). In (5) we calculate the average delay as the sum of all delays divided by the number of messages. This metric is presented in seconds.

$$M_n^T = \int_0^T \sum_0^n \quad delayOf(m_i) \quad dt \quad (4)$$

$$D_n^T = \frac{M_n^T}{n} \quad (5)$$

The third metric is the total time PT spent by a UAV from the moment it receives the CH locations to the instant it completes its tour. This metric is calculated in seconds.

4
Strategies for aerial data collection

This chapter discusses the strategies used to address *the way of* UAVs can effectively collect data from CHs. Figure 4.1 illustrates a possible arrangement of CHs that would need to be visited by UAVs to obtain CH's data during a period T.

As only short-range communication is assumed to be available, all CHs must be visited by at least one UAV during the collection period T. The set of CHs can be interpreted as points of interest, and a path must be set for each UAV. We call the ordered path that a UAV navigates to visit a set of CHs a "tour". Thus, from the perspective of the CHs, this can be treated as a coverage problem and reduced straight to problems as the TSP and the VRP (vehicular routing problem). As any UAV can move from one CH to another in any order, it is a complete graph.

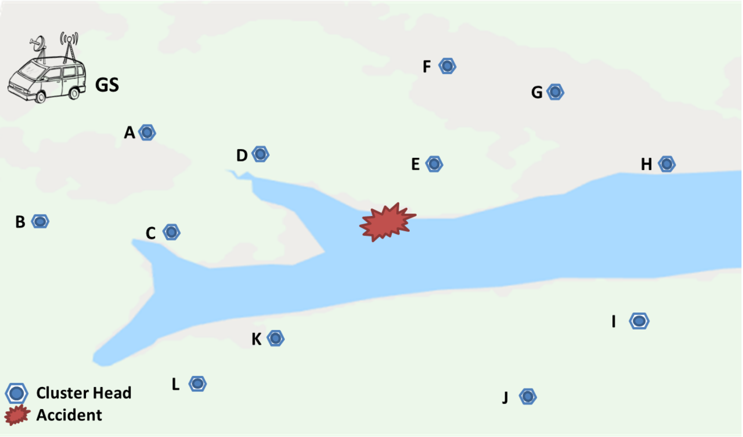


Figure 4.1: Scene view with CHs and GS. Each strategy determines how UAVs will visit CHs to collect data to deliver to GS.

The edges of Figure 4.1 are discretized as a complete graph, which illustrates a possible arrangement of CHs during a period T. CH A represents the GS. The cost of moving from one CH to another can be interpreted as distance or as more complex cost (e.g. composite costs combining more than one value, such as distance and barriers) as presented in Figure 4.1.

The following subsections describe the most recent and advanced approaches that use UAVs to collect data from WSNs, in addition to our approach to a fully distributed UAV application. For all of the following strategies, the time to compute the tours is not discussed until section 6.

4.1 Optimization approachs (TSP-based)

Similar to the strategy presented by Burman et al. [86] and Ho et al. [87], this strategy includes visits to all CHs by way of the best possible tour. This strategy directly maps to the TSP problem. Accordingly, the best tour is the result of a minimization function. For future reference, we named this approach the TSP-based strategy.

This TSP-based strategy represents several up-to-date optimization approaches for collecting data with UAVs [16][46][57][63][78][90-93]. On the complete graph $G(V, E)$, the vertices V are CHs, and the edges E represent the cost of a UAV flight between two vertices V. Obstacles and prohibited flight zones are represented in the costs attached to the edges.

In the studies discussed in the related works section, it is typical to use a single UAV. However, as previously mentioned, this method introduces a single point of failure. Moreover, this approach requires relevant computational effort to compute the best tour in a centralized manner. In our scenario, we implement this strategy by using several UAVs on the same tour.

In most works, on the best tour computed, all UAVs fly keeping an equal distance from each other and perform their data collection following a Hamiltonian Cycle, as shown in Figure 4.2. When each UAV passes near the GS, it delivers its collected data. As the collection takes into consideration a complete graph, it is obviously difficult to compute the best tour with a large number of CHs, because TSP is an NP-hard problem.

Although this approach is not a distributed strategy but a centralized one with UAVs, we present this strategy as a benchmark to be used as a comparative baseline.

4.2 FPPWR algorithm

Fast Path Planning with Rules (FPPWR) is a solution proposed by Wang et al. [50] to address the computational effort necessary to solve the optimization problems in a TSP-based strategy when collecting WSN data with UAVs. This strategy relies on the FPPWR routine to compute the tour, which we explain in next paragraphs. Once the FPPWR tour is computed by the GS, all UAVs are sent to collect the data. We apply this strategy by programming all the UAVs to fly at an equal distance from each other and

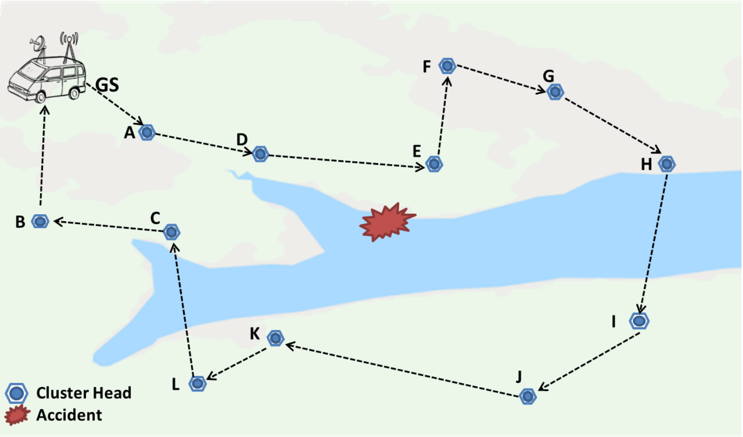


Figure 4.2: A possible tour generated by TSP-based strategy.

perform their data collection following a Hamiltonian Cycle, as in the TSP-based strategy described above. Then, as shown, each UAV delivers its data when it passes near the GS.

In the FPPWR strategy, tours are calculated by splitting the area into smaller geometric clusters, as illustrated by the dotted lines in Figure 4.3. These clusters are calculated using the radio range of UAV and CHs. FPPWR then performs a sort-based computation on the CH positions. This sort-based tour computation runs on each cluster, taking into consideration the most recent cluster processed. All clusters are analyzed in the order indicated by the arrows in Figure 4.3. Figure 4.4 illustrates a possible tour generated by FPPWR.

The algorithm complexity is $\mathcal{O}(p \log p)$ [50] (as p denoting the number of CHs). However, the fixed time to split the clusters cannot be neglected. This issue is discussed in the evaluation section.

As in the previous strategy, this is not originally a distributed strategy but a centralized one, using several UAVs in a straightforward manner. To the best of our knowledge, this is the only other approach that proposes an alternative to optimization strategies, besides our own.

4.3 DADCA

The Distributed Algorithm for Aerial Data Collection (DADCA) is an algorithm [22] that maintains the sensor data collection by UAVs during a period of time T (measured in hours). This strategy is our approach to enable a distributed data collection with several UAVs. We do not claim to get better results than optimization approaches, but we propose DADCA as a viable alternative to centralized approaches.

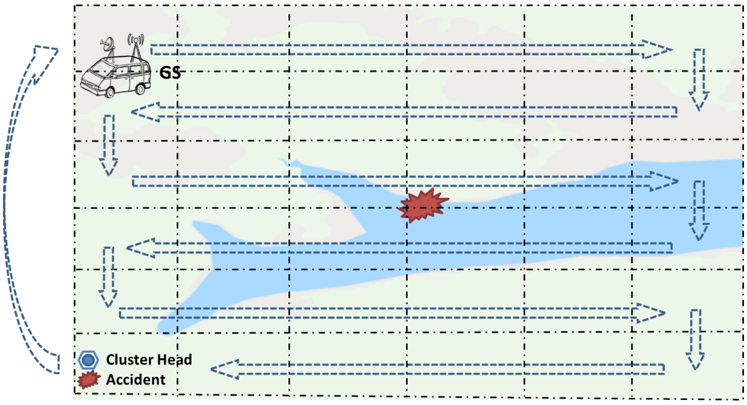


Figure 4.3: Clusters distribution and computation order for tour creation by FPPWR strategy.

In order to present DADCA in details, we divide its presentation into two phases as follows. In Figure 4.5 DADCA's main architecture is presented: (1) Figure.4.5.a indicates the computation responsible for planning the tour that the UAVs will follow and (2) Figure.4.5.b is the strategy for using the computed tour in Figure.4.5.a.

From this point on, the tour computed in Figure.4.5.a will be called the *Original Tour*. Each UAV plan its *Original Tour* and, as all UAVs start working near from GS, all Original Tours are the same. The *Original Tour* is not computed again during the period of time T and all UAVs will have the same *Original Tour*. The *Original Tour* is a simple path that passes by each CH only once and does not return to GS. Instead of doing the Hamiltonian Cycle the UAVs will use the *Original Tour* moving forward and backward meeting each other on the same tour.

We begin with DADCA tour computation explanations and then all DADCA variations share the same UAV behavior that will be explained on

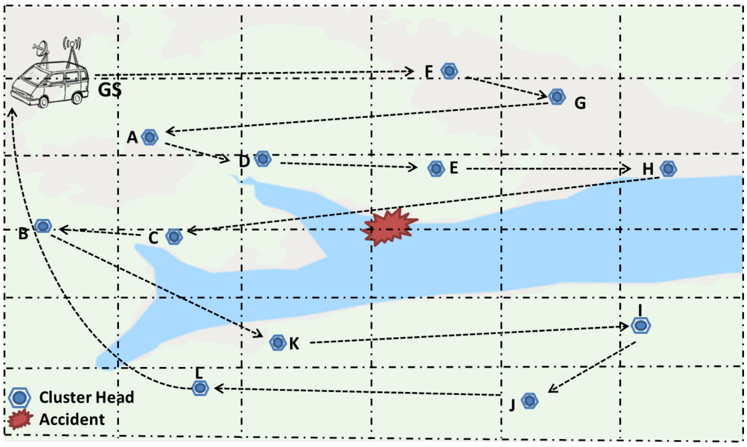


Figure 4.4: A possible tour generated by FPPWR strategy.

section 4.3.5. The main characteristics of the tours used in the first phase of the DADCA method, as illustrated in Figure 4.5.a, are the following:

1. UAVs do not compute a Hamiltonian Cycle. The tours used by DADCA are defined by a single path leaving the GS and passing close to each CH exactly once without returning to the GS. Therefore, there is no round-trip using DADCA, as opposed to the TSP-based and FPPWR methods;

2. The DADCA tours are computed without any optimization techniques. DADCA enables processing time to be predictable based on the size of the CH set, enabling the processing unit present in UAV to execute the processing. Furthermore, DADCA is suitable for use distinct tour planning tecniques with its individual advantages and its restrictions of each tour planning characteristics.

From section 4.3.1 to 4.3.4, we present four distinct tour computation methods that generate non-optimized tours used in Figure.4.5.a. In section 4.3.5, we explore the approach used in Figure.4.5.b.

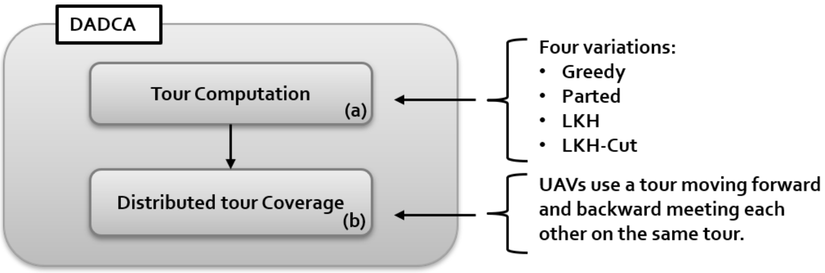

DADCA
Tour Computation
(a)
Distributed tour Coverage
(b)
Four variations:
• Greedy
• Parted
• LKH
• LKH-Cut
UAVs use a tour moving forward and backward meeting each other on the same tour.

Figure 4.5: DADCA phases.

4.3.1 DADCA-greedy

In this DADCA variation, we generate the *Original Tour* with a greedy algorithm, which is polynomial and deterministic and runs at $\theta(p^2)$ (as p denoting the number of CHs) and searches for the nearest CH not included in the tour and addes it to the tour. Algorithm 1 demonstrates the DADCA-greedy first phase, with line 1 presenting the function *NaviteTour()*, which is the tour planning of DADCA-greedy. From the GS to the final CH in the *setOfCH*, the nearest CH is obtained through the *GetNearestNotTaggedCH()* function in line 6.

DADCA-greedy does not necessarily create the best tour. Instead, it creates the tour in a straightforward manner. DADCA-greedy can provide a shorter tour than the best tour, which contains a Hamiltonian cycle, due to the reduced number of edges. Function *GreedyTour()* creates a tour with one edge less than the TSP solution. Nevertheless, as the Function *GreedyTour()* checks the nearest CH from the CH added last, it can also provide a much larger tour than a TSP tour optimization. This is the original DADCA approach presented in [88].

Algorithm 1: DADCA Greedy - Tour Planning

```
function $GreedyTour(setOfCH)$:
    $tour \leftarrow null$
    $nxtCH \leftarrow null$
    $tour$.add($GS$)
    while $tour.size \leq setOfCH.size$ do
        $nxtCH \leftarrow$ GetNearestCH ($tour$.last; $setOfCH$)
        $tour$.add($nxtCH$)
        $setOfCH$.remove($nxtCH$)
    return $tour$
```

4.3.2 DADCA-parted

In this DADCA variation, we create the *Original Tour* in two steps, as presented in Algorithm 2, which is polynomial and deterministic. First, the coverage area is split into two geographic parts, as we illustrate in Figure 4.6. Lines 6 to 10 in Algorithm 2 represent the creation of two sets of CHs based on their coordinates.

After the two geographic sets of CHs are created, we perform the DADCA-greedy method in both sets. Thus, we merge both sets in the following lines (11 to 16) and create the *Original Tour* of DADCA-parted, as illustrated in Figure 4.6. The main idea of this variation of DADCA is to roughly place the GS in the center of the tour without any tour analysis. The role computation remains at a complexity of $\theta(p^2)$.

4.3.3 DADCA-LKH

DADCA-LKH is another variation to DADCA tour planning. For the best of our knowledge, the best method of computing solutions for the TSP problem in polynomial time is the Lin-Kernighan-Helsgaun (LKH) algorithm. The LKH algorithm is approximate and non-deterministic, but optimal solutions for TSP

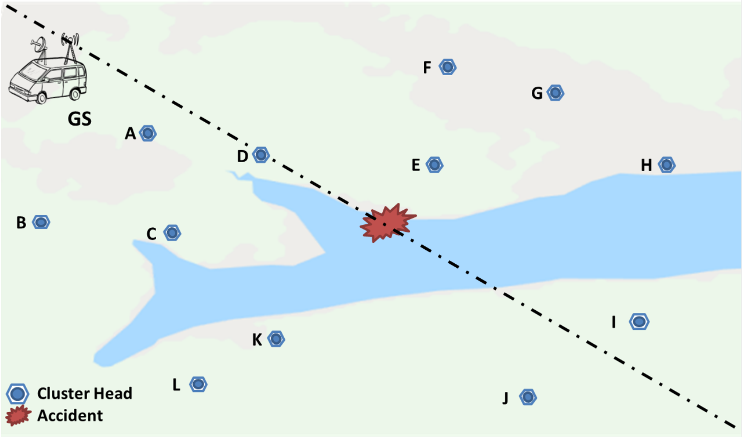


Figure 4.6: DADCA-Parted set of CH being splitted based on their coordinates.

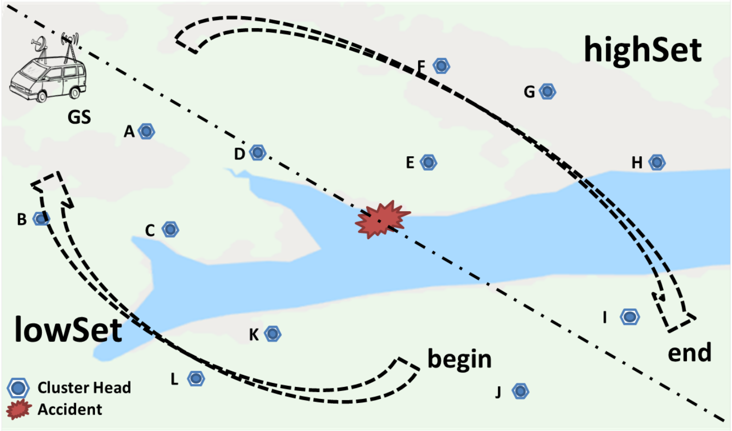


Figure 4.7: DADCA-Parted tour scheme.

are produced with high frequency, and it currently holds the record for all instances with unknown optima[1]. Its complexity is $\mathcal{O}(p^{2.2})$ [50][89].

In DADCA-LKH, LKH is used to compute the best tour that closes a Hamiltonian Cycle in polynomial time, as presented in Algorithm 3, but the tour obtained may not be necessarily the best tour that a TSP heuristic can find. Once this tour is found, DADCA-LKH removes the last edge before the GS to create a tour as exemplified in Figure 4.8.

It is reasonable to assume that DADCA-LKH can produce a smaller tour than a TSP-based strategy, because LKH is likely to find the same tour generated by a TSP-based strategy, and DADCA-LKH removes one edge, making the final tour smaller. The main goals of this strategy are to maintain low processing complexity for the UAV and to improve the DADCA tour.

[1] https://goo.gl/RDYjFS

Algorithm 2: DADCA Parted - Tour Planning

```
function PartedTour(setOfCH):
    $tour \leftarrow null$
    $tourTmp \leftarrow null$
    $lowSet \leftarrow null$
    $highSet \leftarrow null$
    foreach clusterHead $CH \in setOfCH$ do
        if CH.Xcoord $\geq CH.Ycoord$ then
            $highSet$.add($CH$)
        else
            $lowSet$.add($CH$)
    $tourTmp \leftarrow$ GreedyTour ($lowSet$))
    $tourTmp$.ReverseOrder()
    $tour$.addInOrder($tourTmp$)            ▷ Faraway to GS
    $tourTmp \leftarrow$ GreedyTour ($highpSet$))
    $tourTmp$.RemoveFirst()                 ▷ Remove duplicated GS
    $tour$.addInOrder($tourTmp$)            ▷ GS fo faraway
    return tour
```

4.3.4 DADCA-LKH-cut

Finally, we propose another variation on DADCA named DADCA-LKH-Cut, which is described in Algorithm 4. This version computes its tour using the LKH algorithm but does not remove the final tour edge as in DADCA-LKH. Instead, DADCA-LKH-Cut removes the edge roughly in the middle of the Hamiltonian Cycle, as shown in Figure 4.9. DADCA-LKH-Cut is very similar to DADCA-LKH, but the data reaches the GS more often, once GS is not in one of the tour's extremities. This DADCA variation is non-deterministic as its also based on LKH.

A pertinent question regarding this specific strategy would relate to which edge we remove. For example, instead of the middle edge, we could remove the biggest edge, which would increase the chances of the tour be smaller than the one created through the TSP solution for such a set of CHs. However, we do not aim to produce a smaller tour but instead maintain the GS roughly in the middle of the tour. In addition, this algorithm produces more straightforward results regarding message delays.

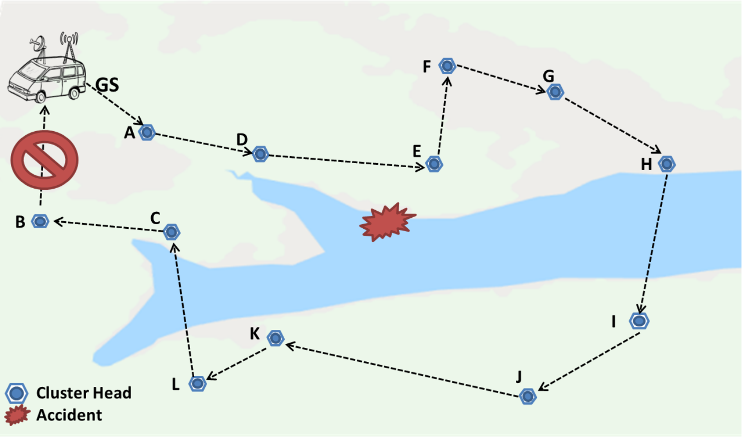


Figure 4.8: A sample tour that could be provided by DADCA-LKH.

Algorithm 3: DADCA LKH - Tour Planning

```
function $LKHTour(setOfCH)$:
    $tour \leftarrow null$
    $tourTmp \leftarrow null$
    import LKHSolver() as solver
    $tourTMP$.add($GS$)
    $tourTMP$.add($setOfCH$)
    $tour$.addInOrder(SOLVER ($tour$))
    $tour$.removeLastEdge()
    return $tour$
```

4.3.5 DADCA - tour coverage

In short, all UAVs have the same behavior: They go from one extreme to the other extreme of a tour and return, such tour passes only once through all CHs; UAVs collect data from CHs and deliver to GS based on UAVs passages over that. When there is more than one UAV in operation, the UAVs try to divide the workload by the number of UAVs in operation. This division of the workload only occurs with the meeting of UAVs in pairs.

4.3.5.1 DADCA - initialization

Once any of the DADCA algorithms from sections 4.3.1 to 4.3.4 creates a tour, the next phase begins (as indicated in Figure.4.5.b). This second phase extends and adapts the algorithm originally proposed by Kingston et al. [90]. Their algorithm controls a set of UAVs surveilling a linear path in $\mathbb{R}^2$ that represents a boundary, such as a frontier or a pipeline. The path in [90] cannot cross itself.

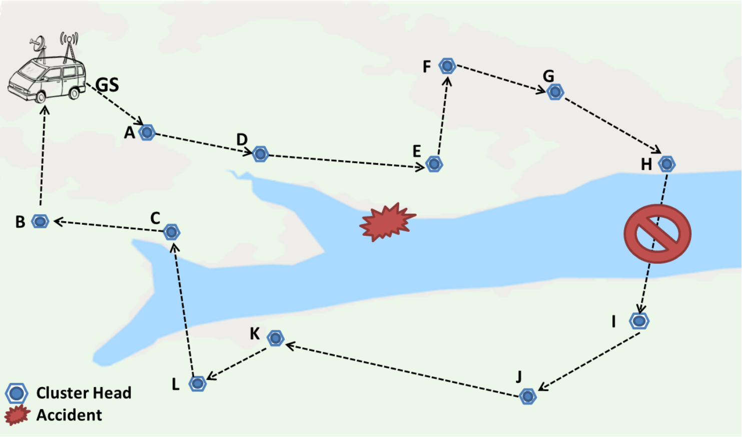


Figure 4.9: A tour computed by LKH without a middle edge to be used on DADCA-LKH-cut.

The first step of DADCA is the initialization of $GS$and $UAVs$, as presented in Algorithm 5. The GS begins informing each UAV of the CHs' geographic locations and flagging the UAV flag *free* as false. The flag *free* is used by each UAV to control their (re)entrance in a data collection activity that has already begun.

At the initialization of each UAV in Algorithm 5, its variables are initialized. Function *RunStrategy()* is responsible for calling one of the planned tour from the algorithms presented in sections 4.3.1 to 4.3.4. What we call *Original Tour* is the first tour computed by a UAV from a *setOfCH*. Once the DADCA tours form a single line that pass through all CHs, we call the first limit of the *Original Tour* our *left limit* and the end the *right limit*.

Furthermore, the variables *rightNeighbors*, *leftNeighbors*, and *totalNeighbors* begin at zero. We use these variables to control the number of other UAVs that each UAV knows are working at the moment t of T. The variable *leftNeighbors* of a UAV$_A$ denotes the number of UAVs working properly between UAV$_A$ and the *left limit*, while *rightNeighbors* denotes the number of UAVs working properly between UAV$_A$ and the *right limit*. These variables are updated when a rendezvous occurs between two UAVs or when a UAV reaches one of the tour limits.

4.3.5.2 DADCA - behaviour

When a UAV is launched, it acts as it would working in isolation, no matter how many UAVs the GS is launching. This UAV acting in isolation will use its *Original Tour* to collect data from all CHs. Figure 4.10 illustrates the UAV$_A$ traveling from the *left limit* (A) to the *right limit* (E) and collecting data. When UAV$_A$ reaches any limit, it returns from the opposite direction.

Algorithm 4: DADCA LKH Cut - Tour Planning

```
function LKHTourCut(setOfCH):
    tour ← null
    tourLKH ← null
    tourTMP.add(GS)
    tourTMP.add(setOfCH)
    tourLKH ← LKHTour (tourTMP)
    middleEdge ← GetMiddleEdge (tourLKH)
    leftCH ← middleEdge.left()            ▷ Nearest from GS in Tour
    rightCH ← middleEdge.right()          ▷ Far from GS in Tour
    for i ← tourLKH[rightCH] to tourLKH.size by 1 do
        tour.add(tourLKH[i])              ▷ From new first CH until end
    for i ← tourLKH[GS] to tourLKH[leftCH] by 1 do
        tour.add(tourLKH[i])              ▷ From GS to new last CH
    return tour
```

Algorithm 5: Initialization processes: GS and UAVs

```
procedure GSInit (setOfUAV;setOfCH ):
    SendMsgTo setOfUAV with setOfCH
    Launch setOfUAV
procedure UAVInit (setOfCH):
    this.free ← false
    this.tour ← RunStrategy (setOfCH)
    this.rightNeighbors ← 0
    this.leftNeighbors ← 0
    this.totalNeighbors ← 0
```

Every time a UAV passes over the GS, it delivers the collected data. Adding the GS to the tour in all four planning algorithm allows for this passage over the GS. In the example of Figure 4.10, GS could be illustrated by A to E varying of the tour planning being in use. The UAV$_A$ remain in that situation until T ends or a rendezvous occurs with another UAV. Each UAV is controlled by Algorithm 6 and Algorithm 7, which we describe in the following paragraphs.

Eventually, other UAVs perform their collection, as illustrated in Figure 4.11. Two UAVs may also fly close enough to be within radio range of each other, and at this point both perform a rendezvous.

DADCA only performs rendezvous in pairs, which means that if three or more UAVs are close enough to perform a rendezvous, each UAV will ignore any UAV after the first UAV it discovers from a logical point of view. For

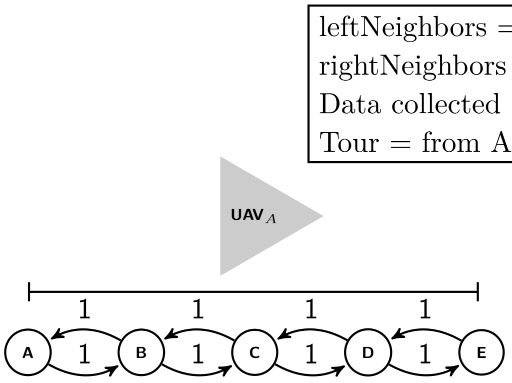


Figure 4.10: A possible solution of presented graph by $DADCA$ with a single UAV collection data from all CH. The logical segment above CHs represents the UAV tour from A to E.

instance, UAV$_A$, UAV$_B$ and UAV$_C$ may all be close enough to each other to perform a rendezvous. There are six possible scenarios of mutual discovery among the three UAVs, but these scenarios can illustrated in just two cases:

1. Not feasible: UAV$_A$ discovers UAV$_B$ before discovering any other UAV, UAV$_B$ discovers UAV$_C$ before discovering any other UAV, and UAV$_C$ discovers UAV$_A$ before discovering any other UAV. In these scenarios, no pair is available to perform a valid rendezvous. In this case, all three UAVs will ignore each other.

2. Feasible: UAV$_A$ discovers UAV$_B$ before discovering any other UAV, UAV$_B$ discovers UAV$_A$ before discovering any other UAV, and UAV$_C$ discovers either UAV before discovering the other UAV. In this case, UAV$_A$ and UAV$_B$ will perform a DADCA rendezvous.

Apart from the requisite logical pairing described above, we introduce another restriction to enable a valid rendezvous: two UAVs can only perform a rendezvous if and only if UAV$_A$'s origin CH is UAV$_B$'s destination CH and *vice versa*. This requirement extends the original idea presented in [90] of a linear non-crossing path in $\mathbb{R}^2$ enabling tours in $\mathbb{R}^3$. When UAVs follow the logical meeting order, it is possible that tours will cross themselves.

4.3.5.3 DADCA - rendezvous

If more than one UAV is active and two UAVs are moving in opposite directions on the *Original Tour*, they will eventually be within radio range, as shown in Figure 4.12. This meeting is what we call a rendezvous—at this moment, the UAVs exchange current information about other known UAVs,

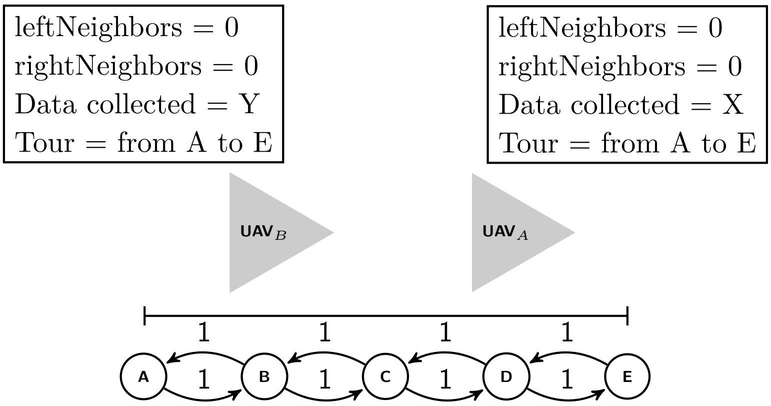


Figure 4.11: Two UAVs flying far from each other.

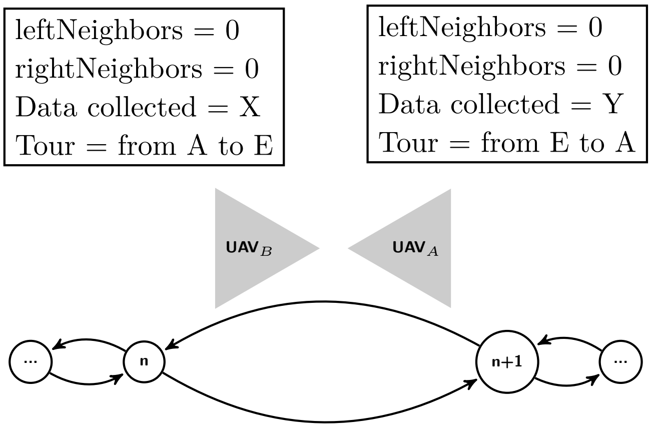


Figure 4.12: Two UAVs a moment **before** a rendezvous. Each one with an independent tour and no references about other UAVs until this moment.

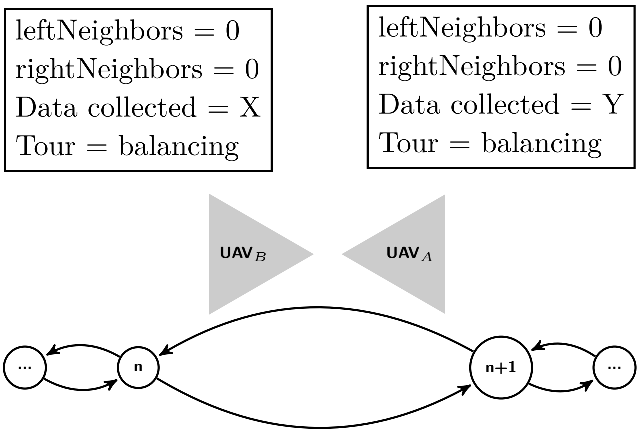
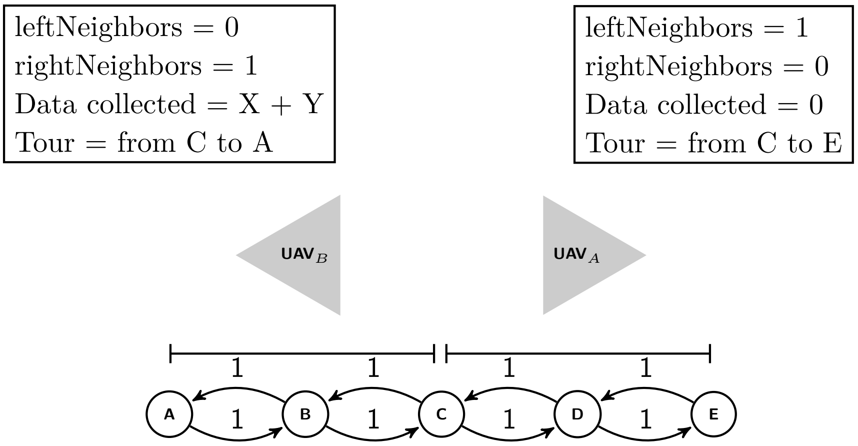


Figure 4.13: Two UAVs running DADCA a moment **during** a rendezvous. Each one updating its data with data from the other.



leftNeighbors = 0
rightNeighbors = 1
Data collected = X + Y
Tour = from C to A

leftNeighbors = 1
rightNeighbors = 0
Data collected = 0
Tour = from C to E

Figure 4.14: Two UAV running DADCA after defined the shared border for the last rendezvous. Is this case, sink node C. Both go to C and then go to opposite sides. Logical Left UAV carries data from Right UAV.

the data they've collected, and then adjust their sub-tours accordingly. These rendezvous runs are presented in Algorithm 6 and in Algorithm 7.

To explain this rendezvous, let UAV$_A$ be the first UAV sent and let UAV$_B$ be the second. When UAV$_A$ begins collecting data, its tour is from A to E; when it reaches E, it begins its return. At some point, UAV$_B$ begins the same *Original Tour*, as shown in Figure 4.11.

Figure 4.13 presents two UAV performing a rendezvous. They exchange their metadata, and are capable of understanding which UAV originates from the nearest side of the GS. This UAV is called the left UAV and the other is the right UAV. Both UAVs update each other on the number of working UAVs that they're aware of by using the other UAV to update their information in the opposite direction. This protocol means that *rightNeighbors* and *leftNeighbors* are updated based on the information relayed during the rendezvous. Thus, the left UAV (UAV$_B$) receives the other UAV's data to deliver to the GS because it is closer, and the right UAV (UAV$_A$) changes its direction.

Both UAVs compute new sub-tours to take into account the ideal division of the *Original Tour* with known available UAVs working at that moment. Therefore, both UAVs compute the same *SharedBorder*, which refers to the theoretical point where both sub-tours meet regarding the number of CHs. As a result, both UAVs navigate to the *SharedBorder* and turn in opposite directions, as presented in Figure 4.14. At this point, both UAVs update their metadata and one UAV transfers its collected data.

The UAVs remain in such a pattern until they reach a tour limit, rendezvous with other UAVs, encounter a malfunction, or the period T ends. As discussed above, it is important to note that whenever a UAV passes within range of the GS, it delivers all the data in its possession.

4.3.5.4 DADCA - algorithm

In order to better describe DADCA in Algorithm 6, we present four possible rendezvous cases from the perspective of the left UAV which is presented as v_{left}. Each UAV have some internal control variables:

- *free*: stands for the UAV in (re)entrance;
- *next*: stands for the ID of next CH to visit;
- *last*: stands for the ID of last CH visited;
- *leftNeighbors*: known number of UAVs from left;
- *rightNeighbors*: known number of UAVs from right;
- *tour*: actual tour;
- *setOfCHs*: whole set of CHs;
- *originalTour*: first tour computed.

Line 2 to line 6 filters cases in which two UAVs have already begun their data collection, otherwise the scenarios start from line 7 which at least one of the UAVs are (re)entering.

Line 3 shows cases in which both UAVs are effectively coming from opposite directions. If both UAVs are close enough to perform a balance but are not on consecutive sub-tours, then the rendezvous is ignored, as in the case of line 6. It is necessary to ignore this rendezvous for crossing tours—an option that was not possible in Kingston's original proposal [90].

In Algorithm 6 line 7, the cases is presented where at least one UAV is (re)entering activity. If two (re)entering UAVs attempt to perform a rendezvous, then the shared information becomes ignored. A (re)entering UAV therefore acts in isolation until it finds an older UAV collecting data.

A UAV (re)entering a data collection that has already begun must meet another UAV in activity. Line 8 indicates cases when only one UAV is (re)entering activity and both UAVs perform the rendezvous. The *free* UAV will use the metadata from the older one in order to function as a working UAV. Therefore, both UAVs perform a regular rendezvous. Lines 18 and 21 represent cases when the UAVs are updated as they reach the first or last CH from the original tour. Line 25 refers to a case in which a UAV reinforces data collection, which can occur when a new UAV is sent to collect data or a temporarily disabled UAV returns to work at any point of the *Original Tour*.

The algorithm 7 in its routine *Rendezvous* is responsible for rebalancing and recalculating a UAV sub-tour. During the period T, all UAVs perform their balancing function to (re)establish the whole system balance. Each UAV reuses the *Original Tour* length and divides it by the updated number of known working UAVs upon a rendezvous. Subsequently, each UAV generates its segments S_{left} and S_{right} and its theoretical intersection b ($S_{left} \cap S_{right}$). In this work, the *RecalculateBalancedSegments()* is the rounded median of the number of CHs in the tour.

Given the research of Kingston et al., this visitation model has been adapted from their research. However, the UAVs did not know their tour before commencing their flight. In consequence, the model does not use a linear path, but instead creates a path by visiting CHs. In $DADCA$ each UAV receives an unordered list of CHs from the GS and creates their tours without any GS processing. Our extension is capable of (re)calculating their tours in $\mathbb{R}^3$ with tour intersections.

Finally, in terms of resilience and malfunctions, we can receive new UAVs at any point in the tour, rather than only the beginning and end of the tour, as in [90]. It is remarkable that is not necessary that UAVs have the same processing capabilities. Each UAV can compute the result of the *Original Tour* at distinct moments, and all *Original Tour* will be the same. Furthermore, each UAV can begin its data collection at distinct moments, and the entire system will adjust its behavior.

4.3.6 Correctness and convergence

In this section, we discuss the correctness and convergence of DADCA. Correctness is defined as the extent to which DADCA enables total coverage of the WSN without leaving any CHs unvisited or any UAVs stuck in a state that does not cover part of a tour. Convergence means that eventually all UAVs have split the original tour in parts that demand approximately the same time for the CH visits.

Correctness: Regardless of the number of UAVs in operation, an UAV will go in one direction on its original tour until the one of the following conditions occurs: (1) it reaches the end of the tour; or (2) it meets with another UAV that is ready to perform a valid rendezvous. In the first case, even if there is only one UAV collecting data, it will cover the entire original tour, always turning around and reversing the flight direction to the extremes. Even if there are multiple UAVs and they cannot perform a rendezvous, they will behave in the same way, so all CHs set by the original tour will have data collected, with no CHs uncovered.

Algorithm 6: Events - from the v_{left} perspective

```
on UAV $v_{left}$ (left) rendezvous $v_{right}$ (right):
    if $v_{left}.free = false$ and $v_{right}.free = false$ then
        if $v_{left}.next = v_{right}.last$ then
            Rendezvous ($v_{left}$,$v_{right}$)                    ▷ Wished rendezvous
        else
            // ignored because they are not consecutive UAVs, listed for
                explanations purpose
    else
        if $v_{left}.free = true$ and $v_{right}.free = true$ then
            // ignored because a free UAV needs to rendezvous with an
                active UAV to do a rendezvous, listed for explanations
                purpose
        else
            // rendezvous between and active UAV and a $free$ one.
            // The active one updates the new one.
            // Lets call them $v_{left}$ and $v_{free}$
            $v_{free}.tour \leftarrow$ Reverse($v_{left}.originalTour$)
            $v_{free}.setOfCHs \leftarrow v_{left}.setOfCHs$
            $v_{free}.last \leftarrow v_{left}.next$
            $v_{free}.next \leftarrow v_{left}.last$
            $v_{free}.rightNeighbors \leftarrow v_{left}.rightNeighbors$
            $v_{free}.leftNeighbors \leftarrow v_{left}.leftNeighbors + 1$
            $v_{left}.rightNeighbors \leftarrow v_{left}.rightNeighbors + 1$
            $v_{free}.free \leftarrow$ false
            Rendezvous ($v_{left}$,$v_{free}$)
on UAV $v_i$ reaches right limit:
    $this.rightNeighbors \leftarrow 0$
    InvertTour ()
on UAV $v_i$ reaches left limit:
    $this.leftNeighbors \leftarrow 0$
    InvertTour ()
on UAV $v_i$ defeats or resets:
    UAVinitialization ($this.setOfCHs$)
    $this.free \leftarrow$ true
```

Algorithm 7: Regular rendezvous with v_{left} and v_{right} on v_{left} perspective

```
procedure RENDEZVOUS ($v_{left}$,$v_{right}$):
    $v_{left}.rightNeighbors \leftarrow v_{right}.rightNeighbors + 1$
    $v_{left}.totalNeighbors \leftarrow v_{left}.rightNeighbors + v_{left}.leftNeighbors + 1$
    SharedBorder $b \leftarrow$ RECALCULATEBALANCEDSEGMENTS ()
    $v_{left}$ flies along $v_{right}$ to $b$
    each UAV sets direction to its own segment and direction
```

In the second case, when one UAV encounters another UAV, they will update their collected sensor information so that the leftmost UAV will always receive all the data collected by the other UAV engaged in the rendezvous in order to avoid causal order issues. Both UAVs will mutually update their collected datasets (from WSNs) in a way that will generate the same shared boundary. Once the boundary is established, both will fly to the boundary and ignore any other UAVs, thus preventing a rendezvous. Eventually, along this boundary, they will move to opposite sides. From this point on, only case (1) or case (2) can occur again, so there is no way for a UAV to become locked-in a internal status with regard to coverage.

Convergence: We define convergence of DADCA in the meaning that eventually all UAV will be flying over the same segment of the original tour and reaching other UAVs at the same positions. The convergence status occurs after a balance/adjusting period which UAVs new computes shared borders, and UAVs does not have the correct knowledge of the total number of working UAVs. When the convergence is reached UAVs always calculate sharing borders in the same position and remain as is until an UAV leaves or reenters the set of working UAVs.

The convergence of DADCA firstly can be demonstrated by reducing it to the problem to linear path coverage by UAVs, omitting the data collection as shown by Kingston et al.[91]. In this case, all UAVs will move due to the tour and rendezvous. This means that the UAVs will only move on a rectilinear path that can be characterized by two endpoints (the GS and the end of the tour) without any intersections with the same tour in a causal order perspective. The tour determines a natural sequence of UAVs along the path, where each UAV only moves along one section of the tour, which is bounded by the sections flown by the “left” and “right” neighbouring UAVs. However, recall that the boundaries are flexible and depend on the moment that neighbouring UAVs meet for a rendezvous.

This reduction allows the problem to be solved as the linear path surveillance problem[90]. McLain et al [92][93] introduced the coordination variable logic early on, and Kingston et al. [91][90] attempted to demonstrate convergence in a meaning that at some point all UAVs will have shared borders in fixed positions with the whole group staying the same. The coordination variable logic consists of a cooperative control approach for teams of vehicles that are dependent on the environment or mission scenario in which the vehicles are acting[92]. Later, McLain et al. [93] proposed a solution strategy for achieving cooperative timing among teams of vehicles to cover a path by splitting it autonomously without global information exchange. Let v be a fixed number of UAVs, let K be the original tour length, and let $W \leftarrow \frac{K}{v}$ be the period that a UAV v needs to cover the entire tour K once. Thus, the adjustment phase is reached after 2W as the proof presented in [91]. The worst-case scenario of 2W occurs when all UAVs are stacked infinitesimally close to one end of the perimeter and are traveling toward the other end. In general, with regard to UAV positions, the adjustment phase ends before W [91].

The DADCA's convergence demonstration relies on three critical pieces of information that each UAV has: (1) the original tour; (2) the number of UAVs on the left side of the perimeter relative to a given UAV_i, and (3) the number of UAVs on the right side of the perimeter relative to a given UAV_i. It also assumes that all UAVs fly at constant speeds. The proof is argumentative as follows:

1. assume that the number of working UAVs will not change for a finite period M;
2. let K be the original tour length being covered by UAVs;
3. let UAV_A(ln,rn) be the tuple of known UAVs from left (ln) and right (rn) sides of the original tour that is known by UAV_A;
4. UAV_A will start to fly as $UAV_A(0,0)$ from the left side to the right side of the original tour;
5. there is at least another $UAV_B(x,y)$ working;
6. $UAV_A(0,0)$ eventually will rendezvous the $UAV_B(x,y)$;
7. both UAVs will update their control variables for $UAV_A(0,y+1)$ and $UAV_B(1,y)$;

8. both UAVs will compute the shared border b by dividing K by the total of known UAVs. The shared border is obtained by $b \leftarrow \frac{K}{y+2}$;

9. both UAVs will fly to b ignoring other UAVs;

10. UAV_A will fly to the left side of b, and UAV_B will fly to the right side of b;

11. if and only if there are only UAV_A and UAV_B working, the system will be balanced and the convergence is reached;

12. otherwise, if there is at least an $UAV_C(z, w)$, that may or may not pass by UAV_A and UAV_B being ignored during their rendezvous;

13. if $UAV_C(z, w)$ did not pass by both UAVs during their rendezvous:

    (a) in this case, UAV_A will have the most updated number of UAVs in their right side due to its rendezvous with UAV_B;

    (b) UAV_B will have the most updated number of UAVs in their left side due to its rendezvous with UAV_A;

    eventually $UAV_C(z, w)$ will rendezvous $UAV_B(1, y)$ in the right side of last b calculated by UAV_B. The UAV_B will update UAV_C, and both UAVs will know about all UAVs working;

    (c) $UAV_B(1, w + 1)$ and $UAV_C(2, w)$ will calculate a new b_2 taking into consideration all UAVs working;

    (d) $UAV_B(1, y)$ will fly to the left side of b_2 and $UAV_C(2, w)$ will fly to the right side of b_2;

    (e) if there are only UAV_A, UAV_B and UAV_C working, then w will zero, and the system will be balanced;

    (f) the knowledge in UAVs will be $UAV_A(0, 2)$, $UAV_B(1, 1)$ and $UAV_C(2, 0)$;

14. if $UAV_C(z, w)$ did pass by both UAV_A and UAV_B during their rendezvous:

    (a) $UAV_C(z, w)$ will reach the left end of original tour and change its knowledge to $UAV_C(0, w)$;

    (b) eventually $UAV_C(0, w)$ will rendezvous $UAV_A(0, y + 1)$ in the left side of last b calculated by UAV_A. The UAV_A will update UAV_C and both UAVs will have the knowlegde of all UAVs working as $UAV_C(0, y + 2)$ and $UAV_A(1, y + 1)$;

(c) $UAV_C(0, y+2)$ and $UAV_A(1, y+1)$ will calculate a new b_3 taking into consideration all UAVs working;

(d) $UAV_C(0, y+2)$ will fly to the left side of b_3 and $UAV_A(1, y+1)$ will flies to the right side of b_3;

(e) if there are only UAV_A, UAV_B and UAV_C working, then eventually $UAV_A(1, y+1)$ will perform a rendezvous with $UAV_B(1, 0)$ and update both updates themselves to $UAV_A(1, 1)$ will rendezvous $UAV_B(2, 0)$ and the system will be balanced;

(f) the knowledge in UAVs will be $UAV_C(0, 2)$, $UAV_A(1, 1)$ and $UAV_B(2, 0)$;

Every rendezvous updates the knowledge of involved UAVs with fresher information from the opposite side of its movement making a step forward in the convergence. If a third UAV does not pass by a pair of UAVs during a rendezvous, the whole system will fastly converge.

However, if a third UAV passes by a pair of UAVs during a rendezvous, it makes information from both UAVs in the rendezvous not up to date regarding the whole set of UAVs. Such third UAV would makes the overall coordination variables not up to date until its next rendezvous. As all UAVs update its information about neighbors when reaches a tour limit, this feature always will correct the two UAVs in the extremes of the original tour and these two UAVs will update and correct UAVs knowledge between them.

4.3.7 DADCA limitations

This subsection presents two possible drawbacks of DADCA. The first one we name as Shadow Effect and is caused by the temporary joint flight of two UAVs that are moving towards a shared border just after a rendezvous and ignoring other UAVs. We name the second possible drawback as Geographic Issue and it is related to the geographic CH distribution on the covered area.

4.3.7.1 Shadow Effect

What we have named the Shadow Effect (SE) is the total accumulated flight distance of all UAVs when flying in pairs, during which only one of the two UAVs will collect data from CHs. As presented in section 4.3.6, the set of UAVs converges to a stable oscillatory behavior, in which the UAVs do not need to calculate any shared border to move torwards. Instead, the UAVs meet at the same time at the shared borders and switch their directions in 180 degrees (a U-turn).

Every time DADCA is in its convergence phase, at least two UAVs are moving toward a shared border. During this, the two UAVs cover one segment of the original tour. When two UAVs are flying side-by-side over the same segment of the tour, they ignore all other UAVs but still consider the CHs on the ground. The two UAVs in co-movement ignore all other UAVs because both UAVs still will be in a rendezvous.

Rendezvous occurs only in pairs to preserve the rendezvous effectiveness; it means to guarantee that at least two UAVs of a whole set will split the original tour at least once as argued in section 4.3.6. In other words, the rendezvous is performed only in pairs because it is necessary for DADCA correctness. As the original tour has only two ends, if more than two UAVs meet each other, it would be necessary to rely on an external variable (ie. GPS, clock, etc) to prioritize which internal variable would be updated in all UAVs involved in such rendezvous as sketched by McLain et al. [92][93].

Even with two UAVs flying along each other to a shared border during a rendezvous, a CH delivers its data to only a single UAV at a time, as stipulated by our system model in section 3.2. When two UAVs fly together over a segment of the tour that includes a CH, one of the two UAVs will not receive data from the CH. At the end of the rendevouz, data will only have been collected from a single UAV. In other words, concerning TD^T, one of the two UAVs does not collect data.

The SE occurs primarily at the beginning of the data collection and to a minor extent when UAVs leave or reinforce the data collection. The SE is a born (natural) inefficiency of DADCA variations and does not occur in the FPPWR and TSP-based strategies, as the UAVs do not fly together in the latter strategies.

To illustrate and discuss the SE, Figure 4.15 illustrates a hypothetical and extreme case of SE. The lines from t_0 to t_{10} denote sequential logical moments of data collection with three UAVs. Each UAV is identified by A, B, or C and an the ordered pair (t_n, rn); taking into consideration a UAV, the t_n is the number of known UAVs working on the left side and rn is the number of known UAVs working on the right side. For example, UAV_A(1,0) stands for the UAV named A that knows that there is 1 (one) UAV working between UAV_A itself and the leftmost limit of the original tour, in the same way, UAV_A knows that there is 0 (zero) UAVs working between UAV_A itself and the end of the original tour. The left side of each line represents the GS and the right side the length of a DADCA tour (K).

In the best-case scenario, it is possible that no SE occurs at all. In such scenario, all UAVs meet each other only in pairs flying over a shared borders at the same time them both UAVs would perform a rendezvous and turn back.

The worst possible scenario, in other words, the largest possible SE, is shown in Figure 4.15. This worst case could occur when all UAVs are heaped up one end of the tour and move without considering the existence of other UAVs. The following paragraphs describe every line of the trace table detail:

Step t_0 : Three UAVs are at one end of the tour and are traveling without considering the existence of other UAVs;

Step t_1 : UAV_A switches direction from the end of the tour and finds UAV_B. Both UAVs are available-it means, not perfoming a rendezvous and perform a rendezvous; at this moment the UAVs share their sensor data and the knowledge of other UAVs to UAV_A(1,0) and UAV_B(0,1);

Step t_2 : UAV_A and UAV_B compute the shared border K/2, because both UAVs recognize that there only two UAVs working at that time;

Step t_3 : UAV_A and UAV_B fly from K to K/2, ignoring UAV_C; UAV_C reaches K and begins its movement to GS;

Step t_4 : UAV_B reaches K/2 and starts to move towards to GS. When UAV_A reaches K/2 and begins moving toward K, UAV_C reaches K/2; UAV_A and UAV_C then perform a rendezvous, updating UAV_A(1,1) and UAV_C(2,0); at this moment, UAV_A and UAV_C compute the segment for the 2K/3 border, because both are aware that three UAVs are working at that time;

Step t_5 : UAV_A and UAV_C arrive at 2K/3; UAV_A turns to go to GS and UAV_C to K;

Step t_6 : UAV_B arrives at GS and will turn to go to K; UAV_A continues going to GS, and UAV_C arrives at K and will turn to go to GS;

Step t_7 : UAV_A and UAV_B reach each other at K/6 and perform a rendezvous, updating UAV_A(1,1) and UAV_B(0,2); both UAVs fly to shared border K/3;

Step t_8 : UAV_A and UAV_B reach the shared border; UAV_A goes to K, and UAV_B turns to go to GS;

Step t_9 : UAV_A and UAV_C reach each other at K/2 and perform a rendezvous without updating each other; UAV_A and UAV_C go to shared border 2K/3;

Step t_{10} : UAV_A and UAV_C reach the shared border 2K/3; UAV_A turns its direction to move towards to GS; UAV_C goes to K, and UAV_B reached GS and turns to go to K; from this moment until any UAV leaves, the UAVs will always meet each other at the shared borders.

The SE presented in Figure 4.15 is calculated in equation (6) below. If v is the velocity of a UAV, the amount of sensor data that is collected due to each shadowed UAV movement is SE/v. The effect of this inefficiency cannot be calculated only relying on K or the number of UAVs. The impact of the SE also depends on the number of CHs present in each segment in which the SE occurs.

$$SE = (K - \frac{K}{2}) + (\frac{2K}{3} - \frac{K}{2}) + (\frac{K}{3} - \frac{K}{6}) + (\frac{2K}{3} - \frac{K}{2}) = K \quad (6)$$

The SE always occurs whenever any of the DADCA-based algorithms are balancing after a UAV (re)entrance but in different intensity due to the position of a possible (re)entrance. The worst-case scenario always occurs at the beginning of a data collection period, when none of the UAVs have information on the other UAVs.

4.3.7.2 Geographic issues

Both DADCA-LKH and DADCA-LKH-Cut remove one edge from the tour generated by LKH, but neither strategy takes into account the size of the edge. In order to maintain the simplicity, there is no heuristic to select an edge to remove; only its position is considered. DADCA-LKH always removes the last edge from the generated Hamiltonian cycle, while DADCA-LKH-Cut removes the edge located roughly in the center of a Hamiltonian cycle. The absence of a heuristic to select the removable edges would add to the inefficiency of the DADCA variations compared to the TSP-based strategies.

As an ilustrative example, Figure 4.16 depicts a hypothetical CH distribution with a tour that is the TSP solution for this set of CHs. Let this tour be

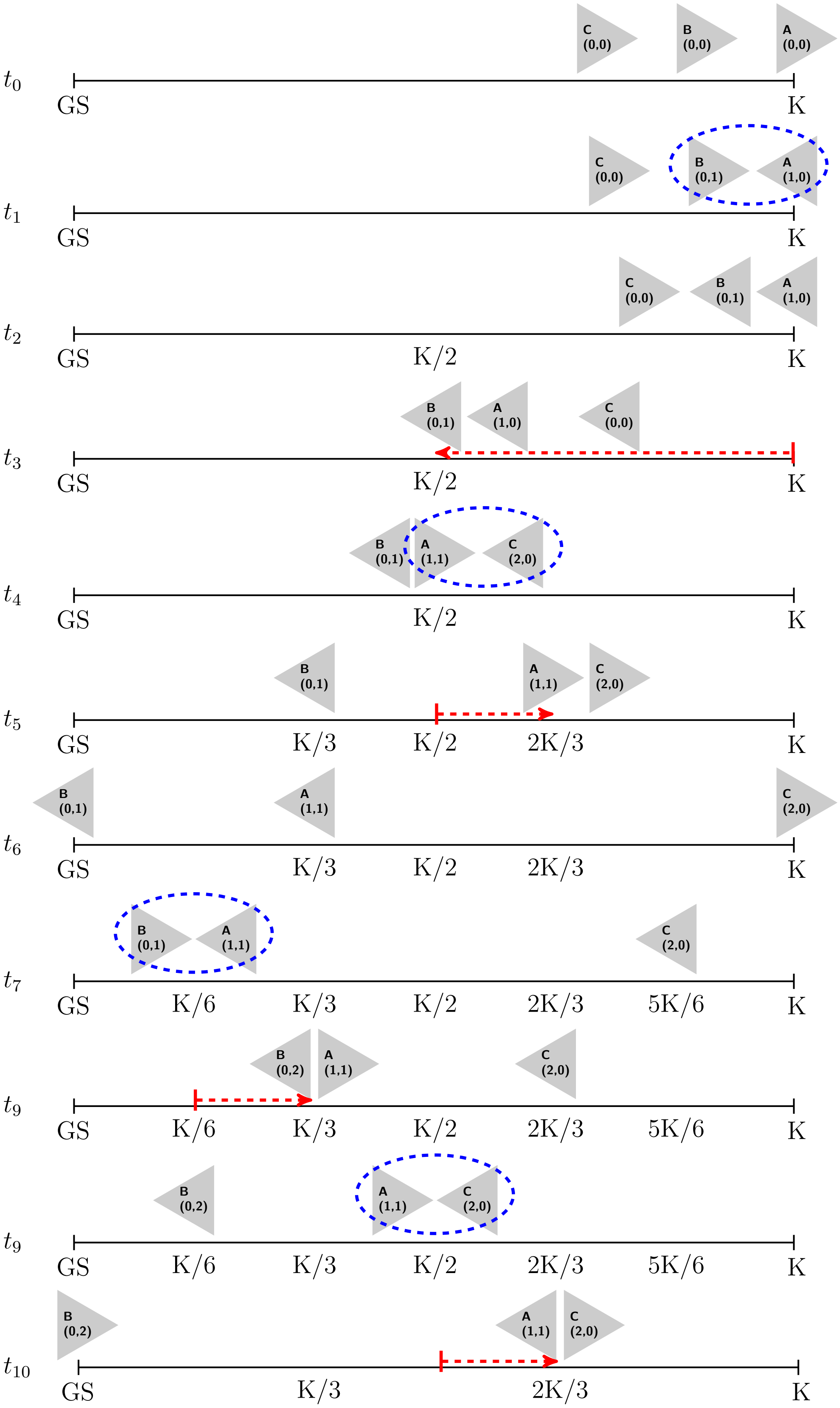


Figure 4.15: Shadow Effect: trace table and demonstration.

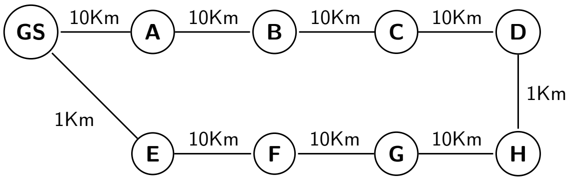


Figure 4.16: Hypothetical best tour provided by an TSP and LKH solutions.

the same found by the LKH heuristic, in this case a tour with a Hamiltonian cycle of length 72km as shown in the Figure 4.16.

The DADCA-LKH strategy would produce a tour by removing the edge $\overline{GSE}$, and the DADCA-LKH-Cut strategy would produce its tour by removing the edge $\overline{DH}$. Both strategies would produce a slightly shorter tour with 71km, which is around 1.4% shorter than the one of the TSP-based strategy.

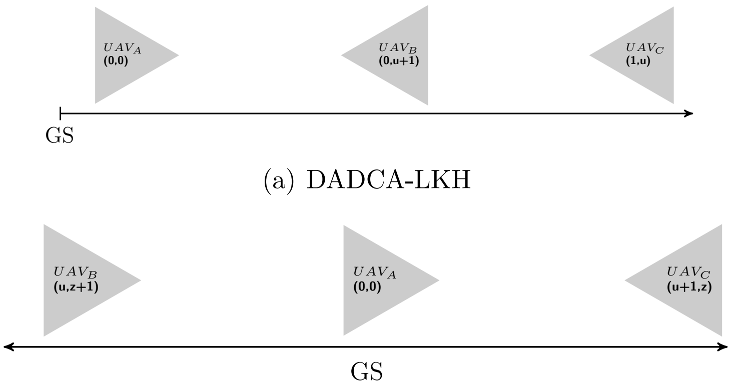
The difference of 1.4% does not determine which strategy produces better results in terms of TD^T. If during a period of T no UAV reenters the group there will not be SE due to the reentrance, but only an initial SE when the all UAVs initiate their mission. If and only if the length difference of the tour lengths of DADCA and other strategies would be smaller than the SE, DADCA will provide better results.It is important to note that the difference between tours from DADCA-LKH and TSP-based strategy must be calculated concerning the whole period T because of the dislocation difference will be applied for each time that a single UAV would perform a Hamiltonian cycle. However, it is reasonable to predict that a DADCA tour based on the LKH will be close to a tour obtained through TSP, but less efficient due to the SE caused by several (re)entrances of UAVs during a long period T.

4.3.8 Analysis of DADCA-LKH vs DADCA-LKH-Cut

We expect that DADCA-LKH would produce better results than DADCA-LKH-Cut in terms of TD^T. The difference between DADCA-LKH and DADCA-LKH-Cut occurs because DADCA-LKH-Cut leads to a longer SE than does DADCA-LKH. We illustrate this outcome in Figure 4.17 and explain it in the following paragraphs.

In Figure 4.17, (a) and (b) show UAV_A leaving the GS, for example, right after a battery recharge/exchange. At this moment UAV_A is not aware of any other UAV, which means its tuple of neighbors is (0,0).

In the following next two paragraphs we describe how DADCA-LKH and DADCA-LKH-Cut evolve.



(a) DADCA-LKH

(b) DADCA-LKH-Cut

Figure 4.17: Reentrance of UAV_A in operation in DADCA-LKH and DADCA-LKH-Cut strategies.

DADCA-LKH: UAV_A(0,0) will have a rendezvous with UAV_B(0,u+1); in such a rendezvous, UAV_A will receive updated information regarding the other UAVs in operation, represented by UAV_A(0,u+2); after the rendezvous, UAV_A and UAV_B will fly in tandem to the shared border at $\frac{K}{total}$, and then both UAVs will proceed in opposite directions and their data be updated with the most current information on the whole set of UAVs. In the best case scenario, UAV_B will propagate the new information to all other UAVs. In the opposite situation, in the worst case scenario, some UAV_C could pass by UAV_A, in which case UAV_B would be ignored, and more rendezvous would be necessary to balance the whole set of UAVs.

DADCA-LKH-Cut: UAV_A(0,0) will engage into a rendezvous with UAV_C(u+1,z). In this rendezvous, UAV_A will only be updated with the number of UAVs flying on the right-hand section of the tour. Both UAV_A and UAV_C will have UAV_A's information regarding the aircraft on the left to GS, which was reset to zero (due to its reentrance right after a battery recharge/exchange). Such information is older than in the analogous case of DADCA-LKH which UAV_C would not lose the knowledge of the existence of UAV_B and would pass this knowledge to UAV_A.

Thus, the rendezvous between UAV_A and UAV_C will erase the knowledge about UAV_B inside both UAVs, creating the status of UAV_A(0,z+1) and UAV_C(1,z). Both UAV_A and UAV_C go to the shared border at $\frac{K}{total-u+1}$ and ignore any other UAVs. This situation requires at least two more rendezvous to balance and will inevitably generate more SE.

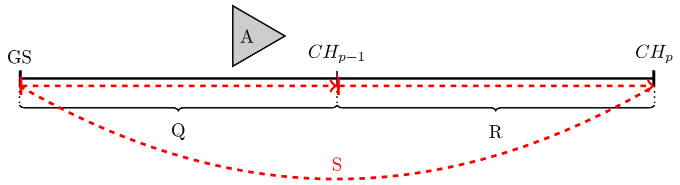


Figure 4.18: Illustration is presenting constant message delay of TSP-based and FPPWR strategies.

4.4 Expected impact of the number of UAVs on delays

The presented strategies will have different behavior regarding their susceptibility to the addition of UAVs. Such dependency will reflect on results regarding D_n^T with constant delays in the FPPWR and TSP-Based strategies unlike decreasing delays in the DADCA variations. In the following paragraphs, we explore this distinct behavior.

Constant delays: The TSP-based and FPPWR strategies have the same D_n^T, regardless of the number of UAVs in use. This is the case because each UAV follows the same tour on the cycle, and no messages are exchanged between UAVs. Figure 4.18 illustrates how the message delays occur in the TSP-Based and FPPWR strategies.

In Figure 4.18, there are p CHs in a tour. UAV_A collects data through the red dashed tour. UAV_A will proceed through a segment of length Q, until it reaches CH_{p-1} and collects data from CH_{p-1}. Thus, UAV_A will proceed to CH_p over a segment of length R and return to the GS over a segment of length S. The total ride of the collected data from CH_{p-1} will be carried by an UAV will be R + S. Let v be UAV_A's speed; then the delay in data collection from CH_{p-1} to GS will be (R + S)/v. Such a delay will be the same regardless of the number of UAVs in operation.

Decreasing delays: All DADCA variations present a decreasing delay when more UAVs are placed in operation. This shorter delay occurs because the ride of the collected data from a CH to the GS is not only a function of the tour size, but also the number of UAVs in operation. Here, we introduce the concept of Wasted Movement (WM)

Figure 4.19 illustrates the dynamics of a delay of data collection in the DADCA variations. Let u be the number of UAVs collecting data and K the total tour size, and consider that the whole set of UAVs are not in a adjusting period and UAV_A is the nearest UAV to the GS.

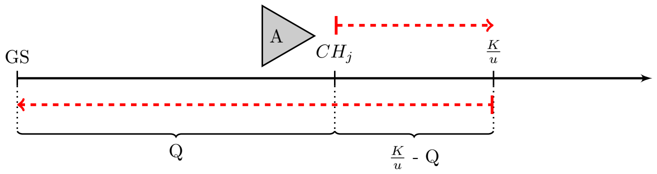


Figure 4.19: The illustration is presenting the Wasted Movement concept.

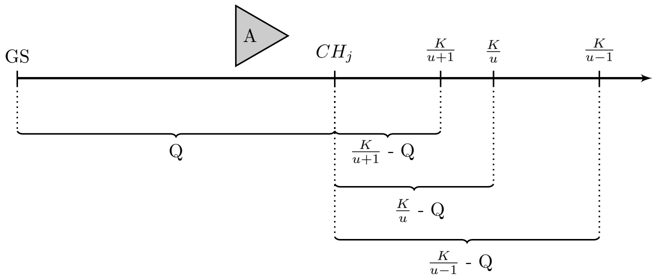


Figure 4.20: The illustration is presenting the Wasted Movement concept and its variation upon the number of UAVs and tour length.

UAV_A will proceed from GS through the segment the length of Q until it reaches CH_j and collects its data. Thus, UAV_A will proceed until it reaches its shared border at $\frac{K}{u}$. Thus, another UAV_x will reach it at the same time, and both UAVs will perform a rendezvous. After this rendezvous, UAV_A will return towards to GS to deliver all the data in has hoarded. It is important to note that UAV_A will collect data from all CH_j again when moving toward the GS.

The total time from the first visit of UAV_A over CH_j will be $\frac{2K}{u} - Q$. Thus the delay is mainly determined by the tour size (K) and the number of UAVs (u).

Figure 4.20 depicts three different possibilities causes that delays happen of DADCA variations when collecting data from CH_j. Let u be the number the UAVs in operation; after UAV_A receives data from CH_j, UAV_A will proceed to $\frac{K}{u}$ before returning to GS, as in the previous example.

Let us consider three set sizes of UAVs: u-1, u, and u+1. Each set of UAVs will produce a different shared border at $\frac{K}{u-1}$, $\frac{K}{u}$, and $\frac{K}{u+1}$. As the number of UAVs increases, the shared border moves closer to CH_j thus reducing the time any UAV moves from the border to the CH_j, and vice-versa.

The WM is the travel distance of data when it has already been collected but is not yet moving in the direction of the GS but just being hauled around. The WM is inversely proportional to the number of UAVs, so as more UAVs are put into operation, less WM occurs. For this reason, the delays decrease in the DADCA variation strategy when more UAVs are collecting data (always assuming that all UAVs move in an equal and constant speed).

5
Evaluation setup

This section presents the tests used to obtain the results presented in the next section. In order to test and compare the strategies mentioned in the last chapter, we implemented them in Sinalgo [94], a distributed algorithm simulation tool. The reason for using Sinalgo is that it provides support to implement the UAV movement and ad-hoc communication layers. All data generated were treated and analyzed in Python dataframes (Pandas) and reported using MatplotLIB [95] standards. Details for each set of simulation items are presented below. Table 5.1 summarizes all the variables used and their values. In this context, *flight endurance* stands for the UAV flight time capability.

5.1
Simulation model

For each simulation the same macro steps were used in an identical manner:

1. The GS arrives near the accident area and sends UAVs to collect data;

2. Every time a UAV is in radio range of the GS, it delivers its data;

3. Every time a UAV is runs out of flight endurance B, it flies to the GS and remains for a period of pt to charge its battery. After pt the UAV starts again as a new UAV leaving the GS;

4. The role process remains for a period T;

5.2
UAV

In order to anchor the simulations, we choose a UAV model capable of a flight endurance B of 16 hours. This period is much higher than actual experiments with off-the-shelf UAVs (i.e., *drones*). However, this choice is reasonable when compared to academic and professional equipment, which can have a flight time of up to 81 hours[96] of flight endurance as presented by [97].

Apart from reaching the limit of its flight endurance, any UAV can stop working at any time due to a malfunction. To simulate this issue, we adopted the concept of Mean Time Between Failures ($MTBF$) for the role simulation. We applied a $failrate$ for each simulation. The UAV's mobility characteristics are the same as Dubin's Car and are represented as a VTOL UAV. All UAVs fly at a constant speed s. We used four sets of UAVs in our simulations: 2, 4, 8, and 16 UAVs. The UAV's buffers are big enough to handle all data necessary for all flying time.

5.3 Radio

All radios were conceived as an 802.15.4 implementation XBee S1 [98], which has a radio range proximity of 100 meters. The message payload is 72 bytes at a rate of 250kb/s. The radio's energy consumption is negligible in these simulations. This radio was selected for this scenario due to two main factors: (1) it is light to carry onboard and (2) it consumes very little power.

5.4 Communication

The communication was not set as perfect in the simulations. Figure 3.5 illustrates a perfect scenario of data collection. However, more realistic details were introduced in the simulations. First, we selected a connectivity model that defines when two nodes are in communication range. The best-known examples are the unit disk graph (UDG) and the quasiUDG (QUDG). A QUDG percentage was set to reduce the radio range R, which signifies that the radio ranges were not perfect.

Second, we introduced interference situations using the signal-to-interference-plus-noise ratio (SINR), which assumes that the signal decays exponentially in relation to the Euclidean distance to the sender. Roughly speaking, SINR drops a message if the signal of the message at the receiver is below the sum of all interfering signals times a given constant. The QUDG and SINR allows the theoretical radio range R to be a statistically feasible range r, which may vary during the flight.

It is assumed that during a valid encounter of two UAVs the time which both will be within the radio range of each other will be enough so that all the data collected from one UAV can be transmitted to the other UAV that will load it on.

Table 5.1: Notation used in this work

Symbol	Definition	Used Values	Symbol	Definition	Used Values
p	Number of CH	70, 140 & 700	s	UAV speed	20m/s
v	Number of UAVs	2, 4, 8 & 16	T	Period of Data Collection	72 hours
A	CH Distribution Area	$700km^2$	pt	UAV (re)preparation time	10 min
h	Flight height	50 meters	B	UAV flight endurance	16 hours
R	Radio range	100 meters	MP	Message payload	72 bytes
S	Radio range path segment	71 meters	ρ	Radio transmition rate	250Kb/s
q	QUDG	10%	FT	UAV failTax	10%
c	Connection probability	90%	$MBTF$	UAV failPeriod MTBF	20 hours
α	SINR alfa	2	C_i	Data creation tax	1B/seg
β	SINR beta	0.7	γ	SINR noise	0.1

5.5 Solvers

The TSP solution is provided by the use of the Concorde solver [99]. To provide the LKH tour, we also used the original LKH implementation [100].

5.6 Simulated area

The physical simulated area A is set to $700km^2$, the size of the evacuated zone in the Fukushima accident. That area also is roughly the same as the Angra dos Reis evacuation area analyzed in the study of Silva et al. [101].

5.7 Datasets

The distribution of CHs ensures that there is at least a distance R between any two CHs. We adopt three different densities to evaluate the data collection strategies: a sparse, a dense and a *full* distribution. As the area does not vary, the densities are the element which varies the number of CHS. Sparse distributions have 1CH/10km^2, dense distributions have 1CH/5km^2, and *full* distributions have 1CH/1km^2. For each density, we used 200 different distributions of CHs and ran each one with every set of UAVs in order to generate the simulation data for our evaluations (two sets of distribution densities times 200 maps times 4 sets of $UAVs$ for each strategy).

6
Results and discussions

In this chapter, we discuss the results of the various experiments comparing the strategies described in section 4. The datasets for all experiments were presented in section 5.7. These experimental results are compared and analyzed from multiple perspectives. In section 6.1 we present the various tour sizes computed with each of the strategies for the same configuration of CHs on a map. Subsequently in section 6.2, we analyze the amount of collected data (TD^T) that each strategy can retrieve during the period T. This chapter includes analysis of variation in the number of UAVs and DADCA efficiency.

The delay in message retrieval (D_n^T) and whether it is impacted by increasing the number of UAVs are analyzed in section 6.3. The resources required in the tour planning of each strategy, in terms of hardware comparable to the hardware available in UAVs, is presented in Section 6.4. The section 6.5 discuss energy issues. The datasets and all raw data obtained are available as described on Annex one. Some of the raw data are also present in this chapter as tables in order to support discussions with more details.

6.1
Tour sizes analysis

As all UAVs fly at a constant speed during the period T, the length of the tour is an important factor in all subsequent experimental results. For this reason, we briefly present a tour size analysis below.

6.1.1
Experiment parameters

A tour was computed using each strategy for every map of the dataset at three densities of CHs (sparse, dense, and *full*). There were 3600 runs to compute the tours for each dataset for each strategy. Taking into consideration that a tour cloud be analyzed as a graph $G(V, E)$ and an edge E_n is a cost to move from the vertex V_n to vertex V_{n+1}, the presented values are the sum of all edges from $G(V, E)$.

In the graphs depicted in the following subsection, the vertical axis represents the tour sizes in kilometers, and the horizontal axis represents the map IDs. The chart title refers to the size of each CH's density set.

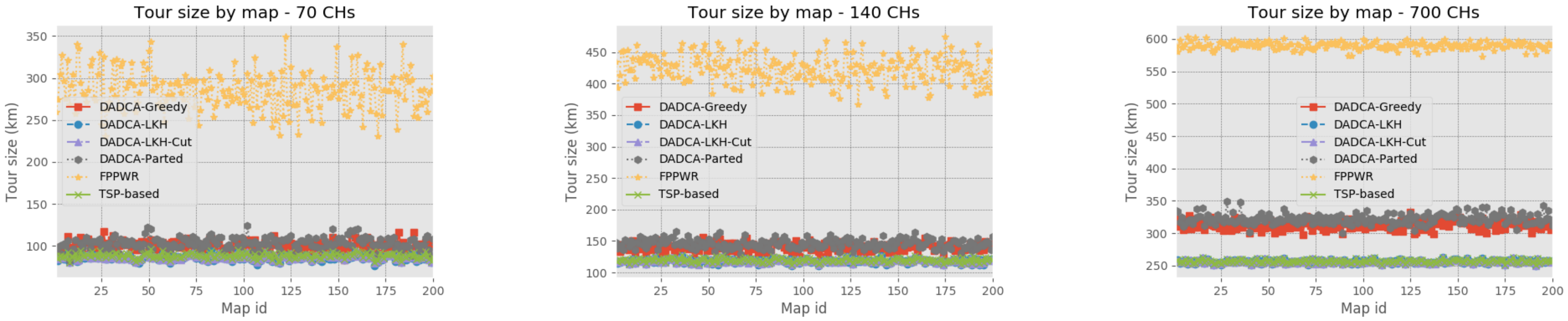
6.1.2 Analysis of tour sizes

The first set of results consists of three charts presented in Figure 6.1, one for each density of CHs, which illustrate all executions at that density. It is important to note that the FPPWR strategy produces tours that are significantly longer than those of the other strategies. In some maps, the FPPWR tours are twice as long as of those of the other strategies with sparse density. In the scenarios with the dense and *full* densities, the FPPWR tours extend up to three times the size of the other tours.

Three main groups of results emerge as the density and the number of CHs increase. These groups can be observed in Figure 6.1.(c).

The charts presented in Figure 6.2 consists of the same three charts but excludes the FPPWR strategy in order to illustrate the other strategies more clearly. Two groups emerge in Figure 6.2: (1) smaller tour sizes generated by the DADCA-LKH, DADCA-LKH-Cut, and TSP-based strategies and (2) longer tours generated by the DADCA-Greedy and DADCA-Parted strategies.

The results are inversely proportional to the computational effort required to create the tours. Strategies aiming route optimization, in fact, produce smaller tours, a result that becomes more clear in the scenarios with a *full* density of CH.

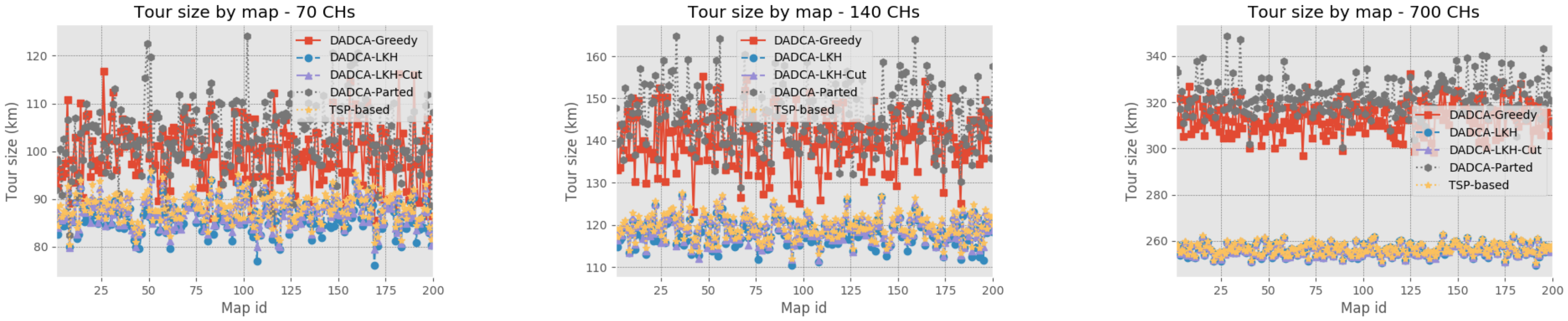


(a) **Sparse** tour lenghts

(b) **Dense** tour lenghts

(c) ***full*** tour lenghts

Figure 6.1: All strategies tour lengths series grouped by CH densities.



(a) **Sparse** tour lenghts

(b) **Dense** tour lenghts

(c) ***full*** tour lenghts

Figure 6.2: All strategies but FPPWR tour lengths series grouped by CH densities.

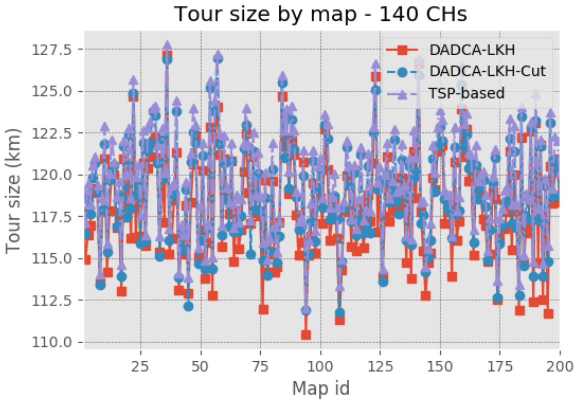


Figure 6.3: Optimization strategies tour lenghts.

Figure 6.1 shows all results for each simulation map and for easier comparison Figure 6.3 displays only the TSP-based, DADCA-LKH, and DADCA-LKH-Cut strategies with data from Figure 6.1.(b). Our analysis indicates that, even with a relatively small optimization load, the tours from strategies DADCA-LKH or DADCA-LKH-Cut were shorter than TSP-based ones for each simulation map. These differences concerning the length of the tours are more significant in the maps with smaller numbers of CHs, where the removal of an edge from $G(V, E)$ (as in DADCA-LKH and DADCA-LKH-Cut) is more significant. The sole fact that a strategy generates a smaller tour does not guarantee better results in the ensuing evaluations, because the efficiency of the strategies when using the generated tour will influence the results.

In sparse scenarios, the DADCA-Greedy and DADCA-Parted strategies presented tours about 18% longer than the strategy TSP-based. Meanwhile, the strategy FPPWR presented tours of about 200% longer than the strategy TSP-based.

6.2
Collected Data - TD^T

In this evaluation, we present TD^T formalized as equation (3) in section 3.3. The UAVs were sent to collect data before the end of the period T, and when period T ends, the TD^T is measured. Section 6.2.2 discusses the results from a normalized perspective, without the outlier values (i.e. a value that lies an abnormal distance from other values in the results). Section 6.2.3 analyzes the impact of varying the number of UAVs on the results by presenting the average of all found values. To conclude this section we sumarize these results in subsection 6.2.4.

6.2.1 Experiment parameters

The results in this section were obtained through simulations, using the data sets analyzed in section 6.1 and the three CHs densities (sparse, dense and *full*). The simulations were also run with different sets of UAVs (2, 4, 8, and 16), and the other variables were set as indicated in table 5.1. A total of 14,400 simulations were run.

6.2.2 Analysis of TD^T

For this specific subsection, the results account for a single set of UAVs (8 UAVs). All results, with the exception of outliers (0,2% of the number of results), are presented in boxplot charts, so 99.8% of the results are included in the charts. The results appear in Figure 6.4, grouped by strategy and density of CHs.

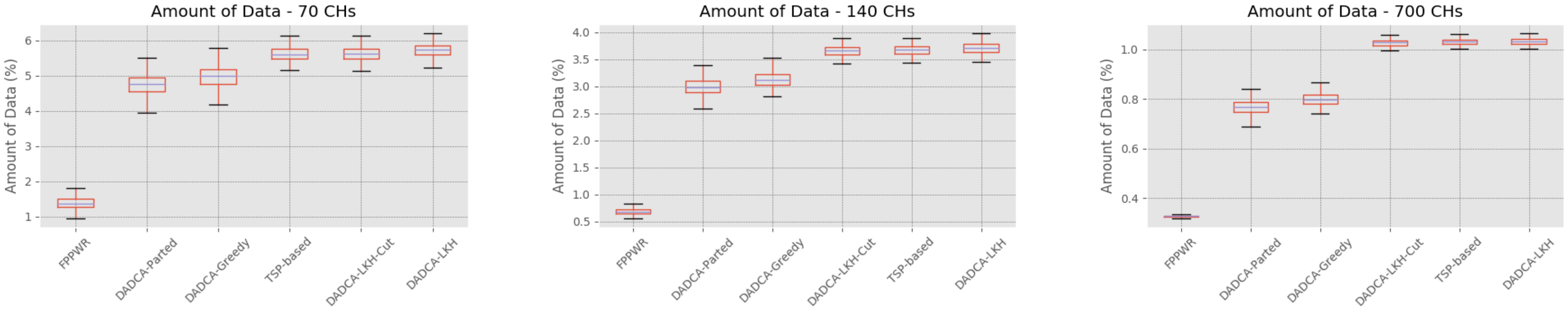
The vertical axis represents TD^T, while the horizontal axis indicates the various strategies. All charts are ordered by the series' medians, from left to right, meaning that the further right a strategy is placed in the charts, higher are the TD^T results.

Based on the charts displayed in Figure 6.4, it is possible to cluster the series into three groups. This group formation is more evident in the *full* scenario, as displayed in Figure 6.4.(c). The FPPWR strategy produces isolated results, with the worst positions in all density scenarios across the charts. The DADCA-Naive and DADCA-Parted strategies form the second group, with significantly better results than FPPWR. The DADCA-LKH, DADCA-LKH-Cut and TSP-based strategies, which aim to create shorter tours, make up the third group with the best results.

The graphics in Figure 6.5 focus only on the series with the three strategies with better results (DADCA-LKH, DADCA-LKH-Cut and TSP-based) from Figure 6.4. Figure 6.5, indicates thus that DADCA-LKH has the best results of the three strategies. The strategy DADCA-LKH have better results due to its shorter tours.

The DADCA-LKH-Cut strategy produces better results than the TSP-based strategy in the sparse scenario, whereas TSP-based produces the second-best results in the dense and *full* scenarios. These results are caused by the efficiency of the DADCA variations in the use of the generated tours, as seen in section 4.3.7. More specifically, the shadow effect (SE), described in section 4.3.7.1, is responsible for DADCA-LKH-Cut strategy having shorter tours and worse results than the TSP-based strategy.

As expected and described in section 4.3.8, the DADCA-LKH strategy obtained better results than the DADCA-LKH-Cut strategy due to the SE of each strategy.

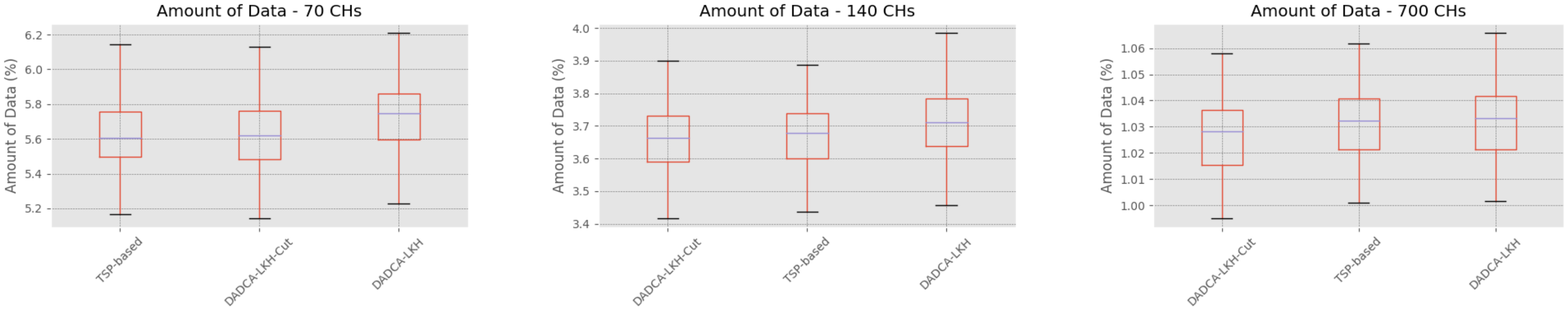


(a) TD^T on **sparse** maps

(b) TD^T on **dense** maps

(c) TD^T on ***full*** maps

Figure 6.4: TD^T of all strategies series divided by charts of map densities. The series is crescent ordered by its medians.



(a) TD^T on **sparse** maps

(b) TD^T on **dense** maps

(c) TD^T on ***full*** maps

Figure 6.5: TD^T of the better strategies series divided by charts of map densities. The series is crescent ordered by its medians.

The numbers corresponding to the chart series in Figure 6.4.b can be found in Table 6.1. This table has two parts; the first part presents the values relativized to TSP-based strategy results. The second presents absolute values that were obtained from experiments and used to plot Figure 6.4.b.

The Table 6.1 columns present the box plot chart as follows: **avg** presents the simple average of results; **std** presents the standard deviation; **min** shows the minimum values except outliers; **first quartile** presents the limit from the first 25% of the results between the **minimum** and the first quartile; **median** splits the results into two groups with the same number of results; **third quartile** presents the limit from the first 75% of the results between the **minimum** and third quartile; **max** shows the maximum values except outliers.

In Table 6.1 we use three colors of shading. Blue shows the values of the TSP-based strategy, which are used as benchmarks; green indicates values better than those of the TSP-based strategy and yellow shows marginal values. Cells without shading present values worse than the TSP-based strategy.

The DADCA-LKH strategy yields a result 1% better in terms of TD^T than the TSP-based strategy. Despite the median results, the bottom results show that DADCA-LKH performs 4% better regarding TD^T than TSP. The DADCA-LKH-Cut strategy varies between better and worse results with a 1% difference.

	TD^T - relative values						
	avg	std	min	first quartile	median	third quartile	max
DADCA-Greedy	85%	197%	57%	84%	85%	86%	91%
DADCA-LKH	101%	147%	104%	101%	101%	101%	101%
DADCA-LKH-Cut	99%	187%	99%	100%	100%	99%	101%
DADCA-Parted	81%	183%	54%	80%	81%	83%	86%
FPPWR	19%	61%	16%	18%	18%	19%	21%
TSP-based	100%	100%	100%	100%	100%	100%	100%

	TD^T - absolute values						
	avg	std	min	first quartile	median	third quartile	max
DADCA-Greedy	3.119847	0.189771	1.94895	3.027265	3.112435	3.231568	3.58178
DADCA-LKH	3.702740	0.141874	3.56974	3.637973	3.710835	3.783408	3.98560
DADCA-LKH-Cut	3.653436	0.179642	3.40569	3.589738	3.662130	3.718971	3.98675
DADCA-Parted	2.983887	0.176058	1.84338	2.890238	2.983050	3.098662	3.39494
FPPWR	0.682334	0.058674	0.56357	0.642643	0.677660	0.722955	0.83244
TSP-based	3.675034	0.096187	3.43780	3.599060	3.678455	3.737715	3.94646

Table 6.1: Table presenting the raw data and relative results from chart in Figure 6.4.b

6.2.3 Impact of the number of UAVs on TD^T

In this subsection we present the same experiments from the previous subsection and include all different numbers of UAVs and outlier results. For all charts in Figure 6.6, there is a series with the results of a running strategy. Each point in a series is the average from all executions in that density of CHs for all maps from the dataset.

For every series in the charts depicted in Figure 6.6, one can notice that an increase in the number of UAVs leads to a direct increase in TD^T for each experiment. This linear dependency is due to the fact that across all strategies, the UAVs collect data by flying over CHs at a constant speed in a computed tour, and hence, the more aircraft fly, the more data is collected.

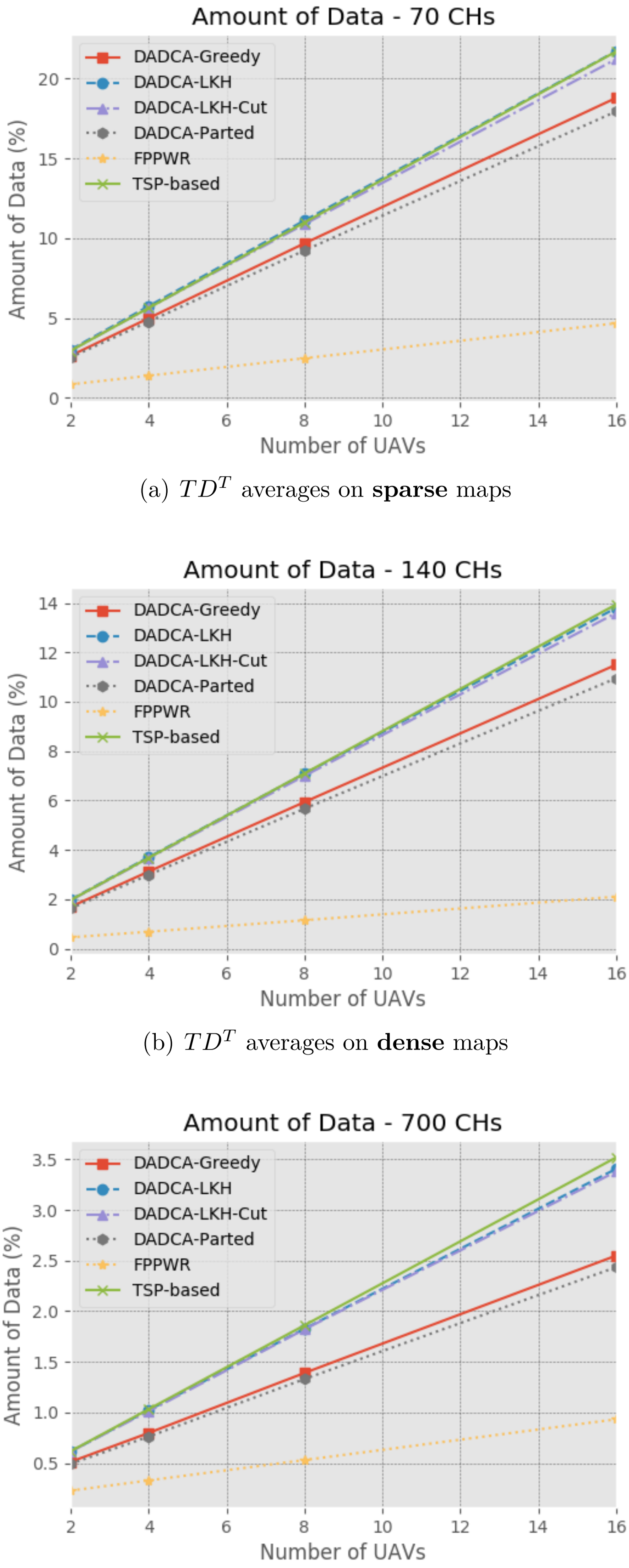
The values of the chart series in Figure 6.6.c are shown in Table 6.2.3. The TD^T values are presented in two columns. The first contains the values used in the chart in Figure 6.6.c and the second shows the results in relative terms, as in the previous subsection with green, yellow and blue shading.

The data in Figure 6.6.c and Table 6.2.3 suggests that the TSP-based strategy produces better results on average than the others, indicating that in certain situations TSP performs better than the DADCA variations. This happens because DADCA can be less efficient than the TSP-based strategy; such cases are detailed in section 4.3.7.

6.2.4 Experiment summary

In general scenarios, the DADCA variation results in almost the same or even more TD^T than the other strategies. Indeed, the individual analysis of each case in which DADCA was less efficient than TSP indicated that this was due to the shadow effect (SE), as described in 4.3.7.1. Increasing the number of UAVs in use may improve TD^T results for any strategy, but may particularly enhance DADCA's SE.

In all tests, there were no cases in which geographic issues were the cause of DADCA's inferior results, as shown in section 4.3.7.2. It is expected that geographic issues may be an issue for DADCA in very sparse distributions. This is because the distance between CHs in sparse distributions is greater than in denser distributions in areas of the same size. In these cases, it is easier for a single distance between two CHs to be much shorter than all others. Therefore, if the DADCA-LKH or DADCA-LKH-Cut strategies remove this edge from the LKH algorithm result, both strategies would have tours less significantly



(a) TD^T averages on **sparse** maps

(b) TD^T averages on **dense** maps

(c) TD^T averages on ***full*** maps

Figure 6.6: TD^T averages with all results.

Strategy	UAVs	CHs	TD^T	Relative TD^T
DADCA-Greedy	2	700	0.512763	83%
DADCA-Greedy	4	700	0.797865	77%
DADCA-Greedy	8	700	1.390456	75%
DADCA-Greedy	16	700	2.549781	72%
DADCA-LKH	2	700	0.614581	100%
DADCA-LKH	4	700	1.022989	99%
DADCA-LKH	8	700	1.826621	98%
DADCA-LKH	16	700	3.406617	97%
DADCA-LKH-Cut	2	700	0.614260	99%
DADCA-LKH-Cut	4	700	1.011870	98%
DADCA-LKH-Cut	8	700	1.821483	98%
DADCA-LKH-Cut	16	700	3.379867	98%
DADCA-Parted	2	700	0.493140	80%
DADCA-Parted	4	700	0.761197	74%
DADCA-Parted	8	700	1.330494	71%
DADCA-Parted	16	700	2.435450	69%
FPPWR	2	700	0.228466	37%
FPPWR	4	700	0.329242	32%
FPPWR	8	700	0.530531	29%
FPPWR	16	700	0.933046	27%
TSP-based	2	700	0.617414	100%
TSP-based	4	700	1.032240	100%
TSP-based	8	700	1.861430	100%
TSP-based	16	700	3.519386	100%

Table 6.2: Table presenting the raw data and relative results from chart in Figure 6.6.c

shorter than the TSP-based strategy, and then the SE would more easily make these two strategies yield worse results than the TSP-based strategy.

6.3 Messages delay - D_n^T

In this section we explore the D_n^T results, in other words, the mean delay of all message collected from each CH, as formalized in section 3.3 (5). The delays analyzed are a sub product of the experiment that was presented in the previous section (6.2). After presenting these results, we analyze the effects of varying the number the UAVs on each of the strategies.

6.3.1 Analysis of D_n^T

Figure 6.7 presents all strategies and the D_n^T of the retrieved messages that reach the GS. Each point in the charts represents the average of 200 maps from the dataset. The horizontal axis represents the number of UAVs.

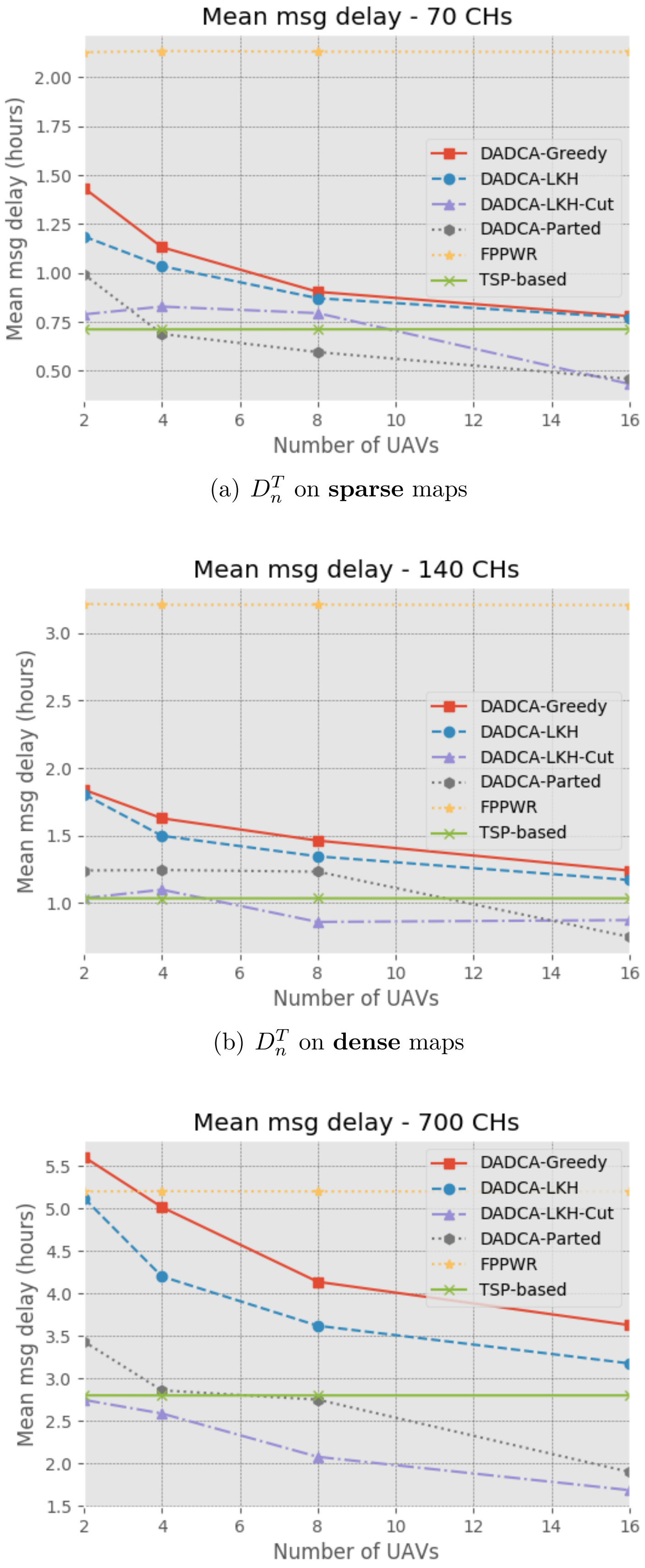
The vertical axis represents D_n^T and smaller values on the vertical axis indicate better results.

Due to the tour length being larger in denser scenarios likewise as the density (sparse, dense, and *full*) increases, the delays increase. The FPPWR strategy produces the worst results in all scenarios in the analysis, because its tours are significantly longer than those of the other strategies. While the results do not indicate that the TSP-based strategy is the best, TSP-based does produce good results in all scenarios.

Figures 6.8 and 6.9 contain charts of delays of 16 UAVs collecting data from 700 CHs. The three strategies with the best results were selected for Figure 6.9. The best results for D_n^T are obtained with DADCA-LKH-Cut and DADCA-Parted. Without loss of generality we will assume that the GS is roughly positioned in the middle of the tours of DADCA-LKH-Cut and DADCA-Parted, so that the distance required to deliver the data drops significantly.

Table 6.3.1 presents the data from the chart in Figure 6.8. This chart clearly shows the advantage of DADCA-LKH-Cut and DADCA-Parted over the others. DADCA-LKH-Cut had 32% shorter delays, and DADCA-Parted had 20% shorter delays than TSP-based when comparing the medians, as well as the maximum values.

As explained in section 4.4, the strategies that follow a Hamiltonian cycle have constant delays, while variations of DADCA take advantage of the increase in the number of UAVs, thus decreasing the message delays. More specifically in the cases of DADCA-LKH-Cut and DADCA-Parted, the positioning of the GS roughly in the middle of the tour increases the frequency with which the UAVs deliver their collected messages.



(a) D_n^T on **sparse** maps

(b) D_n^T on **dense** maps

(c) D_n^T on ***full*** maps

Figure 6.7: D_n^T - average delays of each strategy for each density and distinc number of UAVs.

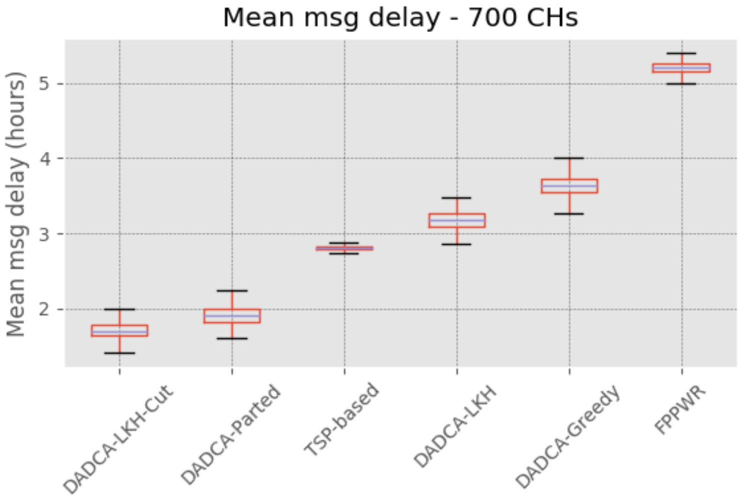


Figure 6.8: D_n^T - delays of each strategy in *full* density and 16 UAVs as in Figure 6.7.(c)

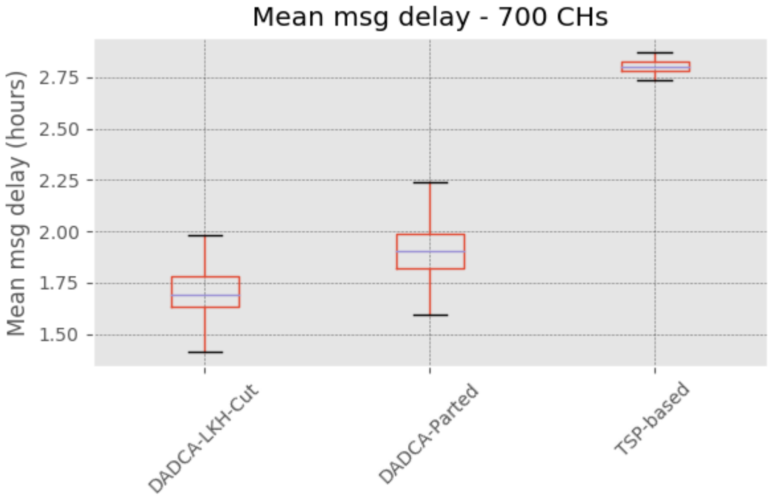


Figure 6.9: D_n^T - delays of the three best strategies in *full* density and 16 UAVs as in Figure 6.8.

	D_n^T - relative values						
	mean	std	min	25%	50%	75%	max
DADCA-Greedy	128%	1289%	71%	127%	130%	132%	208%
DADCA-LKH	112%	876%	60%	111%	113%	116%	130%
DADCA-LKH-Cut	68%	947%	39%	59%	60%	63%	71%
DADCA-Parted	80%	1186%	43%	65%	68%	70%	86%
FPPWR	186%	503%	181%	185%	186%	186%	225%
TSP-based	100%	100%	100%	100%	100%	100%	100%

	D_n^T - absolute values						
	mean	std	min	25%	50%	75%	max
DADCA-Greedy	3.597427	0.368877	1.953889	3.540278	3.630694	3.732569	5.976667
DADCA-LKH	3.146258	0.250700	1.632222	3.092917	3.180139	3.269653	3.746389
DADCA-LKH-Cut	1.893868	0.271003	1.072778	1.631667	1.688472	1.782014	2.050277
DADCA-Parted	2.243707	0,339635	1.188056	1.819167	1.902917	1.986736	2,458611
FPPWR	5.212640	0.143923	4.952500	5.149583	5.203194	5.258542	6.476667
TSP-based	2.804683	0.028628	2.736667	2.783611	2.803611	2.825069	2.873056

Table 6.3: Table showing the raw data and relative results from Figure 6.8.

6.3.2 Experiment summary

With a small number of UAVs, the DADCA variations present similar or worse results than the TSP-based strategy, and this inefficiency happens due to the limitations described above in section 4.3.7 that explains the limitations of DADCA.

However, when more UAVs are used, there is a significant improvement in the performance of the DADCA variations. In all evaluated scenarios, the DADCA-LKH-Cut and DADCA-Parted strategies yielded better results than the TSP-based strategy due to the more frequent connections to GS.

6.4 Tour planning time

This section analyzes the computational effort required for the various strategies to create their tours. As previously mentioned, the complexity of DADCA-Naive and DADCA-Parted is $\Theta(n^2)$, the complexity of DADCA-LKH and DADCA-LKH-Cut is $\mathcal{O}(n^{2.2})$, the complexity of FPPWR is $\mathcal{O}(n \log n)$, and the TSP-based strategy is a NP hard problem.

An analysis of the algorithm's complexity provides helpful estimates regarding the theoretical processing time requirements of each strategy. However, the algorithm complexity does not provide a complete comparison of the tour planning timing of each algorithm. For this reason, we present each strategy's planning time in a real UAV processing system. The UAV processing system contains only the necessary components to control the flight, namely flight controllers.

UAVs generally only rely on flight controllers with processors like the ARM Cortex-M (with 32bits and clocks of 200MHz). Typically, the UAV's flight controllers rely on processing systems that are as light and spend as little energy as possible. Moreover, due to security issues, is highly not recommended to run arbitrary applications on the flight controllers. Similar technical specifications are found in the commonly used PixHawk flight controllers family[1].

In order to run application level programs in UAVs, it is usually necessary to install a dedicated processing unit for the application. The most-used configuration is to adopt single-board computers, which are naturally in compliance with weight and energy consumption requirements. Indeed, the frameworks for platforms such as FlytOS [52], ErleBrain [53] and Dronecode [102] are based on this architecture.

[1] https://pixhawk.org/

The cited projects [52][53][102] rely on single-board main computers, such as the RaspBerry Pi, BeagleBone, and Intel Edison. For evaluating the proposed strategies, we selected the RaspBerry Pi 3 as the hardware due to its high availability to be acquired in the market.

6.4.1 Experiment parameters

This experiment demonstrates the time spent for each strategy to compute its tour using the same hardware. The Raspberry Pi 3 relies on a quad-core processor, the ARM Cortex A53 (ARMv8). The ARM cores run at 1.2GHz, with a GPU core that runs at 400MHz. The dataset and densities used are the same as in previous experiments.

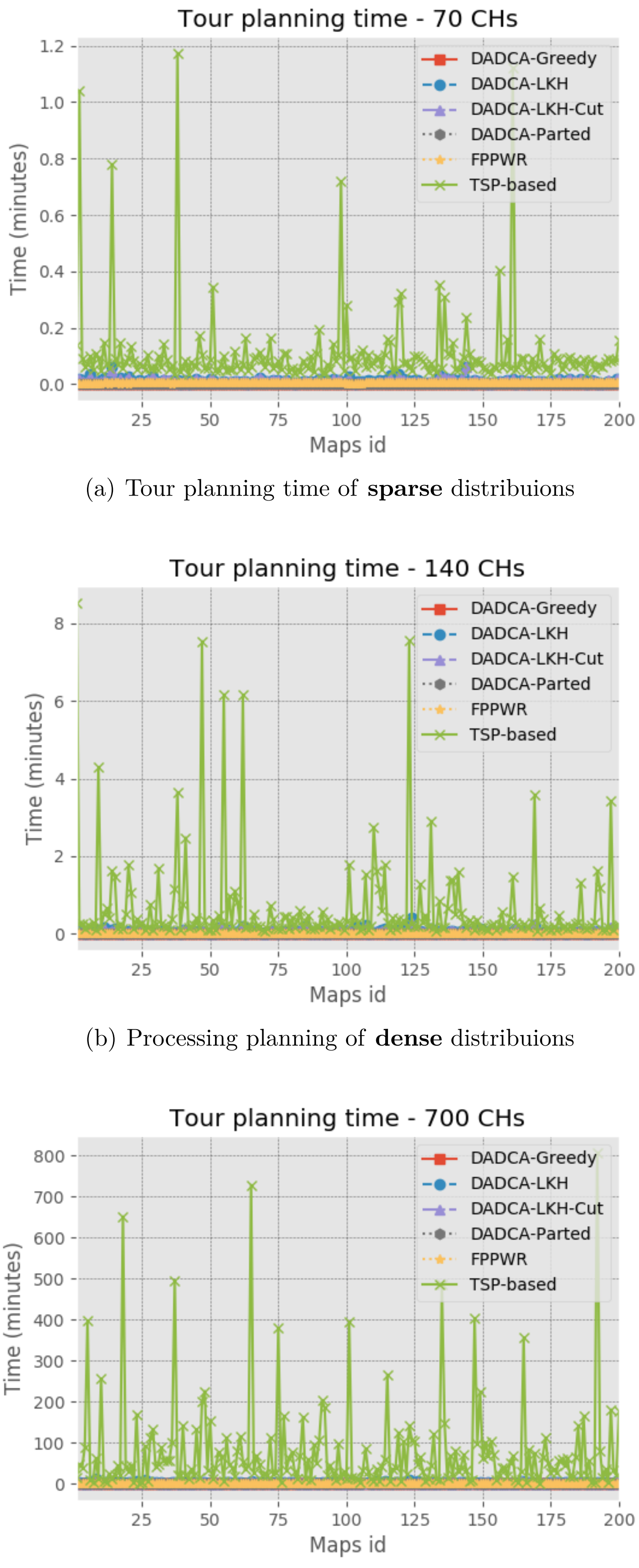
6.4.2 Analysis tour planning time

Figure 6.10 presents the results of each strategy's tour processing time. Each chart in Figure 6.10 refers to a density level. The horizontal axis represents the maps, and the vertical axis represents the processing time. Smaller values along the vertical axis are better.

The first noticed result is that the TSP-based strategy demands significantly more time than any other strategy. In fact, the TSP-based results make it difficult to analyze the other strategies in the same chart because TSP-based results mislead the scale of results. Accordingly, in Figure 6.11 the same results are grouped together and the average is presented. Accordingly, in Figure 6.11 the same results of Figure 6.10 are grouped together and the average is presented. It is more clear that TSP-based results have a complete other scale.

In order to provide a clearer analysis, we present only the *full* series in the charts in Figures 6.12, 6.13, and 6.14, which do not contain outliers and are presented as boxplots. In Figure 6.12 all results are presented. The outliers were removed because some results of the TSP-based strategy differ largely from the average results, as shown in Figure 6.10. In all charts the vertical axis shows the processing time, where the lower values along the vertical axis are faster.

In Figure 6.13 the three strategies with the best results in the previous two experiments produced the worst results, in terms of computing the best tour while consuming computational resources. Even without outliers, at least 25% of the TSP-based strategy's tours still require more than 29 minutes of computing time. In addition, at least 75% of the TSP-based strategy's tours still require more computing time than those of the other strategies. While



(a) Tour planning time of **sparse** distribuions

(b) Processing planning of **dense** distribuions

(c) Processing planning of ***full*** distribuions

Figure 6.10: Tour planning charts for each map.

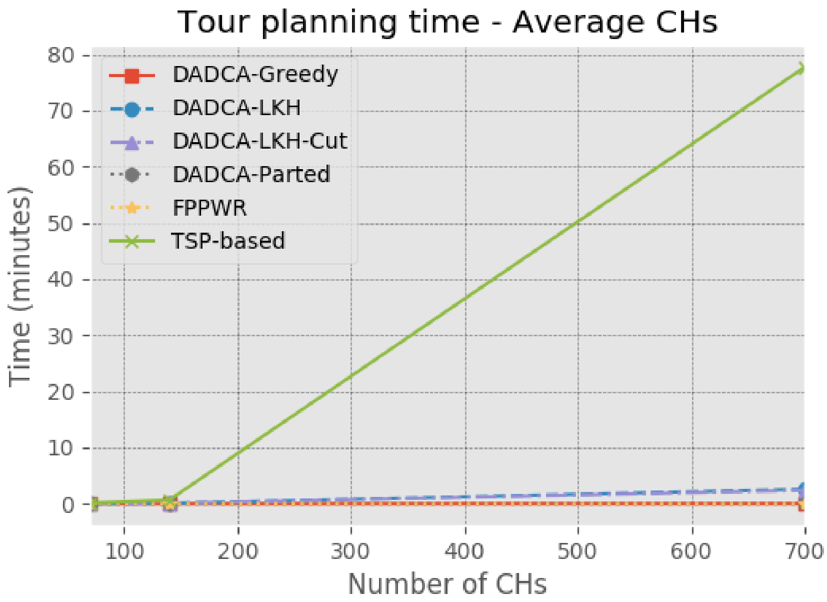


Figure 6.11: Average processing time in diverse distributions.

DADCA-LKH and DADCA-LKH-Cut produce better results than TSP-based, the results of both are still far worse than the others.

In Figure 6.14 we present the three best results in the FPPWR, DADCA-Naive, and DADCA-Parted strategies. DADCA-Parted produced the best results in this experiment, computing its tours in less time than all the others.

DADCA-Parted obtains its results by splitting the original set of CHs into two subsets to compute its tour. For this reason, its results are significantly better than those of DADCA-Naive.

Despite the FPPWR results, the overall results could reasonably have been expected because they are directly proportional to the efforts of each strategy to compute its tour. Optimization heuristics are expected to be harder to solve, followed by polynomial heuristics dealing with optimization problems. Finally, purely polynomial strategies tend to be less costly. It is reasonable to expect that at some number of CHs, the FPPWR strategy would produce the best results, but this did not occur in preliminary tests up to 1000 CHs.

Table 6.4.2 presents in detail the results from the chart in Figure 6.12. Table 6.4.2 shows the impressive difference between the TSP-based strategy and the other results, and how DADCA-Parted yielded better results than the others.

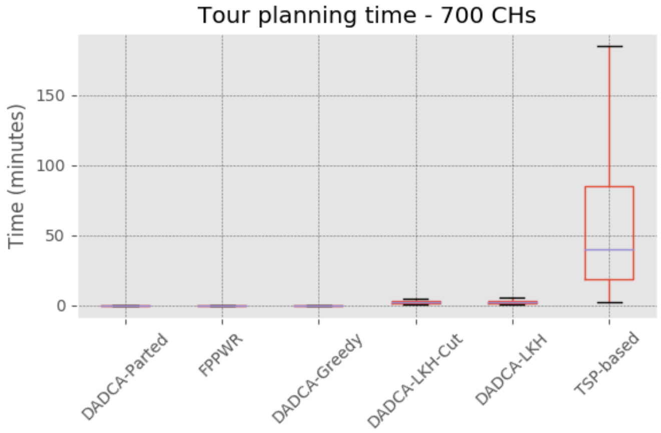


Figure 6.12: Processing planning of ***full*** distribuions in a boxplot view.

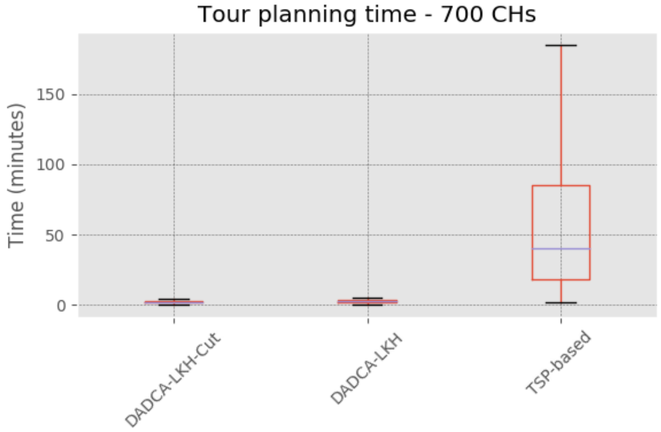


Figure 6.13: Presenting selected worst results. Processing planning of ***full*** distribuions in a boxplot view.

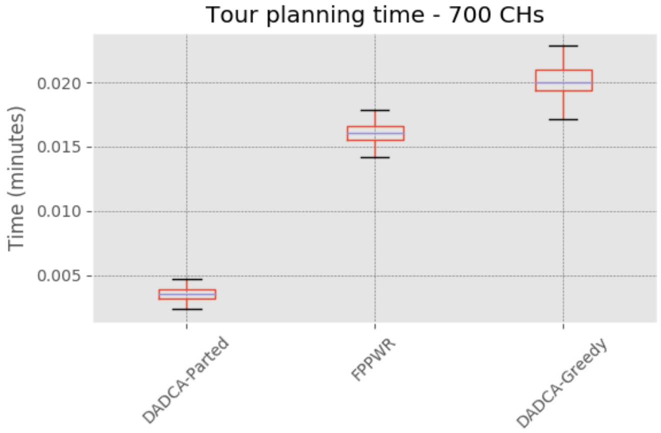


Figure 6.14: Presenting selected best results. Processing planning of ***full*** distribuions in a boxplot view.

	Tour planning time - relative values						
	avg	std	min	25%	50%	75%	max
DADCA-Greedy	0.026%	0.002%	0.586%	0.105%	0.050%	0.025%	0.03%
DADCA-LKH	3%	1%	23%	9%	6%	4%	2%
DADCA-LKH-Cut	3%	1%	24%	8%	5%	3%	1%
DADCA-Parted	0.005%	0.001%	0.099%	0.017%	0.009%	0.005%	0.01%
FPPWR	0.020%	0.001%	0.485%	0.085%	0.040%	0.019%	0.02%
TSP-based	100%	100%	100%	100%	100%	100%	100%

	Tour planning time - absolute values						
	avg	std	min	25%	50%	75%	max
DADCA-Greedy	0.019832	0.001977	0.012567	0.019350	0.020050	0.020950	0.024500
DADCA-LKH	2.585966	1.591076	0.489717	1.653225	2.211267	3.224783	12.879750
DADCA-LKH-Cut	2.389736	1.335887	0.525000	1.495325	2.111917	2.775875	9.831850
DADCA-Parted	0.003603	0.000712	0.002133	0.003175	0.003500	0.003858	0.007550
FPPWR	0.015808	0.001578	0.010400	0.015567	0.016050	0.016567	0.019533
TSP-based	77.719820	116.984831	2.145883	18.402967	39.936850	85.177783	806.893717

Table 6.4: Table presenting the relative values of timing consuming on planning tours.

6.4.3 Experiment limitations

In this subsection we discuss three experimental limitations relating to the following areas: memory utilization, and possible process parallelization and other issues that might have had influence on the processing time.

Memory In the preliminary tests involving up to 1000 CHs, none of the strategies reached 128Mb of memory utilization. As the evaluated hardware has 1Gb of RAM, this work does not delve further into memory evaluation.

Parallel Processing The processing time evaluation ran without parallel optimizations or GPU usage on Raspberry Pi 3. All evaluations ran as a single execution thread.

Processing time The tour processing time presented in this subsection was not considered in the previous section, meaning that the amount of collected data TD^T and delay D_n^T were evaluated as if the processing time of each strategy was zero. It is reasonable to expect that TSP-based could produce significantly different results if the UAVs waited to begin their data collection until after the TSP-based strategy computed the tour.

6.5 Energy constraints

In this section we introduce two energy issues relating to UAVs. Section 6.5.1 explores the UAVs' inversion of flight direction (U turn) after a valid rendezvous. The tour planning aboard UAVs may consume significant energy and is explored in section 6.5.2.

6.5.1 DADCA rendezvous energy impacts

As mentioned earlier, the system model does not consider energy consumption in the alteration of the UAVs' trajectories. The angle of a UAV's turns on a tour does not affect that UAV's energy consumption model in any of the strategies.

In all DADCA variations, the standard behavior of all UAVs is to change direction by a 180°angle; in other words, to perform a U-turn. It is reasonable to expect that this behavior will consume energy. That energy consumption could lead to a need for more frequent battery recharges, and more frequent visits to the GS could increase the SE.

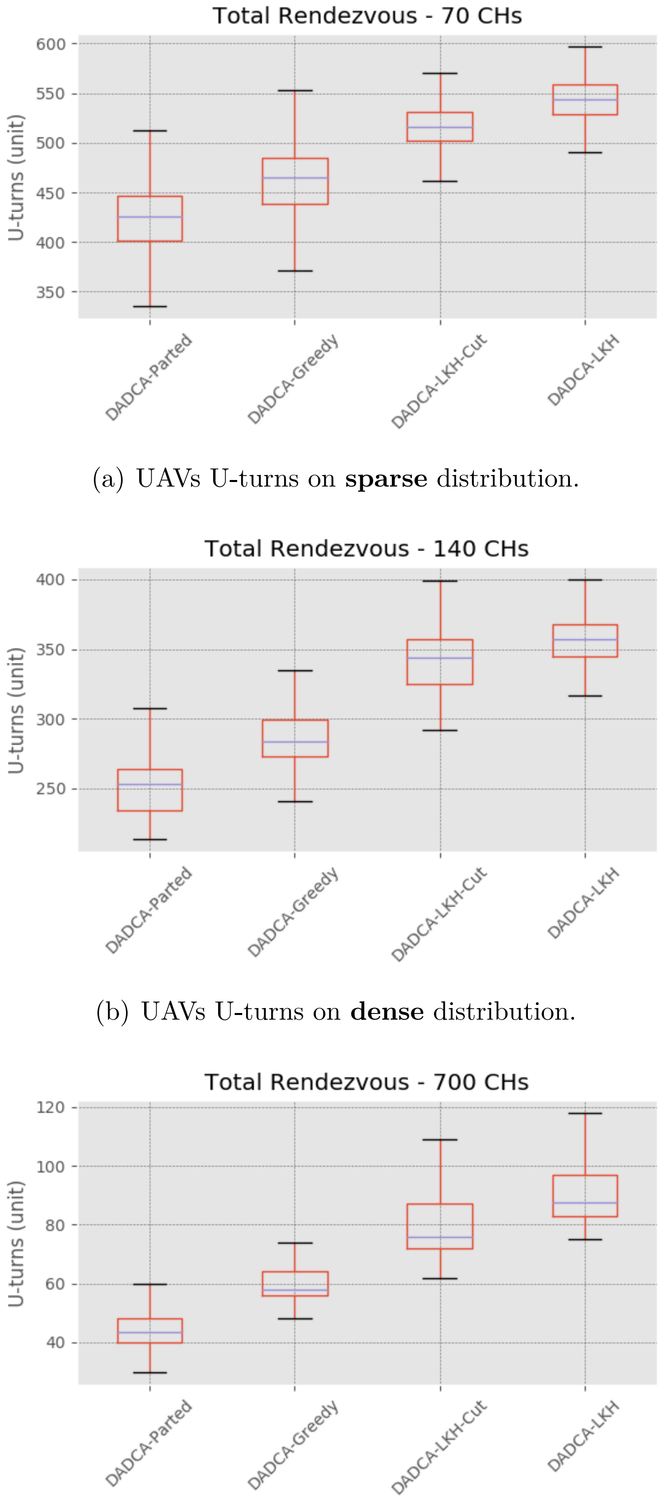
Figure 6.15 consists of the three charts that contain the number of U-turns performed by the UAVs in the experiments. The number of UAVs considered in this Figure is 16 UAVs. It can be noted that DADCA-Naive and DADCA-Parted tours include fewer U-turns than tours of the other strategies because both strategies produce longer tours. The tours of DADCA-LKH and DADCA-LKH-Cut incorporate more U-turns than those of the others, because the same number of UAVs are collecting the data on these shorter tours. Among those two, DADCA-LKH-Cut executions have fewer U-turns than DADCA-LKH executions, because DADCA-LKH-Cut produces a larger SE; when flying in pairs, both UAVs ignore other UAVs during a rendezvous. As a result, DADCA-LKH-Cut tours have fewer rendezvous than those of DADCA-LKH.

6.5.2 Tour planning energy consumption

As proposed, the tour shall be computed by each UAV. In section 6.4 the processing time of each strategy was presented, with some results taking hours to be processed. In this experiment we measured the energy consumed by the hardware discussed in section 6.4 to provide an estimate of the effects that tour planning could have on the energy consumption of the UAVs. In order to do so, we logged the energy consumption of the Raspberry Pi 3 for one hour under full-time processing of the tour.

The measure was taken with an inline ammeter during the hour. The ammeter used was the Keweisi model KWS-V20, with an accuracy of 0.4% in current measures. In idle processing, the Rapberry Pi spent 75mAh with 5,23 volts per hour. In tour processing tasks, the measure experiment resulted in 362mAh with 5,13 volts per hour. This result was divided by 3600 seconds to obtain 0.1mAh with 5,13 volts per second. The results of the energy consumption were used in conjunction with the results from section 6.4 to present the energy necessary for each strategy to compute a tour, as demonstrated in Figure 6.16.

It is important to note that mAh is not a measure of energy; rather, it is a measure of current-flow integrated across time, and to derive the energy value requires the voltage as well. We present the charts in joules, with smaller values along the vertical axis being better. The charts in Figure 6.16 are the straight application of the tour planning time presented in section 6.4. Figure 6.16 presents the TSP-based strategy as the worst strategy spending disproportionate energy related to the others tested strategies. DADCA-Parted presented the best results followed by DADCA-Greedy.

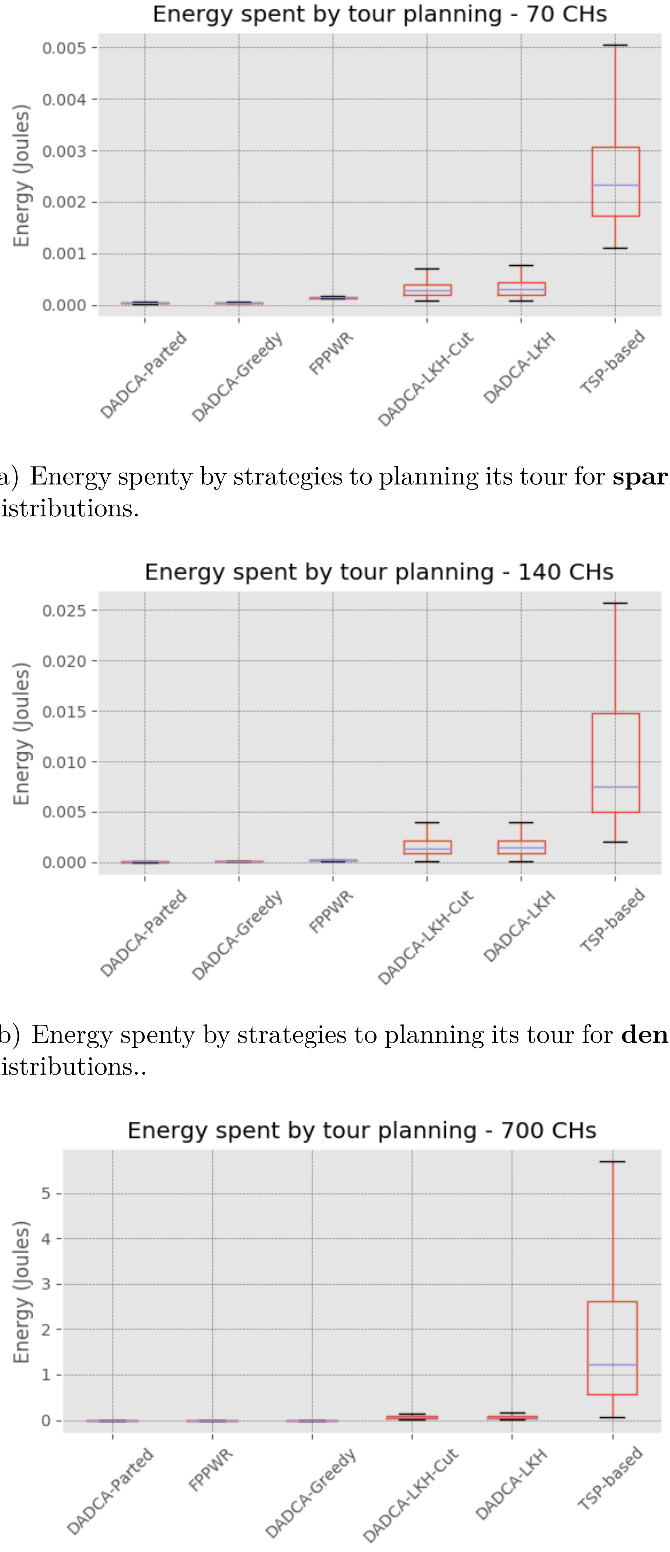


(a) UAVs U-turns on **sparse** distribution.

(b) UAVs U-turns on **dense** distribution.

(c) UAVs U-turns on ***full*** distribution.

Figure 6.15: Total number of UAVs U-turns during the period of T.



(a) Energy spenty by strategies to planning its tour for **sparse** distributions.

(b) Energy spenty by strategies to planning its tour for **dense** distributions..

(c) Energy spenty by strategies to planning its tour for ***full*** distributions.

Figure 6.16: Amount of energy spent by each strategy for each distribution.

7
Conclusions

This thesis proposes and validates a distributed approach for collecting data from WSNs with the use of several UAVs. Unlike other works that presuppose long-range communication and optimization of UAV routes, this work proposes the use of radios with short-range communication and the use of routes calculated on the basis of polynomial algorithms. In addition to this distributed approach, due to its oscillatory behavior DADCA variations can re-balance the entire group of UAVs when there is a UAV failure or a UAV addition, and the number of UAVs is scalable.

The experimental results (see Chapter 6) indicate that DADCA is a natural fit for missions where robustness is valued more highly than efficiency. However, for missions that have strict efficiency requirements, strategies relying on TSP-based approaches are well-suited in terms of the amount of collected data. Regarding delays in data collection, DADCA variations perform better than the other strategies evaluated. In all scenarios, the DADCA variations used less time to compute their tours than all other strategies. However, it is reasonable to expect that with a high number of CHs, the FPPWR strategy would yield better results than the other strategies because of the complexity of the FPPWR tour planning algorithm.

7.1
Limitations

Despite the benefits of the proposed approach, there are points on which the studies could delve deeper into some details. First, the simulations were carried out in the algorithm simulator, which does not focus on the kinematics or the consumption of energy by the aircraft. Thus, the kinematics could be analyzed more deeply if this was the main topic of study. The low level of communication failures was restricted to those made feasible by Sinalgo.

Furthermore, this work presupposes knowledge of the CH positions. Although this point is not explored, there are works directly focused on the best choice of CHs and their location. In addition, this work does not explore possible Byzantine faults of UAVs or cybersecurity issues.

A further limitation of this study is that it does not detail the treatment of possible losses of information collected by UAVs that may stop working before retransmitting. This aspect was not studied because its importance is directly linked to the final application of the data and not to our approach itself.

7.2 Revisiting the proposal and contributions

In order to address the aforementioned problems in section 1.2, this work has proposed DADCA variations. Figure 7.1 represents the details of the search and its intra-relationship. Problems I and RQ I come with constraints in communications, so the UAVs must load the collected data to their destination or another UAV. The FPPWR and TSP-based strategies carry the collected data up to the GS regardless of the number of UAVs. However, as presented in section 6.3, the DADCA strategies make better use of the increase in the number of UAVs, presenting better results in terms of D_n^T as the number of UAVs increases.

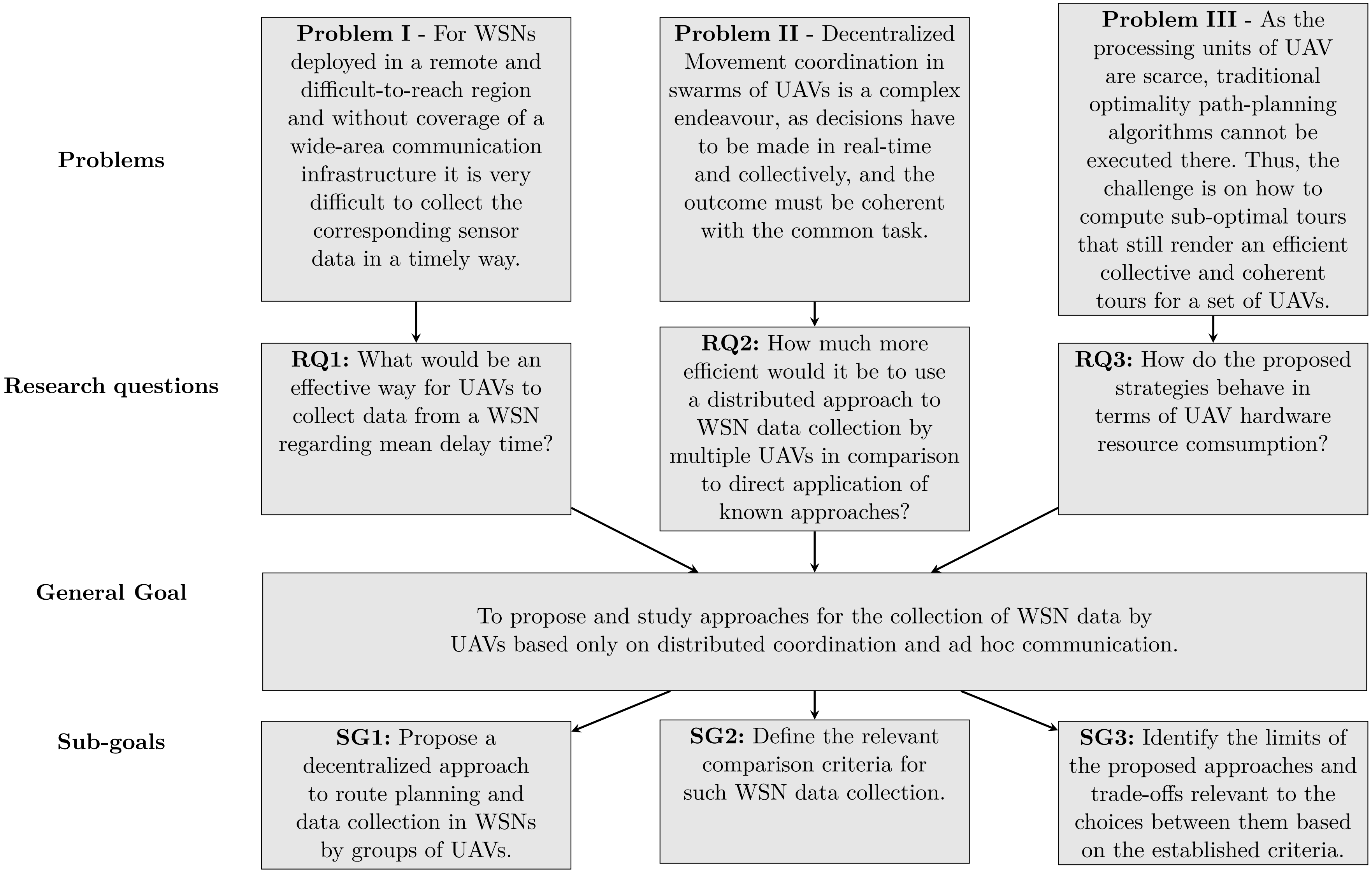
Problems II and RQ II are treated as the control of UAVs in the DADCA strategy, and are approached without using any global information that may be available to control the UAVs. For every eventual UAV failure, a reinforcement is automatically balanced. The number of rendezvous and message exchanges is linear and always oriented to two consecutive UAVs, which makes the approach scalable in terms of the number of UAVs.

Problems III and RQ III are analyzed in section 6.4, as all evaluated approaches have their tour planning phase tested with hardware similar to those used in UAVs. The results show that the DADCA variations are reasonably well-suited for UAVs meanwhile the TSP-based strategy is difficult to implement.

As the contribution of this thesis we proposed and compared different variations of DADCA, an approach for coordinating groups of UAVs for data collection in WSNs on the ground. All the variants of the DADCA allow to determine its tours in polynomial time (unlike usual TSP based approaches) and to be able to easily adjust the entrance and exit of UAVs of the swarm. In addition to the points presented, this thesis also directly produced the following works:

- B. Olivieri and M. Endler, “An Algorithm for Aerial Data Collection from Wireless Sensors Networks by Groups of UAVs,” The 2017 IEEE/RSJ International Conference on Intelligent Bibliography 109 Robots and Systems (**IROS** 2017), p. 6, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8202262/

– B. Olivieri and M. Endler, “DADCA: An Efficient Distributed Algorithm for Aerial DataCollection from Wireless Sensors Networks by UAVs,” The 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (**MSWiM**’17), pp. 129–136, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3127553



Problems

Problem I - For WSNs deployed in a remote and difficult-to-reach region and without coverage of a wide-area communication infrastructure it is very difficult to collect the corresponding sensor data in a timely way.

Problem II - Decentralized Movement coordination in swarms of UAVs is a complex endeavour, as decisions have to be made in real-time and collectively, and the outcome must be coherent with the common task.

Problem III - As the processing units of UAV are scarce, traditional optimality path-planning algorithms cannot be executed there. Thus, the challenge is on how to compute sub-optimal tours that still render an efficient collective and coherent tours for a set of UAVs.

Research questions

RQ1: What would be an effective way for UAVs to collect data from a WSN regarding mean delay time?

RQ2: How much more efficient would it be to use a distributed approach to WSN data collection by multiple UAVs in comparison to direct application of known approaches?

RQ3: How do the proposed strategies behave in terms of UAV hardware resource comsumption?

General Goal

To propose and study approaches for the collection of WSN data by UAVs based only on distributed coordination and ad hoc communication.

Sub-goals

SG1: Propose a decentralized approach to route planning and data collection in WSNs by groups of UAVs.

SG2: Define the relevant comparison criteria for such WSN data collection.

SG3: Identify the limits of the proposed approaches and trade-offs relevant to the choices between them based on the established criteria.

Figure 7.1: Research review

7.3 Future work

In addition to the limitations listed in the section 7.1, some points are also worthy of further study:

Rendezvous kinematics: It is assumed that during a valid encounter of two UAVs, the time during which both will be within range of each other will be great enough that the data collected from one UAV can be transmitted to the other UAV, which will load it. However, this may not happen due to an excess of data in a UAV, the throughput of the radios or even the speed of the UAVs. This line of inquiry could look at different flight patterns (e.g. UAVs either do or do not remain stationary until the end of the transfer).

If a UAV remains stationary until the transfer the end of data transmission, a factor that could also be investigated is whether or not, in a system that is not in equilibrium, a third UAV could pass through the UAVs and ignore them and be ignored. This could happen because two UAVs during a rendezvous ignore any other UAV. This third UAV would fatally affect the entire system balance and may cause more SE, as described in section 4.3.7.1.

Energy consumption: The energy spent on U-turns, as described in section 6.5.1, can be a significant trade-off when using DADCA. We intend to analyze how quadcopters and fixed-wing UAVs behave in such situations with their different models of energy use.

Evaluations: Some items presented in the table 5.1 can be changed to be better correlated in other specific scenarios. For example, MBTF and FT failure rates are set conservatively high and can be changed, which may benefit DADCA. On the other hand, the flight time could be reduced to recreational UAV levels, which would worsen the DADCA results in certain situations.

Cluster heads: In the proposed model, the CHs do not alternate during the experiments. However, the analysis of embedded processing time for tour planning suggests that UAVs running DADCA would be capable of **re**-calculating a completely new itinerary (its Original Path) whenever necessary. Further testing in this area could enrich the research.

In addition, the number of CHs in the *full* scenario is very high and, depending on the sensors in the final application, may include all the sensors of the application and not only CHs. In this specific case, the energy consumption in the WSN itself could be measured by collecting data without clustering instead of with clustering.

Bibliography

[1] S. M. A. Akber, I. A. Khan, S. S. Muhammad, S. M. Mohsin, I. A. Khan, S. Shamshirband, and A. T. Chronopoulos, "Data Volume Based Data Gathering in WSNs using Mobile Data Collector," in *Proceedings of the 22nd International Database Engineering & Applications Symposium on - IDEAS 2018*. New York, New York, USA: ACM Press, 2018, pp. 199–207. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3216122.3216166

[2] S. Liu, Z. Wei, Z. Guo, X. Yuan, and Z. Feng, "Performance Analysis of UAVs Assisted Data Collection in Wireless Sensor Network," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, jun 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8417673/

[3] C. H. Hailong Huang, Andrey V. Savkin, Ming Ding, "Mobile robots in wireless sensor networks: A survey on tasks," *Computer Networks*, vol. 148, pp. 1–19, jan 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912861830255X

[4] J. Grigulo and L. B. Becker, "Experimenting Sensor Nodes Localization in WSN with UAV Acting as Mobile Agent," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, sep 2018, pp. 808–815. [Online]. Available: https://ieeexplore.ieee.org/document/8502536/

[5] M. Garraffa, M. Bekhti, L. Letocart, N. Achir, and K. Boussetta, "Drones path planning for WSN data gathering: A column generation heuristic approach," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, apr 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8377391/

[6] M. Sayeed, R. Kumar, M. A. Sayeed, and R. Kumar, "An Efficient Mobility Model for Improving Transmissions in Multi-UAVs Enabled WSNs," *Drones*, vol. 2, no. 3, p. 31, sep 2018. [Online]. Available: http://www.mdpi.com/2504-446X/2/3/31

[7] J. Wu, "A multi-tiered network with aerial and ground coverage," *Computer Communications*, vol. 131, pp. 39–42, oct 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366418303438

[8] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, "Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Communications*, vol. 15, no. 10, pp. 194–209, oct 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8485481/

[9] D. Reina, H. Tawfik, and S. Toral, "Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks," *Ad Hoc Networks*, vol. 68, no. 1, pp. 16–32, jan 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870517301713?{_}rdoc=1{&}{_}fmt=high{&}{_}origin=gateway{&}{_}docanchor={&}md5=b8429449ccfc9c30159a5f9aeaa92ffb{&}dgcid=raven{_}sd{_}recommender{_}email{&}ccp=y

[10] A. Al-Hourani and K. Gomez, "Modeling Cellular-to-UAV Path-Loss for Suburban Environments," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 82–85, feb 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8048502/

[11] J. Xu, Y. Zeng, and R. Zhang, "UAV-Enabled Wireless Power Transfer: Trajectory Design and Energy Optimization," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8365881/

[12] S. Jeong, O. Simeone, and J. Kang, "Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, mar 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7932157/

[13] H. Q. Qadori, Z. A. Zulkarnain, Z. M. Hanapi, and S. Subramaniam, "Multi-mobile agent itinerary planning algorithms for data gathering in wireless sensor networks: A review paper," *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, p. 155014771668484, jan 2017. [Online]. Available: http://journals.sagepub.com/doi/abs/10.1177/1550147716684841

[14] G. Secinti, P. B. Darian, B. Canberk, and K. R. Chowdhury, "SDNs in the Sky: Robust End-to-End Connectivity for Aerial Vehicular Networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 16–21, jan 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8255732/

[15] C. A. Kerrache, A. Lakas, N. Lagraa, and E. Barka, "UAV-assisted technique for the detection of malicious and selfish nodes in VANETs," *Vehicular Communications*, vol. 11, pp. 1–11, jan 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209617300797?{_}rdoc=1{&}{_}fmt=high{&}{_}origin=gateway{&}{_}docanchor={&}md5=b8429449ccfc9c30159a5f9aeaa92ffb{&}dgcid=raven{_}sd{_}recommender{_}email{&}ccp=y

[16] T. S. Alemayehu and J.-H. Kim, "Efficient Nearest Neighbor Heuristic TSP Algorithms for Reducing Data Acquisition Latency of UAV Relay WSN," *Wireless Personal Communications*, vol. 1, no. 1, pp. 1–15, feb 2017. [Online]. Available: http://link.springer.com/10.1007/s11277-017-3994-9http://dx.doi.org/10.1007/s11277-017-3994-9

[17] S. Rashed and M. Soyturk, "Analyzing the Effects of UAV Mobility Patterns on Data Collection in Wireless Sensor Networks," *Sensors*, vol. 17, no. 2, p. 413, feb 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/2/413

[18] M.-A. Messous, S.-M. Senouci, and H. Sedjelmaci, "Network connectivity and area coverage for UAV fleet mobility model with energy constraint," in *2016 IEEE Wireless Communications and Networking Conference*. IEEE, apr 2016, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7565125/

[19] Z. Xue, J. Wang, Q. Shi, G. Ding, and Q. Wu, "Time-Frequency Scheduling and Power Optimization for Reliable Multiple UAV Communications," *IEEE Access*, vol. 6, pp. 3992–4005, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8249800/

[20] N. Farmani, C. Qian, D. Pack, D. Akopian, and Y. Cao, "Optimal Cooperative Sensing Using Multiple Sensor Platforms," Phd Dissertation, The University of Texas at San Antonio, 2016. [Online]. Available: https://search.proquest.com/docview/1865332108?pq-origsite=gscholar

[21] D. F. Pigatto, L. Gonçalves, G. F. Roberto, J. F. Rodrigues Filho, N. B. Floro da Silva, A. R. Pinto, and K. R. Lucas Jaquie Castelo Branco, "The HAMSTER Data Communication Architecture for Unmanned Aerial, Ground and Aquatic Systems," *Journal of Intelligent & Robotic Systems*, pp. 1–19, mar 2016. [Online]. Available: http://link.springer.com/10.1007/s10846-016-0356-x

[22] B. Olivieri and M. Endler, "DADCA: An Efficient Distributed Algorithm for Aerial DataCollection from Wireless Sensors Networks by UAVs," *The 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'17)*, pp. 129–136, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3127553

[23] CNN, "Floods 2018." [Online]. Available: https://edition.cnn.com/2018/07/13/asia/japan-floods-intl/index.html

[24] ——, "Italy earthquake." [Online]. Available: https://edition.cnn.com/2016/08/23/europe/italy-earthquake/index.html

[25] UNESCO, "Tsunami." [Online]. Available: http://itic.ioc-unesco.org/index.php?option=com{_}content{&}view=category{&}layout=blog{&}id=1160{&}Itemid=1077

[26] Wikipedia, "Hurricanes." [Online]. Available: https://en.wikipedia.org/wiki/List{_}of{_}deadliest{_}Atlantic{_}hurricanes

[27] ——, "Tsunamis detection." [Online]. Available: https://en.wikipedia.org/wiki/Pacific{_}Tsunami{_}Warning{_}Center

[28] NOAA-US, "Tsunami detection NOAA." [Online]. Available: http://www.cpc.ncep.noaa.gov/products/outlooks/hurricane.shtml

[29] Wiki, "Hurricane deads." [Online]. Available: https://en.wikipedia.org/wiki/List{_}of{_}deadliest{_}Atlantic{_}hurricanes

[30] Wikipedia, "SAR." [Online]. Available: https://en.wikipedia.org/wiki/Search{_}and{_}rescue

[31] K. Akkaya, M. Younis, and M. Bangad, "Sink repositioning for enhanced performance in wireless sensor networks," *Computer Networks*, vol. 49, no. 4, pp. 512–534, 2005.

[32] A. Das and D. Dutta, "Data acquisition in multiple-sink sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 82–85, 2005.

[33] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, “Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines,” in *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, IEEE. Lisbon, Portugal: IEEE, 2004, pp. 296–305.

[34] S. R. Nikhitha and M. Panda, “Optimal Sensor Data Harvesting Using A Mobile Sink,” *Procedia Computer Science*, vol. 143, pp. 921–930, jan 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918320398

[35] F. Restuccia and S. K. Das, “Lifetime optimization with QoS of sensor networks with uncontrollable mobile sinks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, IEEE. Boston, MA, USA: IEEE, 2015, pp. 1–9.

[36] D. Turgut and L. Bölöni, “Heuristic approaches for transmission scheduling in sensor networks with multiple mobile sinks,” *The Computer Journal*, vol. 1, no. 1, p. bxp110, 2009.

[37] A. T. Erman, L. van Hoesel, P. Havinga, and J. Wu, “Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management,” *IEEE Wireless Communications*, vol. 15, no. 6, pp. 38–46, 2008.

[38] Q. Yang and S.-J. Yoo, “Optimal UAV Path Planning: Sensing Data Acquisition Over IoT Sensor Networks Using Multi-Objective Bio-Inspired Algorithms,” *IEEE Access*, vol. 6, pp. 13 671–13 684, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8314824/

[39] D. Zorbas, P. Raveneau, Y. Ghamri-Doudane, and C. Douligeris, “The charger positioning problem in clustered RF-power harvesting wireless sensor networks,” *Ad Hoc Networks*, vol. 78, pp. 42–53, sep 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518302440?dgcid=raven{_}sd{_}recommender{_}email

[40] W. Yue and Z. Jiang, “Path Planning for UAV to Collect Sensors Data Based on Spiral Decomposition,” *Procedia Computer Science*, vol. 131, pp. 873–879, jan 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918306719

[41] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous*

Robots, vol. 35, no. 4, pp. 287–300, jul 2013. [Online]. Available: http://link.springer.com/10.1007/s10514-013-9349-9

[42] Q. J. Lindsey, D. Mellinger, and V. Kumar, "Construction of Cubic Structures with Quadrotor Teams," *Robotics: Science and Systems*, 2011.

[43] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP Multiple Micro-UAV Testbed," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 56–65, 2010.

[44] Y. Song and J. M. O'Kane, "Forming repeating patterns of mobile robots: A provably correct decentralized algorithm," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2016, pp. 5737–5744. [Online]. Available: http://ieeexplore.ieee.org/document/7759844/

[45] A. Kaswan, A. Tomar, and P. K. Jana, "An efficient scheduling scheme for mobile charger in on-demand wireless rechargeable sensor networks," *Journal of Network and Computer Applications*, vol. 114, pp. 123–134, jul 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804518300687?dgcid=raven{_}sd{_}recommender{_}email

[46] A. Mazayev, N. Correia, and G. Schütz, "Data Gathering in Wireless Sensor Networks Using Unmanned Aerial Vehicles," *International Journal of Wireless Information Networks*, vol. 23, no. 4, pp. 297–309, dec 2016.

[47] D.-T. Ho, E. I. Grøtli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of Wireless Sensor Network and UAV Data Acquisition," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 1, pp. 159–179, apr 2015. [Online]. Available: http://link.springer.com/10.1007/s10846-015-0175-5

[48] J. R. Martinez-de Dios, K. Lferd, A. de San Bernabé, G. Núñez, A. Torres-González, and A. Ollero, "Cooperation Between UAS and Wireless Sensor Networks for Efficient Data Collection in Large Environments," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 491–508, aug 2013. [Online]. Available: http://link.springer.com/10.1007/s10846-012-9733-2

[49] J. Wang, J. Cao, S. Ji, and J. H. Park, "Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor

networks with mobile sinks," *The Journal of Supercomputing*, vol. 1, no. 1, pp. 1–14, jan 2017. [Online]. Available: http://link.springer.com/10.1007/s11227-016-1947-9

[50] C. Wang, F. Ma, J. Yan, D. De, and S. K. Das, "Efficient Aerial Data Collection with UAV in Large-Scale Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, no. 11, pp. 1–19, nov 2015. [Online]. Available: http://www.hindawi.com/journals/ijdsn/2015/286080/

[51] A. P. Lamping, J. N. Ouwerkerk, and K. Cohen, "Multi-UAV Control and Supervision with ROS," *2018 Aviation Technology, Integration, and Operations Conference*, no. July, 2018. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2018-4245

[52] FlytBase, "FlytBase: Build Smart and Scalable Drone Applications." [Online]. Available: https://flytbase.com/?cn-reloaded=1

[53] Erle Robotics, "Erle Robotics | Intelligent Robotics." [Online]. Available: https://erlerobotics.com/blog/

[54] R. O. System, "ROS.org | Powering the world's robots," 2018. [Online]. Available: http://www.ros.org/

[55] V. Sharma, I. You, and R. Kumar, "Energy Efficient Data Dissemination in Multi-UAV Coordinated Wireless Sensor Networks," *Mobile Information Systems*, vol. 2016, pp. 1–13, 2016. [Online]. Available: http://www.hindawi.com/journals/misy/2016/8475820/

[56] I. Jawhar, N. Mohamed, J. Al-Jaroodi, S. Zhang, I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "A Framework for Using Unmanned Aerial Vehicles for Data Collection in Linear Wireless Sensor Networks," *J Intell Robot Syst*, vol. 74, pp. 437–453, 2014.

[57] D. Pascarella, S. Venticinque, and R. Aversa, "Agent-based design for UAV mission planning," in *Proceedings - 2013 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2013.* Compiegne, France: IEEE, 2013, pp. 76–83.

[58] D. Thakur, M. Likhachev, J. Keller, V. Kumar, V. Dobrokhodov, K. Jones, J. Wurz, and I. Kaminer, "Planning for opportunistic surveillance with multiple robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Vancouver, Canada: IEEE, nov 2013, pp. 5750–5757.

[59] Q. Wu, P. Sun, and A. Boukerche, "An Energy-efficient UAV-based Data Aggregation Protocol in Wireless Sensor Networks," in *Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications - DIVANet'18*. New York, New York, USA: ACM Press, 2018, pp. 34–40. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3272036.3272047

[60] P. Ladosz, H. Oh, and W.-H. Chen, "Trajectory Planning for Communication Relay Unmanned Aerial Vehicles in Urban Dynamic Environments," *Journal of Intelligent & Robotic Systems*, vol. 1, no. 1, pp. 1–19, jan 2017. [Online]. Available: http://link.springer.com/10.1007/s10846-017-0484-y

[61] W. Wang, H. Shi, P. Huang, F. Wu, D. Fang, X. Chen, and X. Yin, "An Efficient Variable Dimension PSO Algorithm for Mobile Node Tour Planning in WSN," in *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition - CSAR '15*. New York, New York, USA: ACM Press, 2015, pp. 47–52. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2820716.2820723

[62] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-Behaved Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1451–1465, nov 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6525428

[63] S. Kashuba, V. Novikov, O. Lysenko, and I. Alekseeva, "Optimization of UAV path for wireless sensor network data gathering," in *2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. Kyiv, Ukraine.: IEEE, oct 2015, pp. 280–283. [Online]. Available: http://ieeexplore.ieee.org/document/7346621/

[64] Google, "Google Project Loon," 2018. [Online]. Available: https://www.google.com/loon

[65] Facebook, "Facebook Aquila." [Online]. Available: https://web.facebook.com/notes/mark-zuckerberg/the-technology-behind-aquila/10153916136506634/?{_}rdc=1{&}{_}rdr

[66] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity," *IEEE*

Transactions on Communications, vol. 66, no. 1, pp. 330–344, jan 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8017572/

[67] Q. Wu, Y. Zeng, and R. Zhang, "Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, mar 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8247211/

[68] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "UAV-Aided Projection-Based Compressive Data Gathering in Wireless Sensor Networks," *IEEE Internet of Things Journal*, pp. 1–1, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8515012/

[69] S. Fu, L. Zhao, Z. Su, X. Jian, S. Fu, L. Zhao, Z. Su, and X. Jian, "UAV Based Relay for Wireless Sensor Networks in 5G Systems," *Sensors*, vol. 18, no. 8, p. 2413, jul 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/8/2413

[70] S. Sekander, H. Tabassum, and E. Hossain, "Multi-Tier Drone Architecture for 5G/B5G Cellular Networks: Challenges, Trends, and Prospects," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 96–103, mar 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8316776/

[71] M. Thammawichai, S. P. Baliyarasimhuni, E. C. Kerrigan, and J. B. Sousa, "Optimizing Communication and Computation for Multi-UAV Information Gathering Applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 601–615, apr 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8062796/

[72] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, jan 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870517301671?{_}rdoc=1{&}{_}fmt=high{&}{_}origin=gateway{&}{_}docanchor={&}md5=b8429449ccfc9c30159a5f9aeaa92ffb{&}dgcid=raven{_}sd{_}recommender{_}email

[73] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7317490/

[74] H. Cao, Y. Liu, X. Yue, W. Zhu, H. Cao, Y. Liu, X. Yue, and W. Zhu, "Cloud-Assisted UAV Data Collection for Multiple Emerging Events in Distributed WSNs," *Sensors*, vol. 17, no. 8, p. 1818, aug 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/8/1818

[75] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, "Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols," in *2017 1st International Conference on Latest Trends in Electrical Engineering and Computing Technologies, INTELLECT 2017*, vol. 2018-Janua, 2018.

[76] X. Ma, S. Chisiu, R. Kacimi, and R. Dhaou, "Opportunistic communications in WSN using UAV," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, jan 2017, pp. 510–515. [Online]. Available: http://ieeexplore.ieee.org/document/7983160/

[77] S. Say, H. Inata, M. E. Ernawan, Z. Pan, J. Liu, and S. Shimamoto, "Partnership and data forwarding model for data acquisition in UAV-aided sensor networks," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, jan 2017, pp. 933–938. [Online]. Available: http://ieeexplore.ieee.org/document/7983258/

[78] S. Mori, "Cooperative sensing data collecting framework by using unmanned aircraft vehicle in wireless sensor network," in *2016 IEEE International Conference on Communications (ICC)*. Kuala Lumpur, Malaysia: IEEE, may 2016, pp. 1–6.

[79] M. Ben Ghorbel, D. Rodriguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Energy Efficient Data Collection for Wireless Sensors Using Drones," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, jun 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8417798/

[80] M. Hua, Y. Wang, Z. Zhang, C. Li, Y. Huang, and L. Yang, "Power-Efficient Communication in UAV-Aided Wireless Sensor Networks," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1264–1267, jun 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8329973/

[81] Y. Zeng and R. Zhang, "Energy-Efficient UAV Communication With Trajectory Optimization," *IEEE Transactions on Wireless*

Communications, vol. 16, no. 6, pp. 3747–3760, jun 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7888557/

[82] S. Berrahal, J.-H. Kim, S. Rekhis, N. Boudriga, D. Wilkins, and J. Acevedo, "Unmanned aircraft vehicle assisted WSN-based border surveillance," in *2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, sep 2015, pp. 132–137. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7314129

[83] S. berrahal.sarra@gmail.com Berrahal, Jong-Hoon Kim2 kimj@cis.fiu.edu, S. slim.rekhis@gmail.com Rekhis, N. noure.boudriga2@gmail.com Boudriga, D. dwilk014@fiu.edu Wilkins, and J. jacev030@fiu.edu Acevedo, "Border surveillance monitoring using Quadcopter UAV-Aided Wireless Sensor N...: Discovery Service para a PUC-Rio," *Journal of Communications Software & Systems*, no. 1, pp. p67–p82. [Online]. Available: https://goo.gl/anb7bo,volume={12},year={2016}

[84] P. Sun and A. Boukerche, "Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network," *Computer Networks*, vol. 146, pp. 217–231, dec 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618309782

[85] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 3465–3471.

[86] J. Burman, J. Hespanha, U. Madhow, J. Isaacs, S. Venkateswaran, and T. Pham, "Autonomous UAV persistent surveillance using bio-inspired strategies," in *SPIE Defense, Security, and Sensing.* Baltimore, Maryland, USA: International Society for Optics and Photonics, 2012, p. 13.

[87] E. I. Grotli, P. B. Sujit, T. A. Johansen, J. Borges de Sousa, D.-T. Ho, E. I. Grotli, P. B. Sujit, T. A. Johansen, and J. B. De Sousa, "Performance evaluation of cooperative relay and Particle Swarm Optimization path planning for UAV and wireless sensor network," in *2013 IEEE Globecom Workshops (GC Wkshps).* Atlanta, USA: IEEE, dec 2013, pp. 1403–1408.

[88] B. Olivieri and M. Endler, "An Algorithm for Aerial Data Collection from Wireless Sensors Networks by Groups of UAVs," *The 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017)*, p. 6, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8202262/

[89] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, oct 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221799002842

[90] D. and Kingston, R. Beard, and R. S. Randal W and Holt, "Decentralized Perimeter Surveillance Using a Team of UAVs," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, dec 2008.

[91] D. Kingston, "Decentralized control of multiple UAVs for perimeter and target surveillance," Ph.D. dissertation, Brigham Young University, 2007. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract{%}5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.6091{&}rep=rep1{&}type=pdf

[92] W. Ren, R. Beard, and T. Mclain, "Coordination Variables and Consensus Building in Multiple Vehicle Systems," in *Lecture Notes in Control and Information Sciences.* Springer-Verlag Series, 2004, vol. 309, ch. Coordinati. [Online]. Available: https://scholarsarchive.byu.edu/facpub

[93] B. ScholarsArchive, R. Beard, and T. W. McLain, "Coordination Variables, Coordination Functions, and Cooperative Timing Missions," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 150–161, 2005. [Online]. Available: https://scholarsarchive.byu.edu/facpub://scholarsarchive.byu.edu/facpub/1225

[94] D. Zurich, "Sinalgo-simulator for network algorithms," p. 9, 2013. [Online]. Available: http://www.dcg.ethz.ch/projects/sinalgo/

[95] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[96] E. Zurich, "81 hours fly." [Online]. Available: http://www.atlantiksolar.ethz.ch/?p=670

[97] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, R. Lotz, D. Siebenmann, S. Leutenegger, K. Alexis, and R. Siegwart, "A Solar-Powered

Hand-Launchable UAV for Low-Altitude Multi-Day Continuous Flight," in *IEEE Robotics and Automation Society's - ICRA*, Seattle - WA - USA, 2015.

[98] D. In.c, "Xbee S1." [Online]. Available: https://www.digi.com/resources/documentation/digidocs/pdfs/90000982.pdf

[99] W. Cook, "Concorde traveling salesman problem solver and libraries." [Online]. Available: http://www.math.uwaterloo.ca/tsp/concorde/

[100] K. Helsgaun, "An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems," Tech. Rep., 2017. [Online]. Available: http://www.akira.ruc.dk/{~}keld/research/LKH/LKH-3{_}REPORT.pdf

[101] C. Silva, L. C. G. Pimentel, L. Landau, P. F. L. H. Filho, F. G. R. Gobbo, and P. de Jesus de Sousa, "Supportive elements to the decision-making process in the emergency planning of the Angra dos Reis Nuclear Power Complex, Brazil," *Environmental Earth Sciences*, vol. 76, no. 3, p. 133, feb 2017. [Online]. Available: http://link.springer.com/10.1007/s12665-017-6450-y

[102] N. Intel, 3DR, Yunteec, "Dronecode Frramework." [Online]. Available: https://www.dronecode.org/about/