Journal of Internet Services
and Applications

RESEARCH Open Access

Evaluating flight coordination approaches of UAV squads for WSN data collection enhancing the internet range on WSN data collection

Bruno José Olivieri de Souza* and Markus Endler

*Correspondence:
bruno@olivieri.com.br
Departamento de Informática,
Pontifícia Universidade Católica do
Rio de Janeiro, Rua Marquês de São
Vicente, 225 RDC, Rio de Janeiro,
Brasil

Abstract

Wireless sensor networks (WSNs) are an important means of collecting data in a variety of situations, such as the monitoring of large or hazardous areas. The retrieval of WSN data can yield better results through the use of unmanned aerial vehicles (UAVs), for example, concerning the increase in the amount of data collected and the decrease in the time between the collection and use of the data. In particular, disaster areas may be left without communication resources and with high residual risk to humans, at which point a WSN can be quickly launched by air to collect relevant data until other measures can be established. The set of rules of each problem's component (e.g., number of UAVs, UAVs dislocation control, sensors, communication) is considered the approaches to solve the problem. In this meaning, some studies present approaches for the use of UAVs for the collection of WSN data, focusing primarily 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 reveal that the amount of data collected by DADCA is similar or superior to path optimization approaches by up to 1%. In our 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 presented approaches, since the path optimization is not a focus, while also incorporating the advantages of a distributed approach.

Keywords: Unmanned Aereal Vehicles, FOG computing, Data collection, Wireles sensors networks, Distributed algorithms, Swarm

1 Introduction

This work extends and details a distributed approach for coordinating multiple unmanned aerial vehicles (UAVs) for collecting data from wireless sensor networks (WSNs). Wireless sensor networks are deployed to collect environmental data for various applications (e.g., seismic monitoring, wildlife tracking, soil, air condition control, etc.). Data collection and transmission are the fundamental operations of WSNs [1]. However, in scenarios in which no infrastructure or regions lack communication, the data collection from sensors of the WSN encounters serious difficulties [2]. Extensive research has been conducted on several types of mobile agents that serve as data collectors for WSNs [3]. The use of UAVs as such mobile agents can enhance the 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, thus increasing the network's capacity and flexibility [4].

The coordination of UAVs in a distributed manner is a complex task [5] because there is no central node that can have a consistent view of the state of all the UAVs (i.e., the instantaneous position, velocity, residual battery level); furthermore, the ad-hoc communication topology limits consensus on deciding how to perform a task sharing of a global mission. This work applies the term ad-hoc communication in the meaning of non-constant communication based on a certain level of proximity. Therefore, such coordination requires effective algorithms to overcome these constraints. Providing a guarantee of effectiveness could threaten the efficiency regarding the amount of collected data or the delay in collecting data from WSNs by UAVs. Unmanned aerial vehicles have exhibited tremendous growth, in both research and applied use [6]. Concerning the swarm of UAVs, the most common model for using UAVs relies on the use of 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 traveling 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 manner. It remains challenging to develop cooperative UAVs on area coverage tasks and energy-efficient UAV communication technology [8]. Moreover, as we aim to employ a fully distributed approach, path planning should be computed in the UAVs regarding its computational constraints.

To address these issues, this work 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 UAV states and relying on only ad-hoc communication. In response to these questions, we previously proposed the DADCA core idea in [9] and some better formulations in [10]. DADCA is a distributed algorithm that combines well-known algorithms of path planning with a cooperative and oscillatory behavior that relies on only the exchange of all payload data from pairs of UAVs' (whenever they approach) decision making.

The main contributions of this work are as follows:

- Extends and details a proposal of a distributed approach that coordinates the flight movement of UAVs for the collection of WSN data, assuming that the positions of the CHs to be visited are known in advance;

- Provides an extensive discussion of the experimental results and analysis of trade-offs based on the selected criteria and other approaches presented.

The use of unmanned aerial vehicles (UAVs), otherwise known as *drones*, has significantly increased in the past decade for several purposes [11]. Unmanned aerial vehicles offer agile and cost-effective solutions for many demanding civilian applications [12] and have drawn significant research interest in recent years due to their wide range of applications, including surveillance and monitoring, footage in movies, sporting events, inventory verification and inspection, cargo delivery, communication platforms, rural environment inspection, and disaster response and emergency relief [13, 14]. Such widespread applicability is primarily due to the vast capabilities of aerial vehicles in terms of mobility, autonomy, communication, and processing, in addition to the relatively low cost [11].

The emergence of WSNs has become an active research area in a broad range of critical applications [15]. 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 of time-critical data generated from WSNs on the ground to remote ground stations connected to the internet [16]. Unmanned aerial vehicles can function as mobile data collectors, such as connection nodes in WSNs [17]. In such cases, UAVs can provide these connections by visiting the WSNs periodically and relaying or carrying the data to the proper destination. Recently, interest has grown in data collection through groups of collaborative UAVs [18, 19].

One important application of UAVs is in emergency and rescue (such as earthquakes or nuclear explosions), in which emergency teams can benefit from a rapid and comprehensive monitoring solution as soon as possible and until proper operation can 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 with using only one UAV [20] due to the typical large size of the geographic area and the demand to haul sensor data in a timely fashion. However, while the use of several UAVs can improve the data collection rate, multiple-UAV operations also introduce some challenges [21]. The approaches most commonly applied in these cases are derivatives of the traveling salesman problem (TSP) and vehicular routing problem (VRP).

A reliable communication infrastructure among UAVs in collaboration is critical in maintaining this connected network for data-relaying tasks [16]. Thus, reliable communication requires a radio base station close to the site; however, in some cases, implementing such infrastructure to provide long-range UAV communication is not possible (e.g., in emergency response and relief situations). Some emergencies, such as earthquakes, could destroy 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 such circumstances. In these cases, ad-hoc communication plays an important role.

With the goal of controlling several UAVs in a distributed manner, relying on only vehicle-to-vehicle (V2V) communication, we propose an approach that can be used as a first-time emergency response approach for repetitive visitation of WSN sensors during a T period. In an early phase of this work [10], 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 superior results to those of the TSP in specific scenarios. It does not rely on central control and runs in a distributed manner, with all processing performed only in the UAVs. This algorithm considers the fact that the number of working UAVs can change during the cyclic visitation period T due to UAV malfunctions or replacements, or expand the set of UAVs during its execution without synchronization. In this work, we extend this concept and propose new distributed approaches based on DADCA for aerial data collection. These new approaches focus on the use of various path planning techniques that aim to yield better results in distinct metrics. Furthermore, we 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.

For example, we present the following scenario: After a nuclear accident at a nuclear plant, some rescue tasks must 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 the vast and affected areas by the radiation. For example, several radiation sensors may be placed around the plant, while UAVs can be used to collect data from the sensors or cluster heads (CHs).

This work is organized as follows. In the next section, we present and discuss the main related works that address the aforementioned problems. Section 3 further elaborates the underlying problem and presents criteria for comparison of the approaches: collected data tax, message delays, and processing time in UAVs. Afterward, Section 4 presents the proposed algorithm, DADCA, and related work used as a benchmark. Section 5 then presents the evaluation setup, simulating two real scenarios with possible nuclear issues. Section 6 presents the simulation results and comparisons between the proposed approaches. Finally, Section 7 presents the concluding remarks concerning the proposed approach and presents suggestions for future work.

2 Related work

This section 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.

2.1 Air-to-ground communication

In WSNs, data can be gathered through the use of the following: (1) the classical approach, including a network with infrastructure; (2) mobile elements, such as ground robots or UAVs [22]. We focus on studies regarding aerial data collection from WSNs with UAVs as the mobile elements.

Regarding data collection from WSNs, it is typical to cluster their sensors to save energy. The low-energy adaptive clustering hierarchy (LEACH) is commonly used to select cluster heads (CHs) for the purpose of conserving energy [23]. Unmanned aerial vehicles can collect data from CHs in the same manner 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 [24]. The authors propose the use of 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 its designed WSN areas in order of visitation.

Some related works, such as [25] and [26], divide the entire set of CHs based on 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 induced 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 this 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, frameworks exist that aim to enable multi-UAV control [27–29]; these rely on hardware enablers through the Robot Operating System (ROS) [30]. However, these frameworks provide no distributed coordination investigations. Mazayev et al. have conducted research that considers multiple-UAV data collection [22] but focuses on a heuristic to optimize the UAV buffer size and speed constraints using a centralized approach. Several UAVs working within a WSN are considered by Sharma et al. [31], who 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. [15] identifies a lack of research on collaboration approaches among agents for collecting and summarizing data. The studies they cite investigate multiple agents working side by side independently (as the UAVs in [24]) rather than in collaboration [15]. This study draws an analogy between agents navigating through a WSN and UAVs navigating among CHs in terms of collaboration.

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, 32, 33]. Other works focus on optimization to solve the TSP by applying genetic algorithm (GA) optimization, as in [34, 35], and [26]. Yet others employ particle swarm optimization (PSO), as in [36] and [37]. Considering the radio range as neighborhoods [38], some works discuss TSP-like heuristics, such as [18] and [36]. 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. [26] 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. [26] is one of the benchmarks to which this work's proposed algorithms are compared.

2.3 Air-to-air communication

Data collection problems in sensor network nodes can follow many different approaches. Similar to the Google project Loon [39], Facebook project Aquila [40], and the project

Absolute [41], the work of Zhang et al. [42] focuses on enabling ground communication through the use of several UAVs. In this work, the UAVs function as mobile sink nodes with large throughput and long-range radios. Specifically, in [42], the proposed solution focuses on coverage modeled as a mixed-integer non-convex optimization problem. This work employs UAVs as a mobile base station for enabling aerial data collection as in [43, 44], and [45] for UAVs acting as flying 5G base stations. The authors in [46] discuss how UAVs are constrained by limited resources, including communication. Our work differs in that our approach does not rely on long-range radios in the UAVs.

Yanmaz et al. [47] 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 utilizes the aforementioned ROS [30] 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 on only ad-hoc communication. However, in [47], the data collection is related not to WSN but to UAV onboard censoring, and the distributed decision-making structure focuses on only UAV collisions [47]. The survey by Gupta et al. [48] brings together several studies regarding UAV networks. The most frequently cited network type relies on a backbone that provides long-range communication, as presented in [49] for cloud-assisted data collection. However, in data collection and search-and-rescue scenarios, it is reasonable to use flying ad-hoc networks (FANET) [48], a specific type of mobile ad-hoc network (MANET) [50].

There are exist other lines of inquiry regarding data collection with UAVs beyond tour-planning techniques. Some works focus on opportunistic sensor data collection with UAVs, as discussed in [51], yet still rely on a single UAV. Regardless of the UAV tour planning, some works investigate a low-level communication approach, as described in [52], which provides medium access control (MAC) protocols dedicated to UAVs collecting data from WSNs [53]. Other works focus on UAV or WSN energy efficiency, as discussed in [8, 54, 55], and [56]. Berrahal et al. describe another approach, in which multiple sensors comprise the structure for a WSN that specializes in border surveillance [57], but the UAVs have no collaboration or knowledge of the other UAVs.

2.4 Comparison overview

The Table 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.

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 [15], further research is required on multi-collector approaches, particularly regarding cooperation among them.

Table 1 Related Works

Related Work	WSN relation	Work & Tour	Comm.	UAV	
		approach	approach	Multi-UAV	Coordination
This work	Data collection	Polynomial path planning	ad hoc	Yes	Distributed
Dios et al.[24]	Data collection	TSP straight use	V2I	No	Centralized
Several researches [5, 32–37]	Data collection	TSP optimization	V2I	No	Centralized
Jin Wang et al. [25]	Data collection	Choice of CHs by UAV tour	V2I	Yes, in isolation	Centralized
Wang et al. [26]	Data collection	FPPWR	V2I	No	Centralized
Some researches [43, 44] & [45]	Data collection	Optimization for area Coverage and relay through 5G	V2I	No	Centralized
Mazayev et al.[22]	Data collection with buffer TTL constraints	Tour optimization upon constraints	V2I	Yes, in isolation	Centralized
Ma et al. [51]	Opportunistic data collection from mobile sensors	Predefined static route	ad hoc	No	Centralized
Berrahal et al. [57, 58]	Data collection borders sensoring & video surveillance	Predefined static route	V2I	Yes, in isolation	Centralized
Jawhar et al.[59]	Data collection, but in specific sensor distribution (LSN)	Predefined static route	Hierarchical V2I	Yes, in isolation	Centralized
Yanmaz et al.[47]	Direct sensoring	mTSP straight use	V2I & ad hoc	Yes, in isolation	Centralized
Zhang et al.[42]	Data dissemination w/ constraints	Optimization for area Coverage	V2I	Yes, but static	Centralized
Sharma et al. [31]	Data dissemination	Workload area divided by the number of UAVs	V2I & ad hoc	Yes	Centralized
Thammawichai et al.[46]	Pursue a single target or survey an area of interest for data relay	Constraints optimization	Hierarchical V2I	Yes	Centralized

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 [60, 61]. 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 various strategies to collect data from a ground-deployed WSN using multiple UAVs. Chapter 4 then presents two strategies for collecting data from the WSN with UAVs based on related works and proposes a distributed approach to achieve this objective. In this section, we define the application architecture and metrics used to compare approaches with distinct strategies in such data collection.

There exist different types of sensors for reading the data that will be consumed in different ways. For example, there are sensors as cameras, for which it is sensible to consume the data in the FIFO (first in first out) perspective in a sense to follow the images' evolution. In another perspective, there exist radiation sensors for which it is sensible to acquire the newer sensor data as quickly as possible, so a LIFO (last in first out) data collection makes more sense.

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 [62], 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 [24].

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 assume that each UAV is capable of avoiding collisions.

3.2 System overview

Our model assumes that each UAV, sensor, and CH, as well as mobile ground station (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 aerial data collection by UAVs 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 Fig. 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 (Fig. 1a), the sensors are spread over the area by air using a larger UAV or manned aircraft (Fig. 1b). As noted in Fig. 1c, 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 [63] 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. [22] may be applied.

As indicated in Fig. 2e above, the UAVs keep on collecting data for a period of time *T*. We illustrate a snapshot the entire scenario in Fig. 3, which presents an area of interest after an accident.

A CH remains in passive mode waiting for a UAV flyover. Figure 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 4 illustrates a UAV collecting data from a CH during its stay within a distance S, with S < R. In Fig. 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

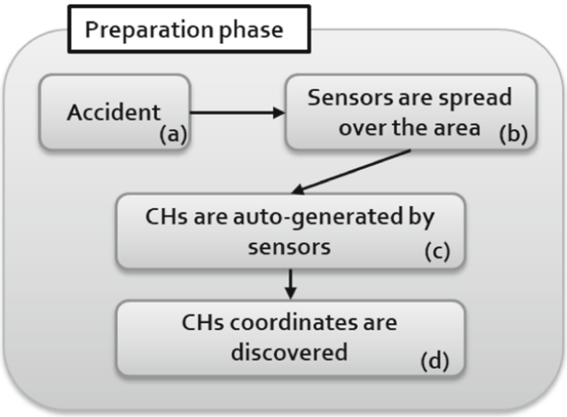

Preparation phase
Accident (a)
Sensors are spread over the area (b)
CHs are auto-generated by sensors (c)
CHs coordinates are discovered (d)

Fig. 1 Figure illustrating actions from the accident to the CH coordinates discover

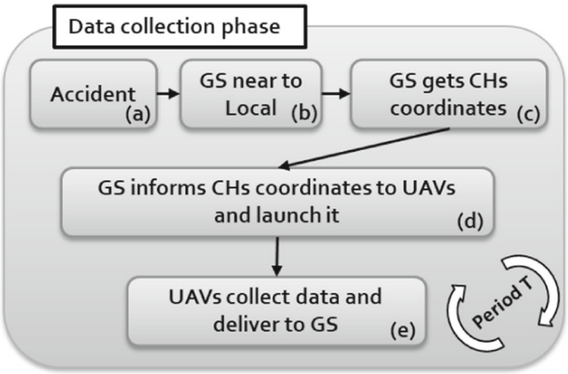

Data collection phase
Accident (a)
GS near to Local (b)
GS gets CHs coordinates (c)
GS informs CHs coordinates to UAVs and launch it (d)
UAVs collect data and deliver to GS (e)
Period T

Fig. 2 Figure illustrating the problem from the GS perspective. Figure 1d information is used here from item (c). Item (e) presents the cyclic data collection

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 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 Fig. 1d.

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.

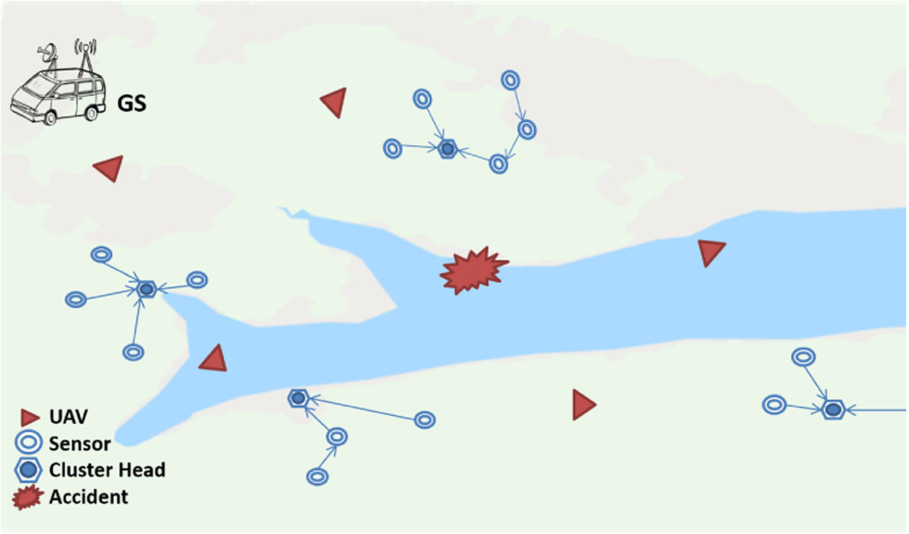


Fig. 3 Figure presenting the interest area with sensors, CH, GS and UAVs flying around

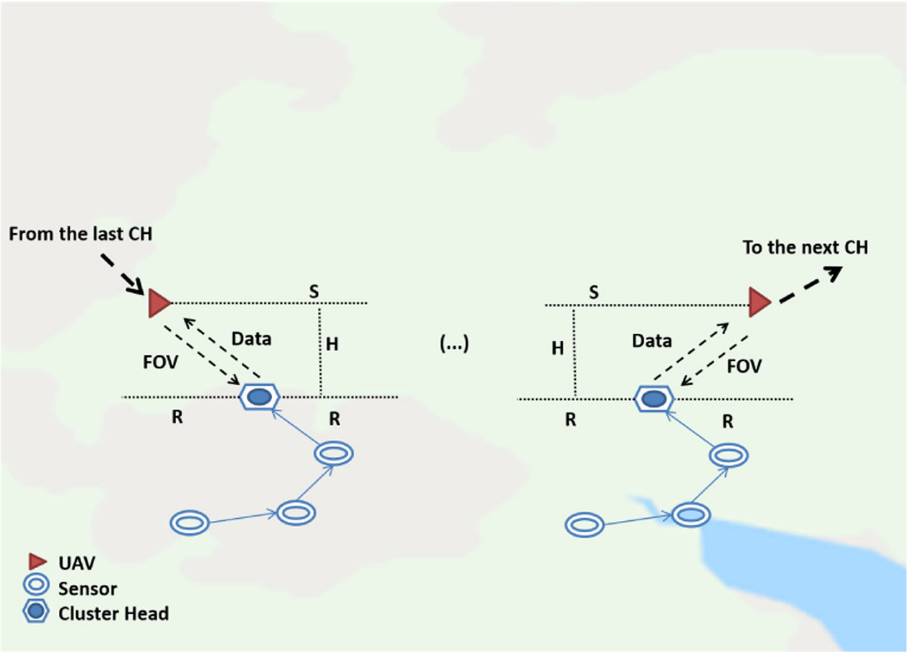


Fig. 4 The Figure presents a UAV before and after the feasible exchange horizontal distance S with the constrain of $S \leq 2\sqrt{H^2 - R^2}$. R is measured on the ground, H is upright from ground and S is parallel to R

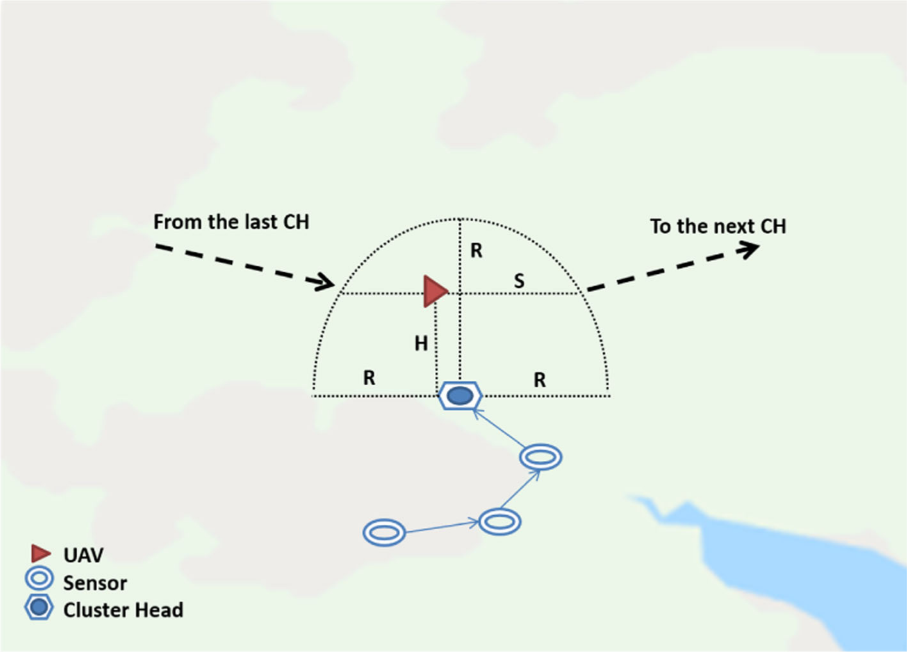


Fig. 5 A UAV inside the range of message exchange with a CH. The images simplify the radio range as a uniform disc, but our simulation does not as present in next chapters

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.

$$C_p^T = \int_0^T \sum_0^p C_i dt \tag{1}$$

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.

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

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.

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

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.

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

$$D_n^T = \frac{M_n^T}{n} \tag{5}$$

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.

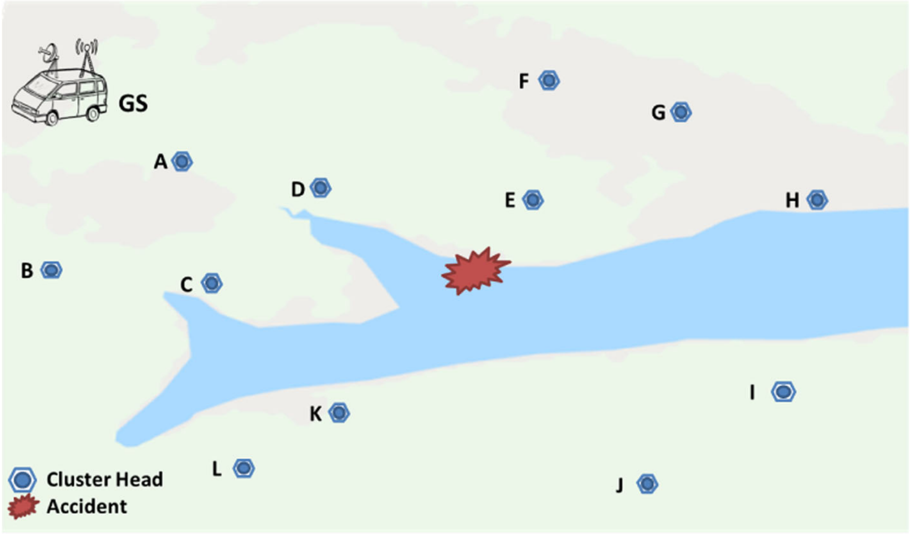


Fig. 6 Scene view with CHs and GS. Each strategy determines how UAVs will visit CHs to collect data to deliver to GS

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 6 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.

The edges of Fig. 6 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 Fig. 6.

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. [64] and Ho et al. [65], 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 [18, 22, 32, 38, 53][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 Fig. 7. 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. [26] 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 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 Fig. 8. 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

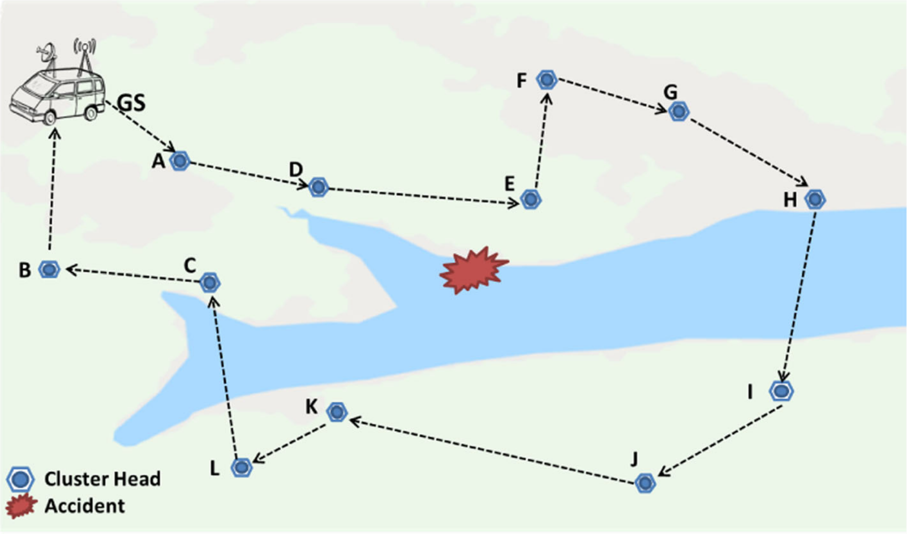


Fig. 7 A possible tour generated by TSP-based strategy

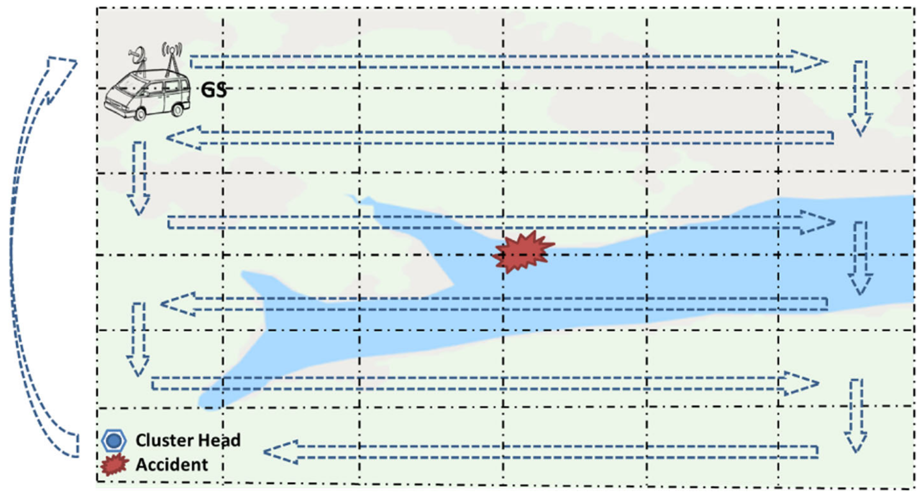


Fig. 8 Clusters distribution and computation order for tour creation by FPPWR strategy

consideration the most recent cluster processed. All clusters are analyzed in the order indicated by the arrows in Fig. 8. Figure 9 illustrates a possible tour generated by FPPWR.

The algorithm complexity is $\mathcal{O}(p \log p)$ [26] (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 [10] that maintains the sensor data collection by UAVs during a period of time T (measured in

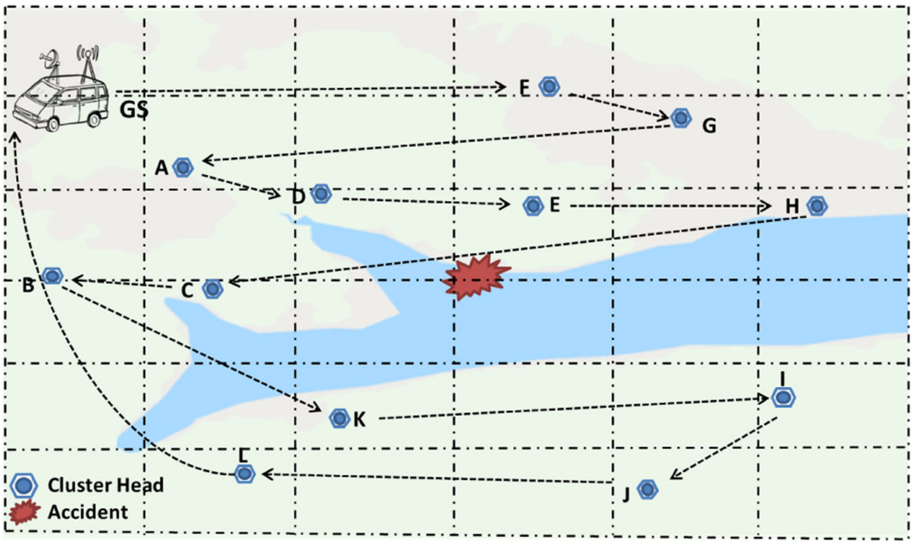


Fig. 9 A possible tour generated by FPPWR strategy

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. An early version of this method and some essential details were introduced in [9], with improvements and further details presented in [10].

In order to present DADCA in details, we divide its presentation into two phases as follows. In Fig. 10 DADCA's main architecture is presented: (1) Figure 10a indicates the computation responsible for planning the tour that the UAVs will follow and (2) Figure 10b is the strategy for using the computed tour in Fig. 10a.

From this point on, the tour computed in Fig. 10a 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 Section 4.3.1. The main characteristics of the tours used in the first phase of the DADCA method, as illustrated in Fig. 10a, 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 Sections 4.3.6 to 4.3.9, we present four distinct tour computation methods that generate non-optimized tours used in Fig. 10a. In Section 4.3.1, we explore the approach used in Fig. 10b.

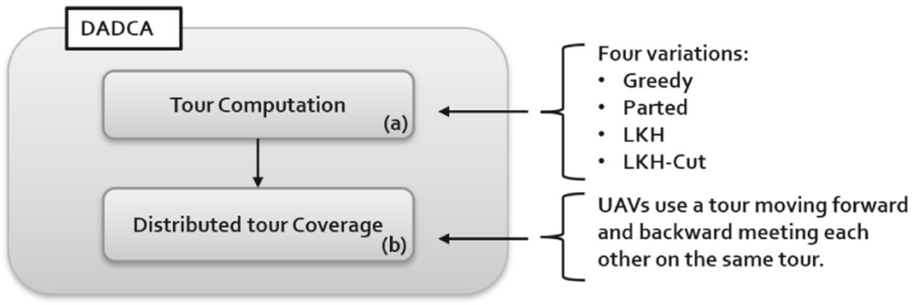

DADCA
Tour Computation
(a)
Distributed tour Coverage
(b)
Four variations:
• Greedy
• Parted
• LKH
• LKH-Cut
UAVs use a tour moving forward and backward meeting each other on the same tour.

Fig. 10 DADCA phases

4.3.1 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.

4.3.2 DADCA - initialization

Once any of the DADCA algorithms from Sections 4.3.6 to 4.3.9 creates a tour, the next phase begins (as indicated in Fig. 10b). This second phase extends and adapts the algorithm originally proposed by Kingston et al. [66]. 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 [66] cannot cross itself.

Algorithm 1: Initialization processes: GS and UAVs

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

The first step of DADCA is the initialization of *GS*and *UAVs*, as presented in Algorithm 1. 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 1, its variables are initialized. Function *RunStrategy()* is responsible for calling one of the planned tour from the algorithms presented in Sections 4.3.6 to 4.3.9. 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.

4.3.3 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 11 illustrates the UAV$_A$ traveling from the *left limit* (A) to the

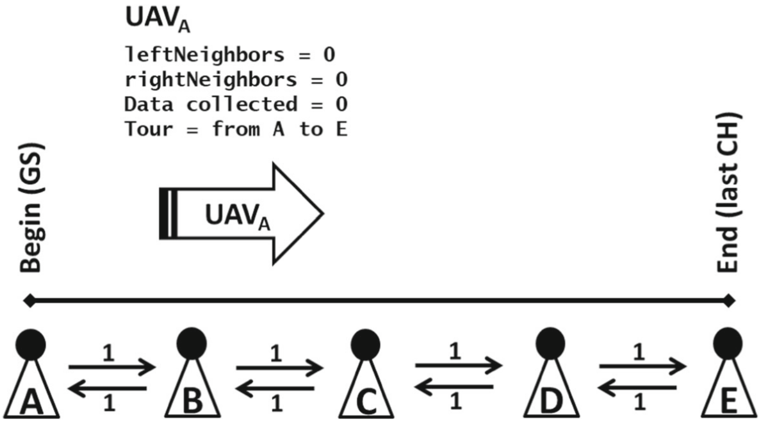


Fig. 11 A possible solution of presented graph by *DADCA* with a single UAV collection data from all CH. The logical segment above CHs represents the UAV tour from A to E

right limit (E) and collecting data. When UAV$_A$ reaches any limit, it returns from the opposite direction.

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 Fig. 11, 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 2 and Algorithm 3, which we describe in the following paragraphs.

Eventually, other UAVs perform their collection, as illustrated in Fig. 12. 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 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

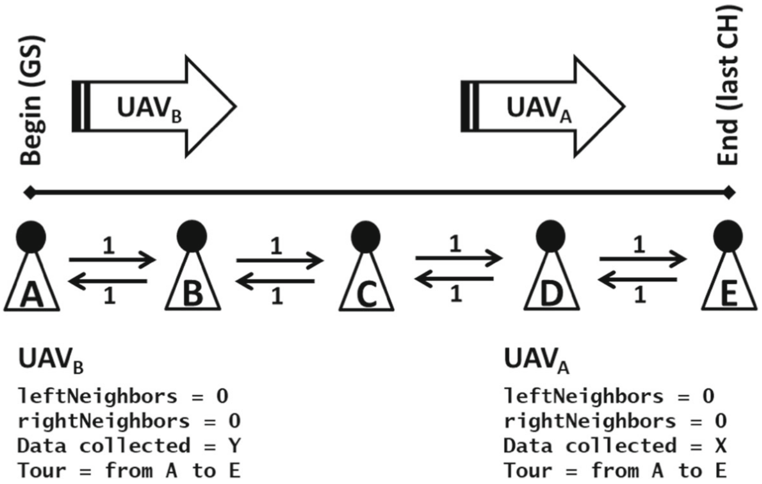


Fig. 12 Two UAVs flying far from each other

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 [66] 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.

4.3.4 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 Fig. 13. This meeting is what we call a rendezvous—at this moment, the UAVs exchange current information about other known UAVs, the data they've collected, and then adjust their sub-tours accordingly. These rendezvous runs are presented in Algorithm 2 and in Algorithm 3.

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 Fig. 12.

Figure 14 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

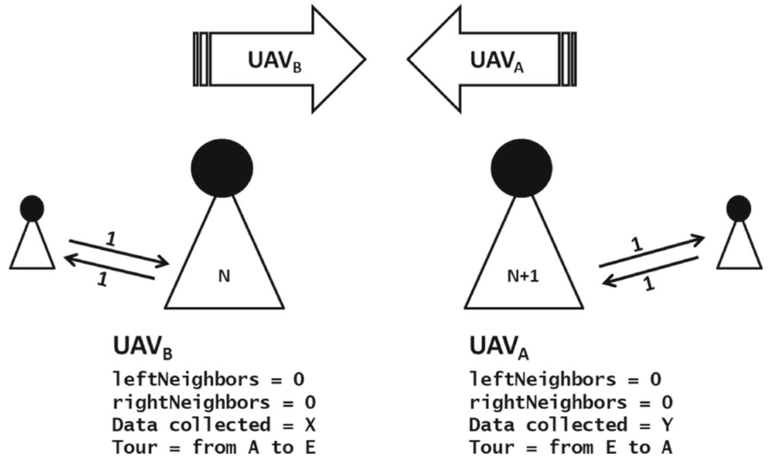


Fig. 13 Two UAVs a moment **before** a rendezvous. Each one with an independent tour and no references about other UAVs until this moment

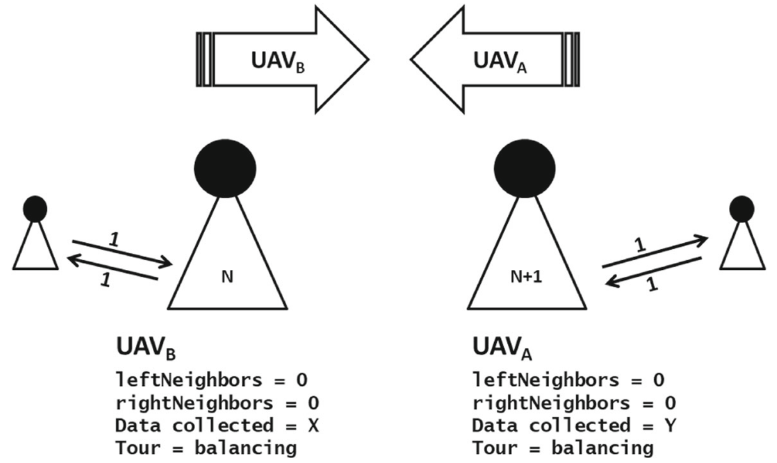


Fig. 14 Two UAVs running DADCA a moment **during** a rendezvous. Each one updating its data with data from the other

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 Fig. 15. 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

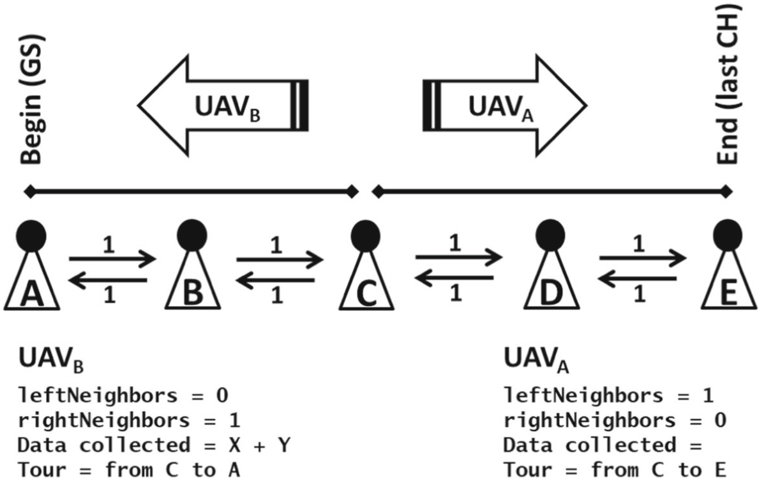


Fig. 15 Two UAV running DADCA after defined the shared border for the last rendezvous. Is this case, cluster head *C*. Both go to *C* and then go to opposite sides. Logical Left UAV carries data from Right UAV

to note that whenever a UAV passes within range of the GS, it delivers all the data in its possession.

4.3.5 DADCA - algorithm

In order to better describe DADCA in Algorithm 2, 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 subtours, 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 [66].

In Algorithm 2 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 3 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.

Algorithm 2: 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 3: 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
```

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 [66]. 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 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 4 demonstrates the DADCA-greedy first phase, with line 1 presenting the function *Navite-Tour()*, 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 [9].

Algorithm 4: DADCA Greedy - Tour Planning

```
1 function $GreedyTour(setOfCH)$:
2     $tour \leftarrow null$
3     $nxtCH \leftarrow null$
4     $tour$.add($GS$)
5     while $tour.size \leq setOfCH.size$ do
6         $nxtCH \leftarrow$ GetNearestCH
            ($tour$.last; $setOfCH$)
7         $tour$.add($nxtCH$)
8         $setOfCH$.remove($nxtCH$)
9     return $tour$
```

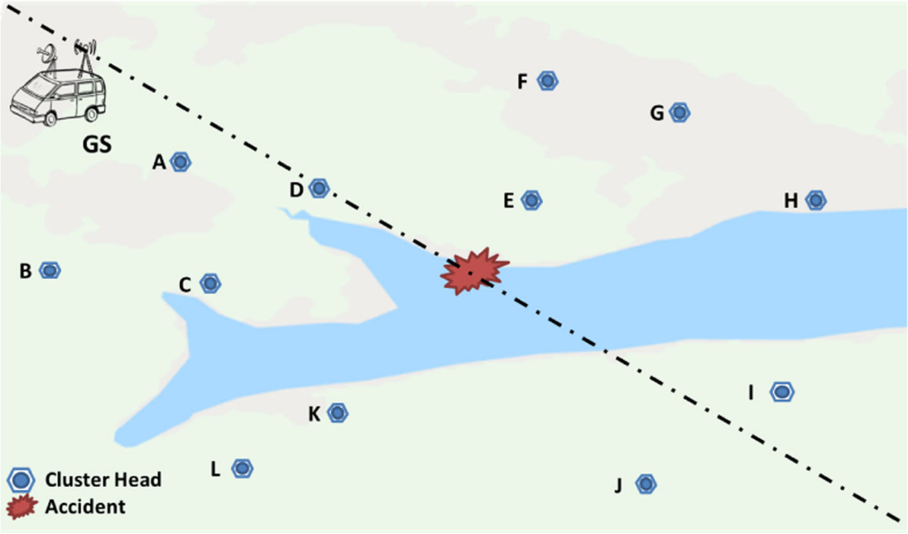


Fig. 16 DADCA-Parted set of CH being splitted based on their coordinates

4.3.7 DADCA-parted

In this DADCA variation, we create the *Original Tour* in two steps, as presented in Algorithm 5, which is polynomial and deterministic. First, the coverage area is split into two geographic parts, as we illustrate in Fig. 16. Lines 6 to 10 in Algorithm 5 represent the creation of two sets of CHs based on their coordinates. (Fig. 17)

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 Fig. 16. 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 $\theta(p^2)$.

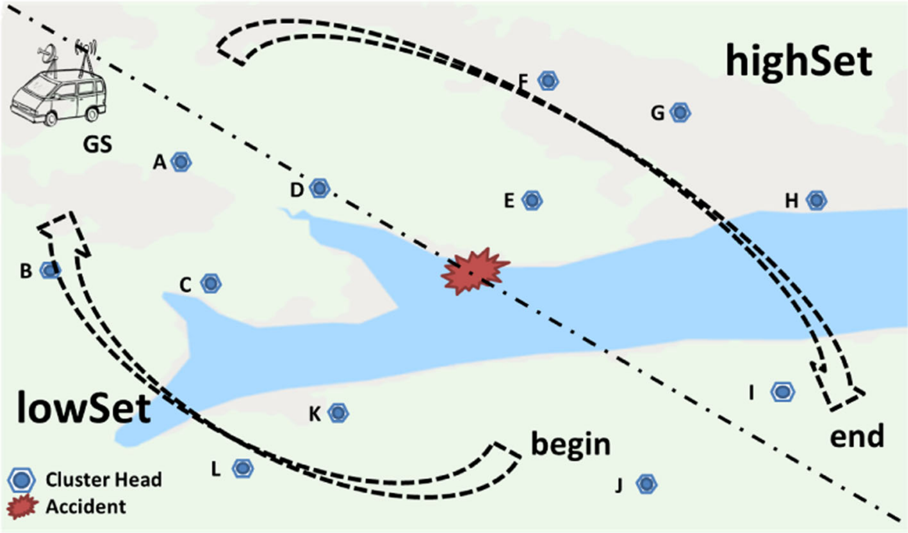


Fig. 17 DADCA-Parted tour scheme

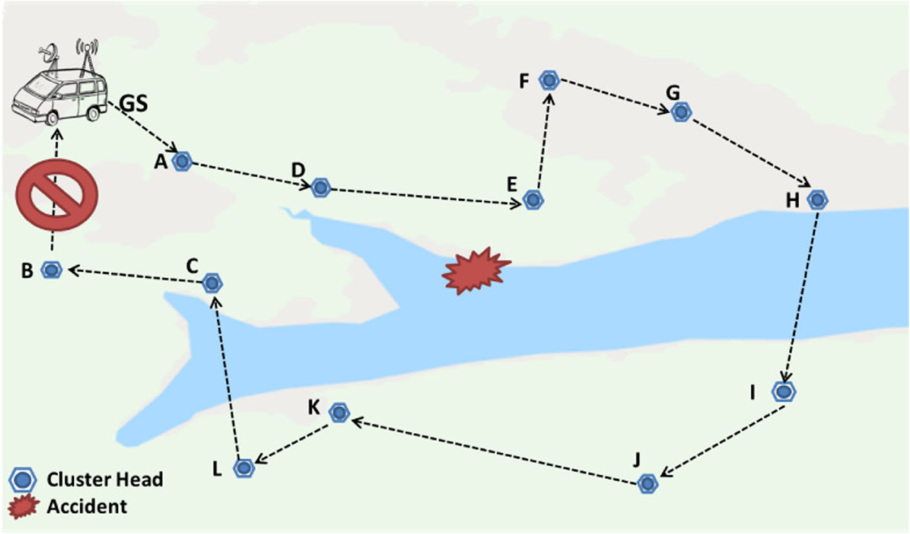


Fig. 18 A sample tour that could be provided by DADCA-LKH

```
Algorithm 5: 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 $tour$
```

4.3.8 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 are produced with high frequency, and it currently holds the record for all instances with unknown optima[1]. Its complexity is $\mathcal{O}(p^{2.2})$ [26, 67].

In DADCA-LKH, LKH is used to compute the best tour that closes a Hamiltonian Cycle in polynomial time, as presented in Algorithm 6, 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 Fig. 18.

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

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.

Algorithm 6: 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 $tour$
```

4.3.9 DADCA-LKH-cut

Finally, we propose another variation on DADCA named DADCA-LKH-Cut, which is described in Algorithm 7. 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 Fig. 19. 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

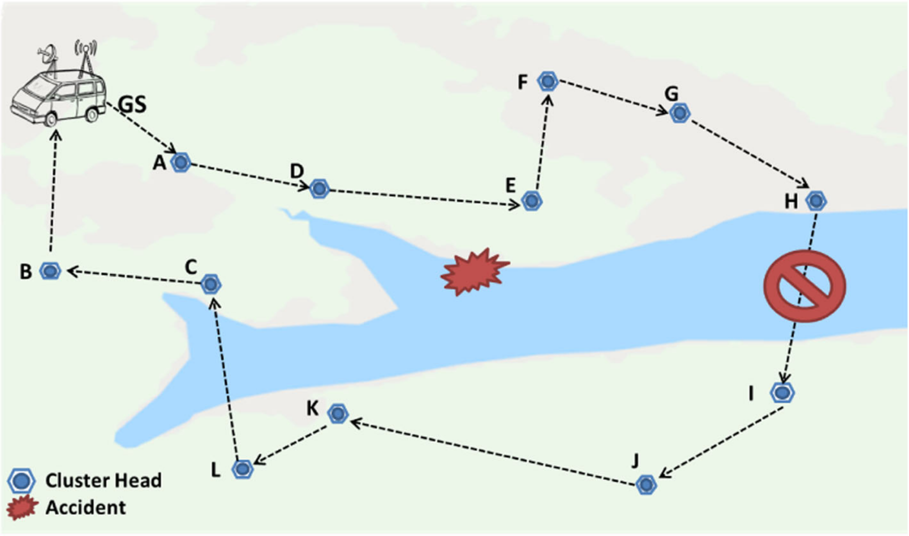


Fig. 19 A tour computed by LKH without a middle edge to be used on DADCA-LKH-cut

instead maintain the GS roughly in the middle of the tour. In addition, this algorithm produces more straightforward results regarding message delays.

Algorithm 7: DADCA LKH Cut - Tour Planning

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

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 [68], 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 [69] standards. Details for each set of simulation items are presented below. Table 2 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:

Table 2 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

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 UAVs

In order to set a reasonable baseline for our 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[70] of flight endurance, as presented by [71].

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 [72], 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 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.

5.5 Solvers

The TSP solution is provided by the use of the Concorde solver [73]. To provide the LKH tour, we also used the original LKH implementation [74].

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. [75].

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 section, we discuss the results of the various experiments, comparing the strategies described in Section 4. The data sets for all the experiments are presented in Section 5.7. In Section 5.7, we present the various tour sizes computed with each of the strategies for the same configuration of CHs on a map. Subsequently, in Section 6.1, we analyze the amount of collected data (TD^T) that each strategy can retrieve during the period T. This section includes an 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 6.2. The resources required in the tour planning of each strategy, in terms of hardware comparable with that available in UAVs, is presented in Section 6.3. The Section 6.4 discusses energy issues.

6.1 Collected data - TD^T

In this evaluation, we present TD^T, formalized as Eq. (3) in Section 3.3. The UAVs are sent to collect data before the end of the period T, and when period T ends, TD^T is measured. Section 6.1.2 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 6.1.3 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 summarize these results in subsection 6.1.4.

6.1.1 Experiment parameters

The results in this section were obtained through simulations, the data sets analyzed in Section 5.7, and the three CHs densities (sparse, dense, and *full*). In addition, the simulations were performed using different sets of UAVs (2, 4, 8, and 16), and the other variables were set according to Table 2. In total, 14,400 simulations were performed.

6.1.2 Analysis of TD^T

For this subsection (6.1.2), the results account for a single set of UAVs (eight UAVs). All the results, save for the outliers (0.2% of the results), are presented in boxplot charts; therefore, 99.8% of the results are included in the charts. The results appear in Fig. 20, grouped based on strategy and the density of CHs.

The vertical axis represents TD^T, while the horizontal axis indicates the various strategies. All charts are ordered based on the series' medians, from left to right; thus, farther right a strategy is placed in the charts, the higher are the TD^T results.

Based on the charts displayed in Fig. 20, it is possible to cluster the series into three groups. This group formation is more evident in the *full* scenario, as displayed in Fig. 20c. 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 achieve significantly better results than FPPWR. The DADCA-LKH, DADCA-LKH-Cut and TSP-based strategies, which aim to create shorter tours, comprise the third group with the best results. The strategy DADCA-LKH achieves the best results because it has shorter tours than the other strategies.

The DADCA-LKH-Cut strategy produces better results than the TSP-based strategy in the sparse scenario, whereas the TSP-based strategy produces the second-best results in the dense and *full* scenarios. These results are due to the efficiency of the DADCA variations in the use of the generated tours. More specifically, the DADCA-LKH-Cut strategy has shorter subtours than the DADCA-LKH strategy and worse results than the TSP-based strategy.

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

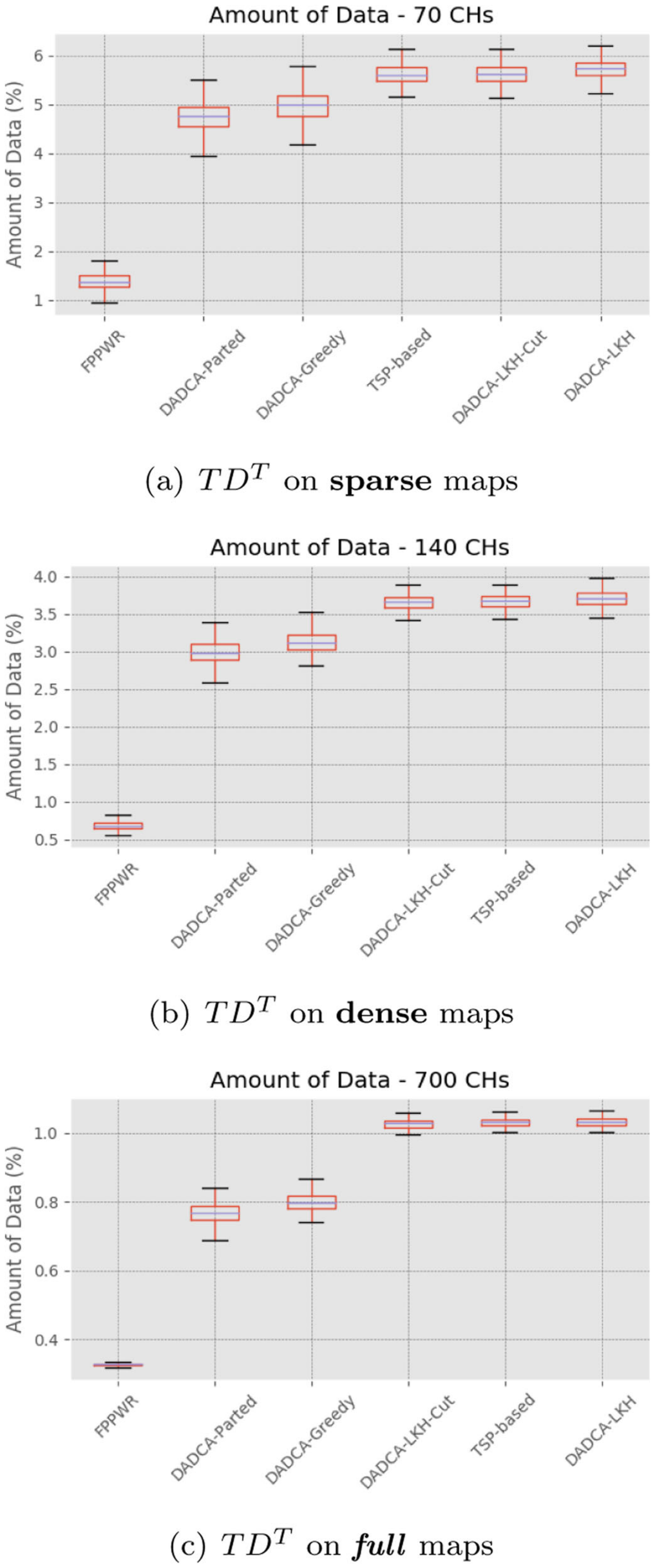
6.1.3 Impact of the number of uAVs on TD^T

In this subsection, we present the same experiments from the previous subsection and include different numbers of UAVs and outlier results. For all charts in Fig. 21, there is a series with the results of a running strategy. Each point in a series represents the average from all executions in that density of CHs for all maps from the data set.

For each series in the charts depicted in Fig. 21, one notices that an increase in the number of UAVs results in a direct increase in TD^T for each experiment. This linear dependency is created because across all strategies, the UAVs collect data by flying over CHs at a constant speed in a computed tour, and therefore, the more aircraft that fly, the more data is collected.

6.1.4 Experiment summary

In general scenarios, the DADCA variation results in nearly the same or even more TD^T than do the other strategies. Indeed, the individual analysis of each case in which DADCA versions was less efficient than TSP indicated that this was due to the two UAVs flying side by side, resulting in the UAVs being unable to obtain from CH at the same time.

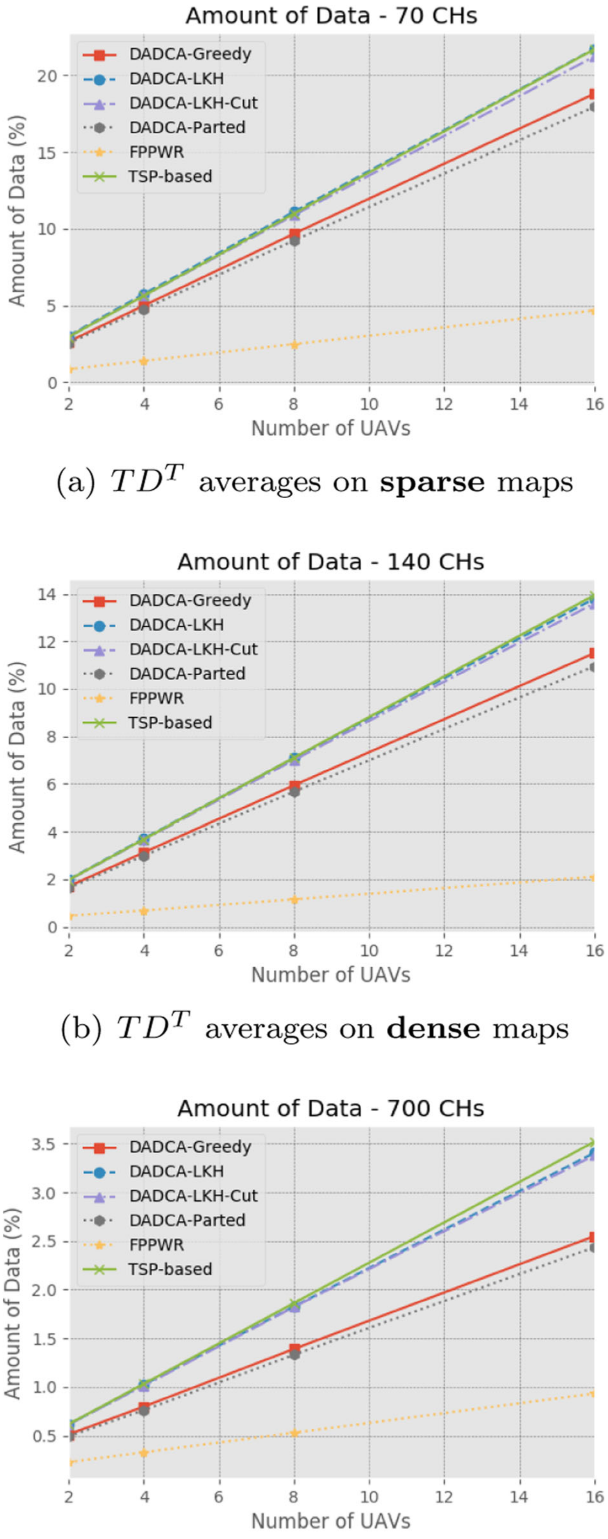


(a) TD^T on **sparse** maps

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

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

Fig. 20 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



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

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

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

Fig. 21 TD^T averages with all results. **a** TD^T averages on **sparse** maps (**b**) TD^T averages on **dense** maps (**c**) TD^T averages on ***full*** maps

6.2 Messages delay - D_n^T

In this section, we explore the D_n^T results (i.e., the mean delay of all message collected from each CH, as formalized in Section 3.3) (5). The delays analyzed are sub-products of the experiment discussed in the previous Section (6.1). After presenting the results, we analyze the effects of varying the number the UAVs on each of the strategies.

6.2.1 *Analysis of D_n^T*

Figure 22 presents all the 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 data set. The horizontal axis represents the number of UAVs, while the vertical axis represents D_n^T; smaller values on the vertical axis indicate better results.

Due to the tour length being larger in denser scenarios, 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, it does produce satisfactory results in all scenarios.

Figure 23 contain the chart of delays of 16 UAVs collecting data from 700 CHs. The best results for D_n^T are obtained using DADCA-LKH-Cut and DADCA-Parted. Without loss of generality, we 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 is significantly lowered.

The chart in Fig. 23 clearly displays the advantages of DADCA-LKH-Cut and DADCA-Parted over the other strategies. DADCA-LKH-Cut had 32% shorter delays and DADCA-Parted had 20% shorter delays than the TSP-based strategy when comparing the medians and the maximum values.

DADCA presents superior delays because the strategies that follow a Hamiltonian cycle have constant delays (FPPWR and any TSP-based strategy), 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.

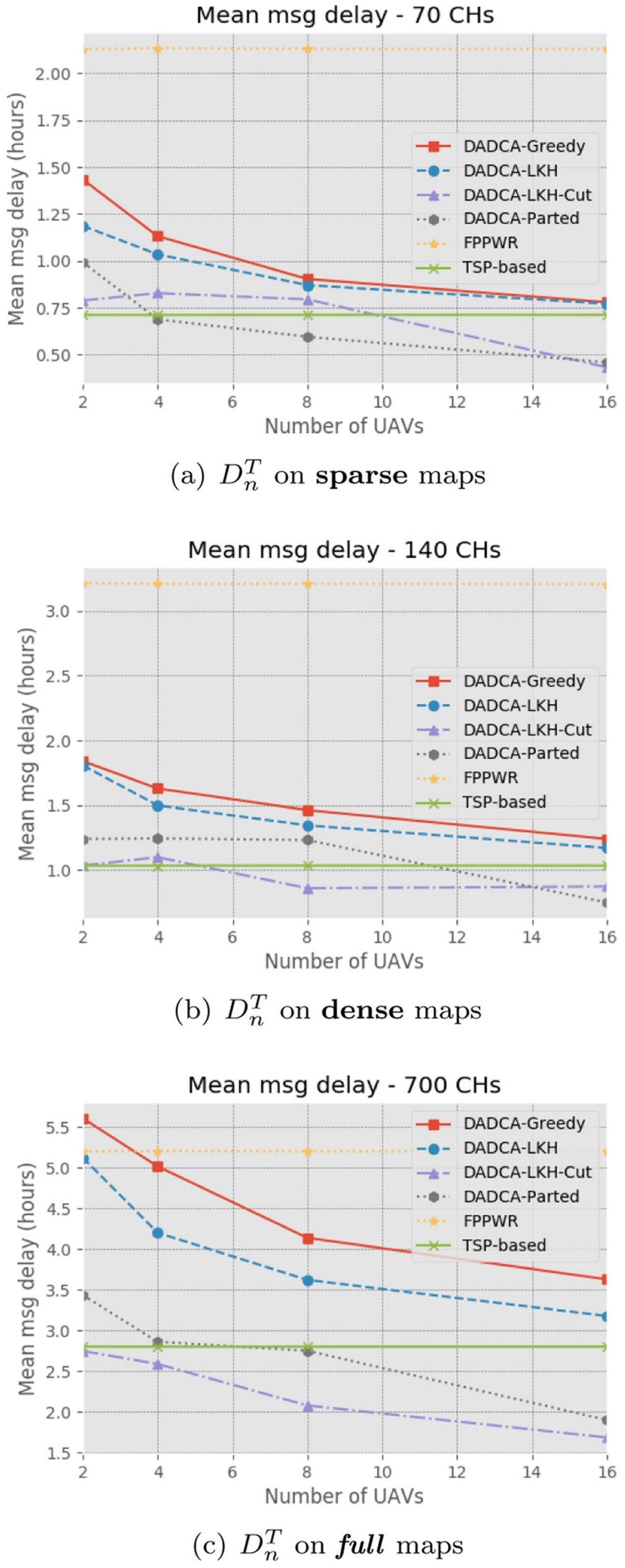
6.2.2 *Experiment summary*

With a small number of UAVs, the DADCA variations present similar or worse results than the TSP-based strategy, and this inefficiency occurs due to the path overlapping when two UAVs are flying together, which explains the limitations of DADCA.

However, when more UAVs are used, the performance of the DADCA variations significantly improves. 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. These points are presented in Fig. 22.

6.3 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 $\mathcal{O}(n^{2.2})$, and the complexity of FPPWR is $\mathcal{O}(n \log n)$; the TSP-based strategy is an NP-hard problem.



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

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

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

Fig. 22 D_n^T - average delays of each strategy for each density and distinc number of UAVs. **a** D_n^T on **sparse** maps (**b**)D_n^T on **dense** maps (**c**)D_n^T on ***full*** maps

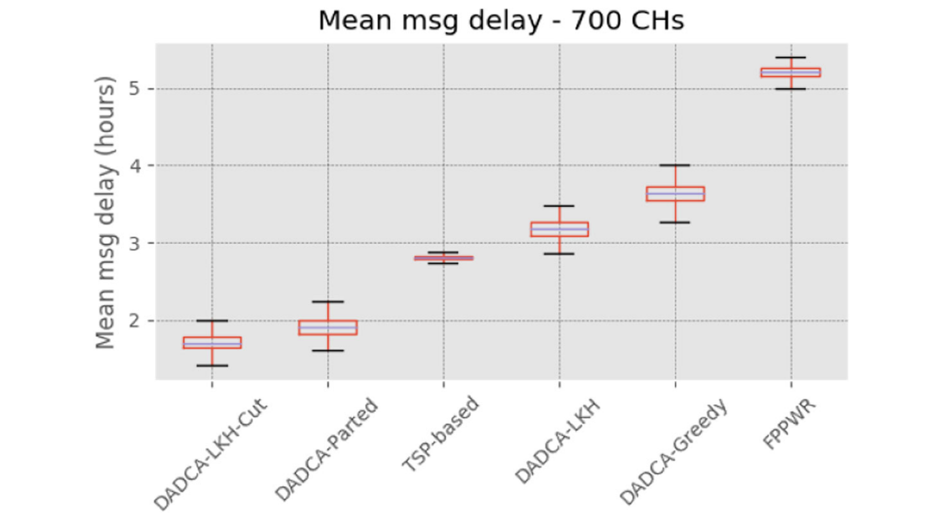


Fig. 23 D_n^T - delays of each strategy in *full* density and 16 UAVs as in Fig. 22c

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. Thus, 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 the flight controllers.

Unmanned aerial vehicles generally rely on only flight controllers with processors such as the ARM Cortex-M (with 32bits and clocks of 200MHz). Typically, a UAV's flight controllers rely on processing systems that are light and spend as little energy as possible. Moreover, due to security issues, it is not recommended to run arbitrary applications on the flight controllers. Similar technical specifications are found in the commonly used PixHawk flight controllers family[2].

To run application-level programs in UAVs, it is generally necessary to install a dedicated processing unit for the application. The most-used configuration is to adopt single-board computers, which naturally comply with weight and energy-consumption requirements. Indeed, the frameworks for platforms such as FlytOS [28] and Dronecode [76] are based on this architecture.

The cited projects [28, 76] rely on single-board main computers, such as RaspBerry Pi, BeagleBone, and Intel Edison. To evaluate the proposed strategies, we selected the RaspBerry Pi 3 as the hardware given its high availability in the market.

6.3.1 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 1.2 GHz, with a GPU core that runs at 400 MHz. The data set and densities used are the same as in the previous experiments.

[2] https://pixhawk.org/

6.3.2 Analysis tour planning time

Figure 24 presents the average results of each strategy's tour processing time. The horizontal axis represents the maps, while the vertical axis represents the processing time; smaller values along the vertical axis are better.

The first result noticed is that the TSP-based strategy demands significantly more time than any other strategy. In fact, the results of the TSP-based strategy create difficulty in analyzing the other strategies in the same chart because TSP-based results mislead the scale of results. Accordingly, in Fig. 24, the same results are grouped together, and the average is presented. It becomes clearer that TSP-based results have a complete other scale.

To provide a clearer analysis, we present only the *full* series in the charts in Figs. 25, 26, and 27, which do not contain outliers and are presented as boxplots. In Fig. 25, all the results are presented. The outliers were removed because some results of the TSP-based strategy differed largely from the average results. In all charts, the vertical axis represents the processing time, in which lower values along the vertical axis are faster.

In Fig. 26, the three strategies that yielded 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 require more than 29 minutes of computing time. In addition, at least 75% of the TSP-based strategy's tours require more computing time than those of the other strategies. While DADCA-LKH and DADCA-LKH-Cut produce better results than the TSP-based strategy, the results of both are still far worse than the others.

In Fig. 27, we present the three best results of the FPPWR, DADCA-Naive, and DADCA-Parted strategies. DADCA-Parted produced the best results in this experiment, computing its tours in less time than the others.

DADCA-Parted achieved 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.

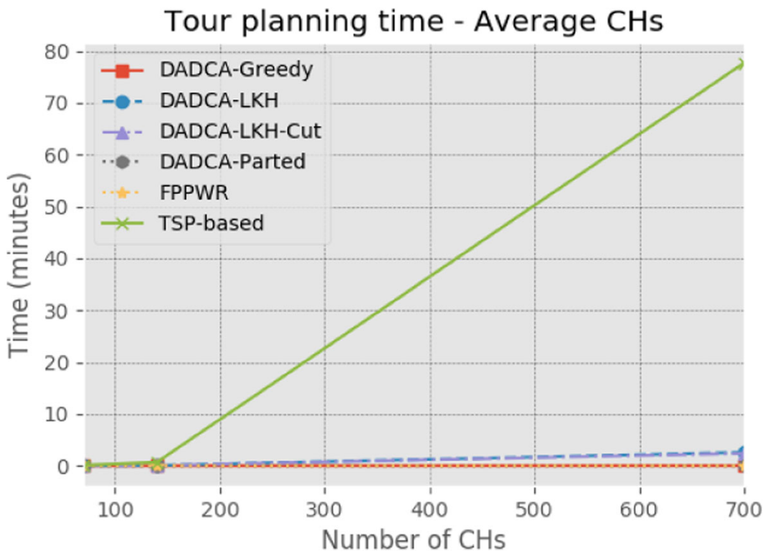


Fig. 24 Average processing time in diverse distributions

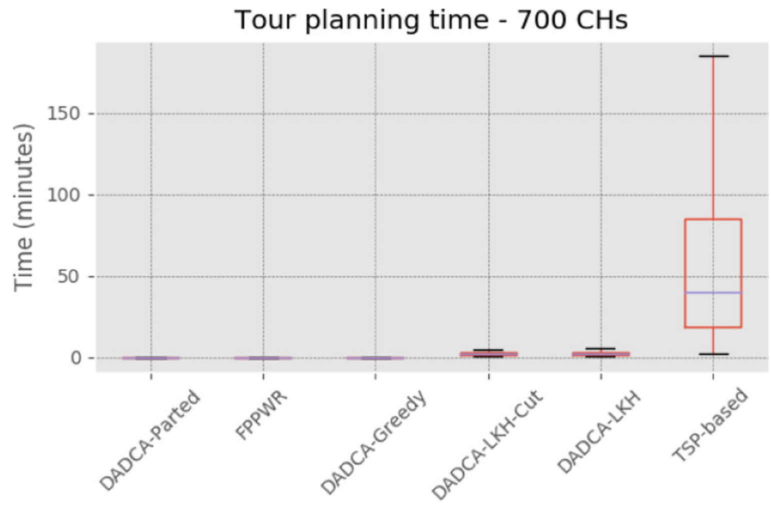


Fig. 25 Processing planning of ***full*** distribuions in a boxplot view

Despite the FPPWR results, the overall results reasonably could have been expected because they are directly proportional to the efforts of each strategy for computing its tour. Optimization heuristics are expected to be more difficult to solve, followed by polynomial heuristics that deal 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 1,000 CHs.

6.3.3 Experiment limitations

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

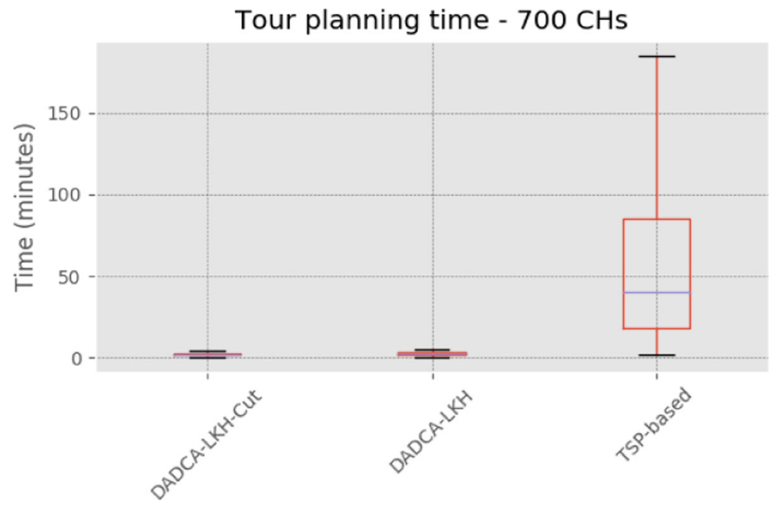


Fig. 26 Presenting selected worst results. Processing planning of ***full*** distribuions in a boxplot view

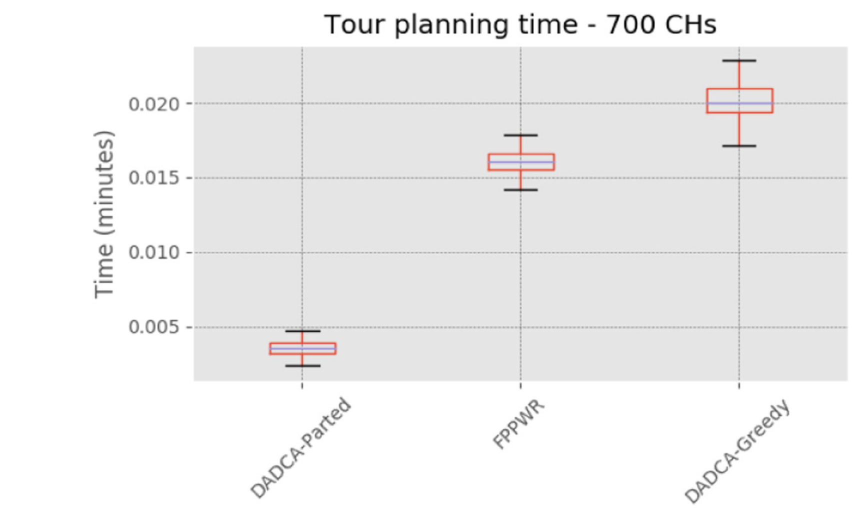


Fig. 27 Presenting selected best results. Processing planning of ***full*** distribuions in a boxplot view

Memory In the preliminary tests involving up to 1,000 CHs, none of the strategies reached 128 Mb of memory utilization. As the evaluated hardware has 1 Gb 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 the 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, indicating that the amount of collected data TD^T and delay D_n^T were evaluated as though the processing time of each strategy were zero. It is reasonable to expect that the TSP-based strategy could produce significantly different results if the UAVs waited to begin their data collection until after the TSP-based strategy computed the tour.

6.4 Energy constraints

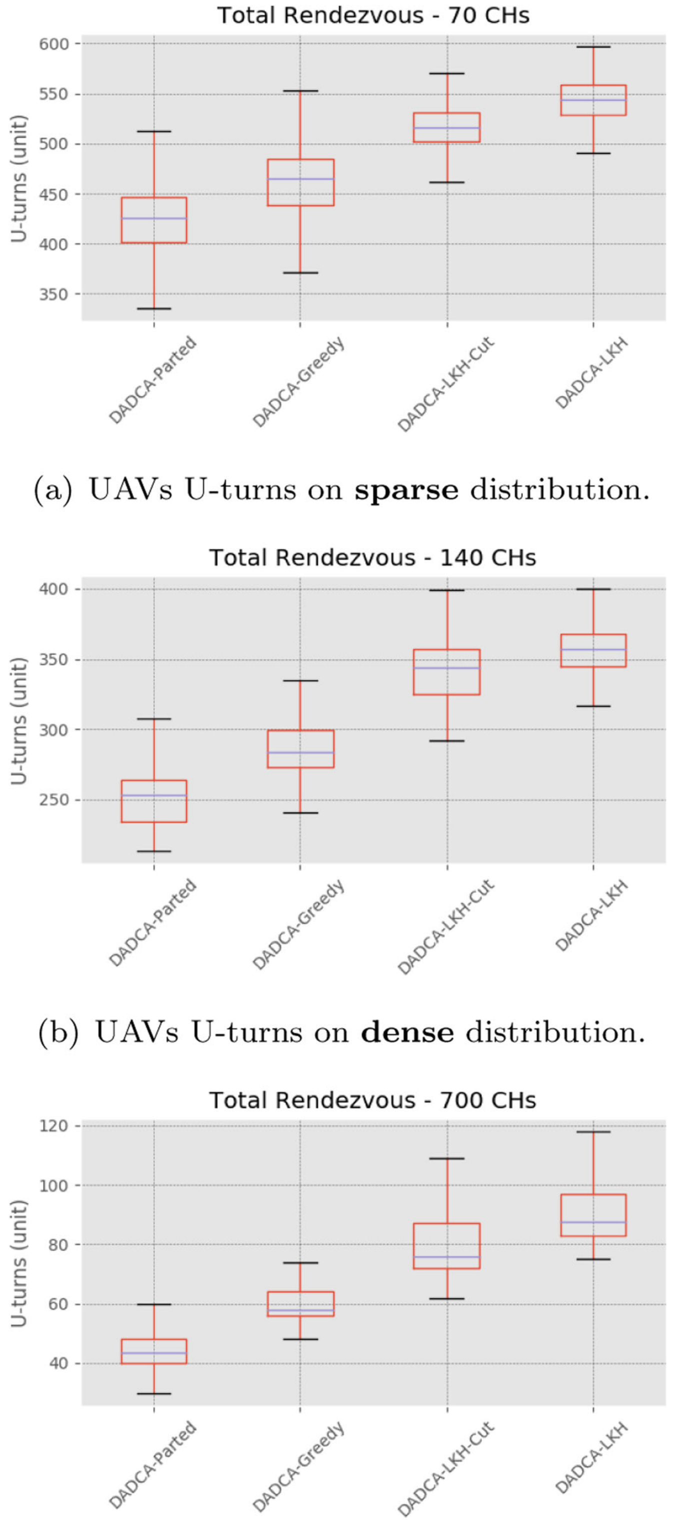
In this section, we introduce two energy issues pertaining to UAVs. Section 6.4.1 explores the UAVs' inversion of flight direction (U-turn) after a valid rendezvous. The tour planning aboard UAVs may consume significant energy and is explored in Section 6.4.2.

6.4.1 DADCA rendezvous energy impacts

As previously mentioned, 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 the UAVs is to change direction by a 180°angle (i.e., perform a U-turn). It is reasonable to expect that this behavior would consume energy. Such energy consumption could result in a need for more frequent battery recharges, and more frequent visits to the GS could increase the SE.

Figure 28 consists of three charts containing the number of U-turns performed by the UAVs during the experiments. The number of UAVs considered in this Figure is 16 UAVs. It can be noted that the DADCA-Naive and DADCA-Parted tours include fewer U-turns

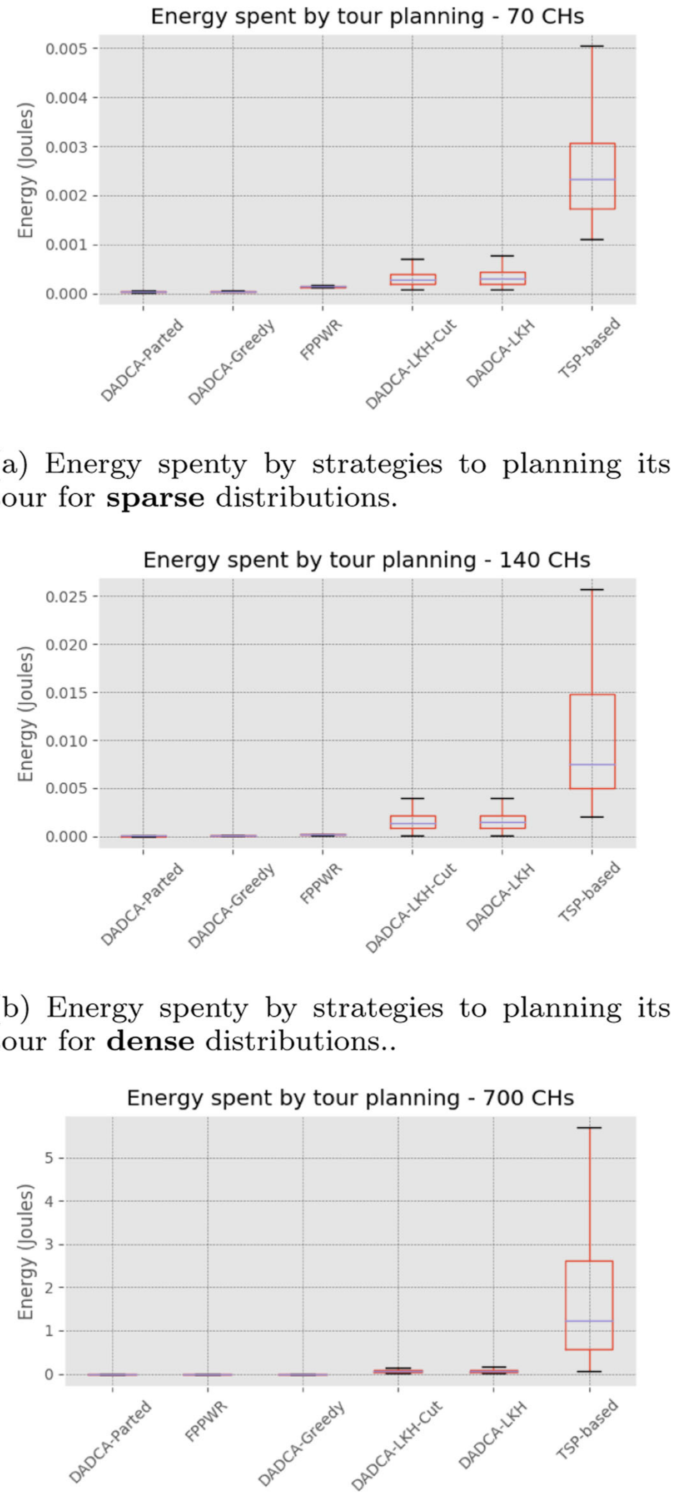


(a) UAVs U-turns on **sparse** distribution.

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

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

Fig. 28 Number of UAVs U-turns during the period of T. **a** UAVs U-turns on **sparse** distribution. **b** UAVs U-turns on **dense** distribution. **c** UAVs U-turns on ***full*** distribution



(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.

Fig. 29 Amount of energy spent by each strategy for each distribution. **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

than the 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 do the other strategies because the same number of UAVs collect 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.

6.4.2 Tour planning energy consumption

As proposed, the tour shall be computed using each UAV. In Section 6.3, the processing time of each strategy is presented, with some results taking hours to be processed. In this experiment, we measured the energy consumed by the hardware discussed in Section 6.3 to provide an estimate of the effects that tour planning could have on the energy consumption of the UAVs. To do so, we logged the energy consumption of the Raspberry Pi 3 for one hour under full-time processing of the tour.

The measurement was taken using an inline ammeter during the hour. The ammeter used was a Keweisi model KWS-V20, with an accuracy of 0.4% in current measurements. In idle processing, the Rapberry Pi spent 75 mAh with 5.23 volts per hour. In tour processing tasks, the measurement experiment resulted in 362 mAh with 5.13 volts per hour. This result was divided by 3,600 seconds to obtain 0.1 mAh with 5.13 volts per second. The results of the energy consumption were used in conjunction with the results from Section 6.3 to present the energy necessary for each strategy to compute a tour, as demonstrated in Fig. 29.

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 deriving the energy value requires the voltage, as well. We present the charts in joules, in which smaller values along the vertical axis are better. The charts in Fig. 29 display the straight application of the tour planning time presented in Section 6.3. Figure 29 presents the TSP-based strategy as the worst strategy, spending disproportionate energy with respect to the other tested strategies. DADCA-Parted yielded the best results, followed by DADCA-Greedy.

7 Conclusions

Typically, research on WSN data collection with UAVs focuses on the use of only a single UAV. Through a brief literature review, this work illustrates the lack of research on collaboration among UAVs in data collection tasks.

Given this context, it was proposed that distributed approaches can be equally or even more efficient than centralized approaches in terms of particular issues in UAV data collection. Accordingly, a distributed approach—DADCA—was proposed and evaluated against other strategies (FPPWR and TSP-based approaches).

In addition, we evaluated the performance of DADCA variations against conventional strategies with optimized tour approaches. The results of the simulations indicated better results than those obtained using other strategies.

Abbreviations

D_n^T: Mean message delay time of received data in GS during the period T; S_{right}: Right side of K of a given UAV; ρ: Radio transmittion rate; Ad hoc: A decentralised type of wireless network; CH: Cluster Head; FOV: Field of view; GS: Ground Station; LKH: Lin-Kernighan-Helsgaun Heuristic; MTBF: Mean Time Between Failures; R: Radio range: ROS: Robot

Operating System; SE: DADCA's Shadow Effect; T: Period of data collection operation; TSPN: Variation of TSP with neighbourhoods; V2I: Communication Vehicle to Infrastructure; W: Period that a UAV needs to cover the entire tour K once; WSN: Wireless Sensors Network: p: Number of Cluster Heads; u: Number of UAVs; S_{left}: Letf side of K of a given UAV; TD^{T}: Tax of collected data (collected divided by total generated); *b*: Theoretical intersection between $S_{left} \cap S_{right}$; B: UAV flight time, as known as flight endurance; DADCA: Distributed Algorithm for Aerial Data Collection; FT: UAV Fail Tax; K: Original tour length; LSN: Linear sensors network; mTSP: Variation of TSP with multiple salesmen; SNIR: Signal-to-interference-plus-noise ratio; TSP: Travelling Salesman Problem; UAV: Unmanned Aerial Vehicle; V2V: Communication Vehicle to Vehicle; WM: Wasted moviment

Acknowledgments
Not applicable.

Authors' contributions
BJOS is the main contributor and writer of this work, undertaken as part of his Ph.D. studies. ME have contributed to the conception of this study. Both authors have read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 17 May 2019 Accepted: 2 July 2020
Published online: 21 July 2020

References

1. Akber SMA, Khan IA, Muhammad SS, Mohsin SM, Khan IA, Shamshirband S, Chronopoulos AT. 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: ACM Press; 2018. p. 199–207. https://doi.org/10.1145/3216122.3216166.
2. Liu S, Wei Z, Guo Z, Yuan X, Feng Z. Performance Analysis of UAVs Assisted Data Collection in Wireless Sensor Network. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring). IEEE; 2018. p. 1–5. https://doi.org/10.1109/VTCSpring.2018.8417673.
3. Huang H, Savkin AV, Ding M, Huang C. Mobile robots in wireless sensor networks: A survey on tasks. Comput Netw. 2019;148:1–19. https://doi.org/10.1016/J.COMNET.2018.10.018.
4. Grigulo J, Becker LB. 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; 2018. p. 808–15. https://doi.org/10.1109/ETFA.2018.8502536.
5. Garraffa M, Bekhti M, Letocart L, Achir N, Boussetta K. Drones path planning for WSN data gathering: A column generation heuristic approach. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE; 2018. p. 1–6. https://doi.org/10.1109/WCNC.2018.8377391.
6. Sayeed M, Kumar R, Sayeed MA, Kumar R. An Efficient Mobility Model for Improving Transmissions in Multi-UAVs Enabled WSNs. Drones. 2018;2(3):31. https://doi.org/10.3390/drones2030031.
7. Wu J. A multi-tiered network with aerial and ground coverage. Comput Commun. 2018;131:39–42. https://doi.org/10.1016/J.COMCOM.2018.07.007.
8. Ruan L, Wang J, Chen J, Xu Y, Yang Y, Jiang H, Zhang Y, Xu Y. Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework. China Commun. 2018;15(10):194–209. https://doi.org/10.1109/CC.2018.8485481.
9. Olivieri B, Endler M. An Algorithm for Aerial Data Collection from Wireless Sensors Networks by Groups of UAVs. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2017. https://doi.org/10.1109/IROS.2017.8202262.
10. Olivieri B, Endler M. DADCA: An Efficient Distributed Algorithm for Aerial DataCollection from Wireless Sensors Networks by UAVs; 2017. p. 129–36. https://doi.org/10.1145/3127540.3127553.
11. Reina DG, Tawfik H, Toral SL. Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks. Ad Hoc Netw. 2018;68(1):16–32. https://doi.org/10.1016/J.ADHOC.2017.09.005.
12. Al-Hourani A, Gomez K. Modeling Cellular-to-UAV Path-Loss for Suburban Environments. IEEE Wirel Commun Lett. 2018;7(1):82–5. https://doi.org/10.1109/LWC.2017.2755643.
13. Xu J, Zeng Y, Zhang R. UAV-Enabled Wireless Power Transfer: Trajectory Design and Energy Optimization. Trans Wirel Commun IEEE. 20181–1. https://doi.org/10.1109/TWC.2018.2838134.
14. Jeong S, Simeone O, Kang J. Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning. IEEE Trans Veh Technol. 2018;67(3):2049–63. https://doi.org/10.1109/TVT.2017.2706308.
15. Qadori HQ, Zulkarnain ZA, Hanapi ZM, Subramaniam S. Multi-mobile agent itinerary planning algorithms for data gathering in wireless sensor networks: A review paper. Int J Distrib Sensor Netw. 2017;13(1):155014771668484. https://doi.org/10.1177/1550147716684841.
16. Secinti G, Darian PB, Canberk B, Chowdhury KR. SDNs in the Sky: Robust End-to-End Connectivity for Aerial Vehicular Networks. IEEE Commun Mag. 2018;56(1):16–21. https://doi.org/10.1109/MCOM.2017.1700456.
17. Kerrache CA, Lakas A, Lagraa N, Barka E. UAV-assisted technique for the detection of malicious and selfish nodes in VANETs. Veh Commun. 2018;11:1–11. https://doi.org/10.1016/J.VEHCOM.2017.12.001.

18. Alemayehu TS, Kim J-H. Efficient Nearest Neighbor Heuristic TSP Algorithms for Reducing Data Acquisition Latency of UAV Relay WSN. Wirel Pers Commun. 2017;1(1):1–15. https://doi.org/10.1007/s11277-017-3994-9.
19. Rashed S, Soyturk M. Analyzing the Effects of UAV Mobility Patterns on Data Collection in Wireless Sensor Networks. Sensors. 2017;17(2):413. https://doi.org/10.3390/s17020413.
20. Messous M-A, Senouci S-M, Sedjelmaci H. Network connectivity and area coverage for UAV fleet mobility model with energy constraint. In: 2016 IEEE Wireless Communications and Networking Conference. IEEE; 2016. p. 1–6. https://doi.org/10.1109/WCNC.2016.7565125.
21. Xue Z, Wang J, Shi Q, Ding G, Wu Q. Time-Frequency Scheduling and Power Optimization for Reliable Multiple UAV Communications. IEEE Access. 2018;6:3992–4005. https://doi.org/10.1109/ACCESS.2018.2790933.
22. Mazayev A, Correia N, Schütz G. Data Gathering in Wireless Sensor Networks Using Unmanned Aerial Vehicles. Int J Wirel Inf Networks. 2016;23(4):297–309. https://doi.org/10.1007/s10776-016-0319-y.
23. Ho D-T, Grøtli EI, Sujit PB, Johansen TA, Sousa JB. Optimization of Wireless Sensor Network and UAV Data Acquisition. J Intell Robot Syst. 2015;78(1):159–79. https://doi.org/10.1007/s10846-015-0175-5.
24. Martinez-de Dios JR, Lferd K, de San Bernabé A, Núñez G, Torres-González A, Ollero A. Cooperation Between UAS and Wireless Sensor Networks for Efficient Data Collection in Large Environments. J Intell Robot Syst. 2013;70(1-4): 491–508. https://doi.org/10.1007/s10846-012-9733-2.
25. Wang J, Cao J, Ji S, Park JH. Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks. J Supercomput. 2017;1(1):1–14. https://doi.org/10.1007/s11227-016-1947-9.
26. Wang C, Ma F, Yan J, De D, Das SK. Efficient Aerial Data Collection with UAV in Large-Scale Wireless Sensor Networks. Int J Distrib Sensor Netw. 2015;2015(11):1–19. https://doi.org/10.1155/2015/286080.
27. Lamping AP, Ouwerkerk JN, Cohen K. Multi-UAV Control and Supervision with ROS. In: Multi-UAV Control and Supervision with ROS; 2018. https://doi.org/10.2514/6.2018-4245.
28. FlytBase. FlytBase: Build Smart and Scalable Drone Applications. https://flytbase.com/?cn-reloaded=1. Accessed 23 July 2018.
29. Erle Robotics. Erle Robotics | Intelligent Robotics. https://erlerobotics.com/blog/. Accessed 23 July 2018.
30. System RO. ROS.org | Powering the world's robots. 2018. http://www.ros.org/. Accessed 23 July 2018.
31. Sharma V, You I, Kumar R. Energy Efficient Data Dissemination in Multi-UAV Coordinated Wireless Sensor Networks. Mob Inf Syst. 2016;2016:1–13. https://doi.org/10.1155/2016/8475820.
32. Pascarella D, Venticinque S, Aversa R. Agent-based design for UAV mission planning. In: Proceedings - 2013 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2013. Compiegne: IEEE; 2013. p. 76–83. https://doi.org/10.1109/3PGCIC.2013.18.
33. Thakur D, Likhachev M, Keller J, Kumar V, Dobrokhodov V, Jones K, Wurz J, Kaminer I. Planning for opportunistic surveillance with multiple robots. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver: IEEE; 2013. p. 5750–7. https://doi.org/10.1109/IROS.2013.6697189.
34. Wu Q, Sun P, Boukerche A. 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: ACM Press; 2018. p. 34–40. https://doi.org/10.1145/3272036.3272047.
35. Ladosz P, Oh H, Chen W-H. Trajectory Planning for Communication Relay Unmanned Aerial Vehicles in Urban Dynamic Environments. J Intell Robot Syst. 2017;1(1):1–19. https://doi.org/10.1007/s10846-017-0484-y.
36. Wang W, Shi H, Huang P, Wu F, Fang D, Chen X, Yin X. 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: ACM Press; 2015. p. 47–52. https://doi.org/10.1145/2820716.2820723.
37. Fu Y, Ding M, Zhou C, Hu H. Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-Behaved Particle Swarm Optimization. IEEE Trans Syst Man Cybern Syst. 2013;43(6):1451–65. https://doi.org/10.1109/TSMC.2013.2248146.
38. Kashuba SV, Novikov VI, Lysenko OI, Alekseeva IV. Optimization of UAV path for wireless sensor network data gathering. In: 2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD). Kyiv: IEEE; 2015. p. 280–3. https://doi.org/10.1109/APUAVD.2015.7346621.
39. Google. Google Project Loon. 2018. https://www.google.com/loon. Accessed 19 July 2018.
40. Facebook. Facebook Aquila. https://web.facebook.com/notes/mark-zuckerberg/the-technology-behind-aquila/10153916136506634/?_rdc=1&_rdr. Accessed 19 July 2018.
41. Azari MM, Rosas F, Chen K-C, Pollin S. Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity. IEEE Trans Commun. 2018;66(1):330–344. https://doi.org/10.1109/TCOMM.2017.2746105.
42. Wu Q, Zeng Y, Zhang R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. IEEE Trans Wirel Commun. 2018;17(3):2109–21. https://doi.org/10.1109/TWC.2017.2789293.
43. Ebrahimi D, Sharafeddine S, Ho P-H, Assi C. UAV-Aided Projection-Based Compressive Data Gathering in Wireless Sensor Networks. Internet Things J IEEE. 20181–1. https://doi.org/10.1109/JIOT.2018.2878834.
44. Fu S, Zhao L, Su Z, Jian X, Fu S, Zhao L, Su Z, Jian X. UAV Based Relay for Wireless Sensor Networks in 5G Systems. Sensors. 2018;18(8):2413. https://doi.org/10.3390/s18082413.
45. Sekander S, Tabassum H, Hossain E. Multi-Tier Drone Architecture for 5G/B5G Cellular Networks: Challenges, Trends, and Prospects. IEEE Commun Mag. 2018;56(3):96–103. https://doi.org/10.1109/MCOM.2018.1700666.
46. Thammawichai M, Baliyarasimhuni SP, Kerrigan EC, Sousa JB. Optimizing Communication and Computation for Multi-UAV Information Gathering Applications. IEEE Trans Aerosp Electron Syst. 2018;54(2):601–15. https://doi.org/10.1109/TAES.2017.2761139.
47. Yanmaz E, Yahyanejad S, Rinner B, Hellwagner H, Bettstetter C. Drone networks: Communications, coordination, and sensing. Ad Hoc Netw. 2018;68:1–15. https://doi.org/10.1016/J.ADHOC.2017.09.001.
48. Gupta L, Jain R, Vaszkun G. Survey of Important Issues in UAV Communication Networks. IEEE Commun Surv Tutorials. 2016;18(2):1123–52. https://doi.org/10.1109/COMST.2015.2495297.
49. Cao H, Liu Y, Yue X, Zhu W, Cao H, Liu Y, Yue X, Zhu W. Cloud-Assisted UAV Data Collection for Multiple Emerging Events in Distributed WSNs. Sensors. 2017;17(8):1818. https://doi.org/10.3390/s17081818.

50. Khan MA, Safi A, Qureshi IM, Khan IU. 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. https://doi.org/10.1109/INTELLECT.2017.8277614.
51. Ma X, Chisiu S, Kacimi R, Dhaou R. Opportunistic communications in WSN using UAV. In: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE; 2017. p. 510–5. https://doi.org/10.1109/CCNC.2017.7983160.
52. Say S, Inata H, Ernawan ME, Pan Z, Liu J, Shimamoto S. Partnership and data forwarding model for data acquisition in UAV-aided sensor networks. In: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE; 2017. p. 933–8. https://doi.org/10.1109/CCNC.2017.7983258.
53. Mori S. Cooperative sensing data collecting framework by using unmanned aircraft vehicle in wireless sensor network. In: 2016 IEEE International Conference on Communications (ICC). Kuala Lumpur: IEEE; 2016. p. 1–6. https://doi.org/10.1109/ICC.2016.7511187.
54. Ben Ghorbel M, Rodriguez-Duarte D, Ghazzai H, Hossain MJ, Menouar H. Energy Efficient Data Collection for Wireless Sensors Using Drones. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring). IEEE; 2018. p. 1–5. https://doi.org/10.1109/VTCSpring.2018.8417798.
55. Hua M, Wang Y, Zhang Z, Li C, Huang Y, Yang L. Power-Efficient Communication in UAV-Aided Wireless Sensor Networks. IEEE Commun Lett. 2018;22(6):1264–7. https://doi.org/10.1109/LCOMM.2018.2822700.
56. Zeng Y, Zhang R. Energy-Efficient UAV Communication With Trajectory Optimization. IEEE Trans Wirel Commun. 2017;16(6):3747–60. https://doi.org/10.1109/TWC.2017.2688328.
57. Berrahal S, Kim J-H, Rekhis S, Boudriga N, Wilkins D, Acevedo J. Unmanned aircraft vehicle assisted WSN-based border surveillance. In: 2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE; 2015. p. 132–7. https://doi.org/10.1109/SOFTCOM.2015.7314129.
58. Berrahal S, Kim J-H, Slim R, Wilkins D, Acevedo J. Border surveillance monitoring using Quadcopter UAV-Aided Wireless Sensor N...: Discovery Service para a PUC-Rio. J Commun Softw Syst. 2016;12(1):67–82.
59. Jawhar I, Mohamed N, Al-Jaroodi J, Zhang S, Jawhar I, Mohamed N, Al-Jaroodi J, Zhang S. A Framework for Using Unmanned Aerial Vehicles for Data Collection in Linear Wireless Sensor Networks. J Intell Robot Syst. 2014;74: 437–453. https://doi.org/10.1007/s10846-013-9965-9.
60. Nikhitha SR, Panda M. Optimal Sensor Data Harvesting Using A Mobile Sink. Procedia Comput Sci. 2018;143: 921–930. https://doi.org/10.1016/J.PROCS.2018.10.343.
61. Sun P, Boukerche A. Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network. Comput Netw. 2018;146:217–231. https://doi.org/10.1016/J.COMNET.2018.09.022.
62. Wiki. Hurricane deads. https://en.wikipedia.org/wiki/List_of_deadliest_Atlantic_hurricanes. Accessed 19 July 2018.
63. Ritz R, D'Andrea R. Carrying a flexible payload with multiple flying vehicles. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2013. https://doi.org/10.1109/iros.2013.6696850.
64. Burman J, Hespanha J, Madhow U, Isaacs J, Venkateswaran S, Pham T. Autonomous UAV persistent surveillance using bio-inspired strategies. In: SPIE Defense, Security, and Sensing. Baltimore: International Society for Optics and Photonics; 2012. p. 13. https://doi.org/10.1117/12.920018.
65. Ho D-T, Grotli EI, Sujit PB, Johansen TA, De Sousa JB. 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: IEEE; 2013. p. 1403–8. https://doi.org/10.1109/GLOCOMW.2013.6825191.
66. Kingston D, Beard RW, Holt RS. Decentralized Perimeter Surveillance Using a Team of UAVs. IEEE Trans Robot. 2008;24(6):1394–404. https://doi.org/10.1109/TRO.2008.2007935.
67. Helsgaun K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. Eur J Oper Res. 2000;126(1):106–30. https://doi.org/10.1016/S0377-2217(99)00284-2.
68. Zurich D. Sinalgo-simulator for network algorithms. 2013. http://www.dcg.ethz.ch/projects/sinalgo/. Accessed 10 Dec 2016.
69. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9(3):90–5. https://doi.org/10.1109/MCSE.2007.55.
70. Zurich E. 81 hours fly. http://www.atlantiksolar.ethz.ch/?p=670. Accessed 07 July 2018.
71. Oettershagen P, Melzer A, Mantel T, Rudin K, Lotz R, Siebenmann D, Leutenegger S, Alexis K, Siegwart R. A Solar-Powered Hand-Launchable UAV for Low-Altitude Multi-Day Continuous Flight. In: IEEE Robotics and Automation Society's - ICRA. Seattle; 2015. https://doi.org/10.1109/ICRA.2015.7139756.
72. In.c D. Xbee S1. https://www.digi.com/resources/documentation/digidocs/pdfs/90000982.pdf. Accessed 07 July 2018.
73. Cook W. Concorde traveling salesman problem solver and libraries. http://www.math.uwaterloo.ca/tsp/concorde/. Accessed 23 July 2018.
74. Helsgaun K. An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems, Technical report. 2017. http://www.akira.ruc.dk/~keld/research/LKH/LKH-3_REPORT.pdf.
75. Silva C, Pimentel LCG, Landau L, Filho PH, Gobbo FR, de Sousa P. Supportive elements to the decision-making process in the emergency planning of the Angra dos Reis Nuclear Power Complex, Brazil. Environ Earth Sci. 2017;76(3):133. https://doi.org/10.1007/s12665-017-6450-y.
76. N. Intel, 3DR, Yunteec. Dronecode Frramework. https://www.dronecode.org/about/. Accessed 21 Nov 2018.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.