Efficient Data Collection and Tracking with Flying Drones

Christelle Caillouet, Frédéric Giroire, Tahiry Razafindralambo

Efficient Data Collection and Tracking with Flying Drones

Christelle Caillouet[a,*], Frédéric Giroire[a], Tahiry Razafindralambo[b]

[a] *Université Côte d'Azur, CNRS, Inria, I3S, France*
[b] *Université La Réunion, LIM, France*

Abstract

Data collection is an important mechanism for wireless sensor networks to be viable. This paper addresses the Aerial Data Collection Problem (ADCP) from a set of mobile wireless sensors located on the ground, using a fleet of flying devices. The objective is i) to deploy a set of Unmanned Aerial Vehicles (UAVs) in a 3D space to cover and collect data from all the mobile wireless sensors at each time step through a ground-to-air communication, ii) to send these data to a central base station using multi-hop wireless air-to-air communications through the network of UAVs, iii) while minimizing the total deployment cost (communication and deployment) over time. The Aerial Data Collection Problem (ADCP) is a complex time and space coverage, and connectivity problem. We first present a mixed-integer linear program solving ADCP optimally for small instances. Then, we develop a second model solved by column generation for larger instances, with optimal or heuristic pricing programs. Results show that our approach provides very accurate solutions minimizing the data collection cost. Moreover, only a very small number of columns are generated throughout the resolution process, showing the efficiency of our approach.

Keywords: UAV, Coverage, Wireless Sensor Network

1. Introduction

Wireless sensor networks (WSNs) have tremendous applications in environmental monitoring or surveillance. Within the context of the Internet of mobile things, this paper focuses on observing a set of mobile targets, i.e., sensors, using a fleet of Unmanned Aerial Vehicles (UAVs), i.e., flying drones. Specific applications such as wildlife monitoring [1], vehicle observation and tracking [2] in smart cities and monitoring sporting events [3, 4] can benefit from the results presented in this paper. In our context, the flying drones act as a wireless mobile backbone, covering the targets on the ground and collecting information from them. Gathered data is continuously sent to a base station for further processing using a multi-hop communication paradigm through the wireless backbone. Our problem can be described as an optimization of spacial and temporal coverage with mobile flying drones and is named the Aerial Data Collection Problem (ADCP).

The use of mobile flying devices to cover mobile ground targets has become an important topic in the past few years. The problem tackled in this paper is crucial since it allows the monitoring system to optimally adapt to the evolving space and time placement of the ground targets, but also to the changing requirements of the application itself. Therefore, deployment efficiency and cost can be jointly optimized for a wide adoption of wireless mobile monitoring systems. Figure 1 shows the studied multi-tier network architecture composed of mobile sensors, flying drones, and a fixed base station. The sensors are located on the ground and produce data. Their mobility pattern is usually unknown. Data is then gathered by a fleet of UAVs whose mobility is fully controlled in order to track the sensors. Each ground sensor must be covered at any time by at least one UAV for ground-to-air communications. Each UAV moves in a 3D-space and their altitude must be managed for the coverage and data collection. When a UAV is at a high altitude, the covered area is wider on the ground. However, altitude increases the distance between UAVs and ground sensors and thus reduces the ground-to-air communication transmission quality [5]. Finally, data gathered by UAVs are sent to a base station in a multi-hop fashion. A connected backbone of UAVs

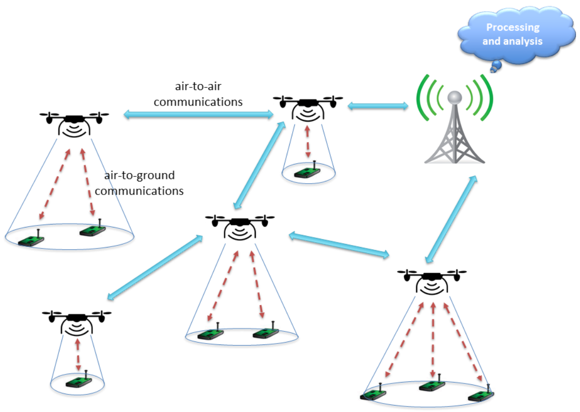


Figure 1: Multi-tier network architecture for the aerial data collection problem (ADCP).

*Corresponding author.
Email addresses: christelle.caillouet@unice.fr (Christelle Caillouet), frederic.giroire@cnrs.fr (Frédéric Giroire), tahiry.razafindralambo@univ-reunion.fr (Tahiry Razafindralambo)

and the base station must be formed and maintained at anytime to transmit the data.

Deployment of UAVs is studied both from theoretical and practical considerations. From a practical point of view, research focuses on distributed UAV deployment, especially the optimal way of moving one UAV from one point to another. The robotics community is strongly involved in this research topic. From theoretical and practical points of view, the networking community and especially the wireless sensor network community focus on designing the appropriate communication protocol for UAV mobile networks. From a theoretical point of view, research focuses on optimal or approximated computations of the positions of each UAV. Multiple works propose exact mathematical formulations or heuristic algorithms for the coverage and positioning problems using UAVs with several objective functions. The work presented in this paper lays in this last category. However, in this paper we try to encompass positioning aspects for coverage, communication aspects, especially connectivity, and constraints related to UAV movement. This paper tries to fill the gap between the networking approach and the theoretical approach by providing and describing a tool and framework for the evaluation and comparison of data collection and tracking strategies algorithms for wireless flying drones. Indeed, the results of this paper can be used and modified to serve as a basis for the networking community to evaluate the performance and enhancement an algorithm can bring.

This paper is organized as follows. In Section 2, we present an overview of related works and compare our paper to specific results in the literature. In Section 3, we describe the studied ADCP with its assumptions and the considered scenario. In Section 4, we present a mixed-integer linear program optimally solving ADCP. However, the model does not allow us to solve large instances of the problem. We thus propose a scalable decomposition model in Section 5. This new formulation can be solved efficiently using column generation. We present the results in Section 5.4 validating our model and studying the evolution of the UAV deployment over time. A heuristic enhancing the resolution time of the column generation process is presented and validated in Section 6. Section 7 discusses the model assumptions and possible applications, together with potential future work. We finally conclude this paper in Section 8.

2. State of the art

This section is dedicated to the position of our paper regarding different aspects of the literature: global fundamental approaches (collection or dissemination), types of mobility patterns (random, constrained, fully controllable), community specificities (robotics, networking, operational research), and finally specific assumptions (3D-positioning, coverage, data gathering, and objectives).

2.1. Global positioning of our contribution

Mobile nodes such as UAVs (Unmanned Aerial Vehicles) have been introduced into WSNs (Wireless Sensor Networks) for multiple purposes. There are two main approaches: 1) The first one is the up-link process in which the set of mobile nodes help the WSN in the data collection/gathering process from sensors to a central entity [12]. 2) The second approach is to help the down-link process in which the set of mobile nodes help a central entity, for example the base station, disseminate/deliver information, update software or recharge sensors' batteries [13] which implies visiting all the sensors for a given period of time or at a given time. In this paper, we focus on the data collection/gathering process and cover this aspect from different perspectives. One important aspect is the type of mobility.

The literature is very rich regarding mobility management. There are three main mobility categories including random mobility, partially controllable mobility and fully controllable mobility of mobile nodes. In this paragraph, we provide some examples of each category.

- Random mobility (not controllable). In this category, results cannot provide guarantees on data collection/gathering and solutions are often subject to long delays. Early papers such as the one by Shah *et al.* [14] and Hamida *et al.* [15] lie in this category and provide a first approach of data collection/gathering in wireless sensor networks using mobile nodes. Unlike the results from [14] and [15], we are interested in data gathering guarantees and optimality.

- Partially controllable mobility. Trajectories and/or stop positions are constrained. This case is often environment specific, for example bus trajectories in a city. In this case, data gathering/collection can suffer from high delays. The paper from Gandham et al. [16], from Basagni et al. [17] and from Luo et al. [18], were among the first to provide some solutions in this category. Unlike [16], [17] and [18], we want a data gathering process with minimal delay. Each sensor has to be permanently in contact with a mobile node in order to transmit the data directly after their production.

- Fully controllable mobility. It widens the application usage of wireless sensor networks. It is interesting to focus on fully controllable mobility since it is possible to compute the node's trajectory to optimize the collection process [19]. The contributions of this paper lie under this category. Unlike the paper from Wichmann *et al.* [19], we use multiple mobile nodes for data gathering. Unlike other works such as the one presented by Huang et al. [10], the data gathering process in this paper is time and space dependent since the data producers (sensors) are mobile and should therefore, be tracked and covered during the deployment procedure.

2.2. Community positioning of our contribution

The data collection process with mobile nodes or UAVs with fully controllable mobility have already been studied by different communities such as the wireless sensor network community, the robotics community and the operational research community. Each community has its own view and way to tackle the data gathering process, focusing on different issues,

Table 1: Contributions of our paper and related SOA.

Constraints/Objectives \ Papers	[6]	[7]	[8]	[9]	[10]	[11]	This work
Coverage of ground sensors	✓	✓	✓	✓	✓	✓	✓
3D-positioning of UAVs				✓		✓	✓
Mobile ground sensors						✓	✓
Connectivity btw UAVs		✓	✓				✓
Minimize number of UAVs				✓	✓	✓	✓
Minimize UAVs altitude							✓
Minimize remoteness of UAVs					✓		✓

and thus, putting more importance on some constraints, objectives or characteristics. i) The *wireless sensor network community* focuses on the communication protocol quality which may include connectivity of the network [20]. For example, given a location for each device, the objective is to implement protocols above the deployed network [21, 22]. In these papers, the main focus is to implement algorithms for data gathering. The deployment and its optimality are not considered. An interesting point from [20, 21, 22] is the fact that permanent connectivity in the UAV or mobile robots network is important and should be guaranteed. In our work, we also consider this point and add another constraint related to optimal coverage and position cost. ii) The *robotics community* mostly focuses on the robots' physical and mobility constraints such as turn curvature or obstacle avoidance [23], and collaboration among mobile devices for a specific task [7]. Moreover, it focuses on the deployment itself or how to drive each device to its specific location while maintaining connectivity among them and ensuring a good coverage of the area to monitor [8]. In [23, 7, 8], authors put a focus on velocity, obstacle avoidance, and UAVs or robot movement. Unlike our work, these papers do not focus on the application itself and how to optimally perform the data gathering process. However, it would be interesting to consider high-level constraints on movement in future works. iii) The *operational research community* is mostly focused on high-level path planning, looking for optimal trajectory [24, 25], optimal topology construction [26] or to maximize end-to-end throughput in UAV networks [27]. This community mostly proposes mathematical formulations for the coverage (which can be coupled with data gathering) and positioning problems using UAVs to find optimal or approximated solutions. Most contributions from this community simplify the network and robotics aspects and do not consider the network's connectivity during the data gathering process nor the 3D-placement of mobile robots/UAVs ([24, 25, 26, 27]).

Some contributions from the literature are very interesting since they try to mix the constraints, objectives, characteristics, and needs from different communities. For example, [10] mixes approaches from the operational research community and the robotics community by providing an optimal path planning solution for data gathering with multiple robots while taking into account the physical constraints such as velocity or turning curvature. In the same way, [20] takes into account the physical constraints of a robot while guaranteeing wireless mobile sensor network connectivity. In [28], dynamic programming is used to find an optimal routing for two camera-equipped UAVs, cooperatively tracking a single target moving on the ground. In [29], the authors provide a study on node placement including connectivity constraints, but the model is not optimal and is based on approximation. Moreover, the algorithms are designed for 2D-space with fixed coverage radius. In [6], the authors consider maximizing the total coverage area of the UAVs and their lifetime. But, in the models, the UAVs are not connected and assumed to be placed at the same altitude and thus, do not consider real 3D-positioning. [30] uses the same assumptions as we do in this paper. However, they to not consider connectivity between UAVs. In our paper, following the previous work from [31], we provide a solution that considers constraints from the wireless sensor community, namely connectivity and communication quality, while computing the optimal placement of each UAV. Moreover, we show that the classical Aerial Data Collection Problem (ADCP) formulation is outperformed by the connected set formulation for bigger instances of ADCP, and we propose a heuristic algorithm to enhance resolution time while relaxing the optimality condition.

2.3. Specific positioning of our contribution

In this section, we position our paper regarding relevant specific objectives and constraints of ACDP, and with the related literature. We briefly expose the originality of our paper compared to the state of the art. The following points highlight the specificity of our work and Table 1 shows the positioning compared to the literature. This table highlights that, compared to the literature, the models we described here include different assumptions and constraints. Some of the assumptions and constraints were individually tackled in the literature but our model considers them together in a complete framework.

- **application:** The application we consider is the coverage of ground sensors by UAVs. Therefore, we consider the 3D-positioning of each UAV which affects the coverage range on the ground since ground coverage depends on the UAVs' altitudes.

- **sensors' characteristics:** We assume that ground sensors are mobile (random unknown mobility). We want to ensure that all sensors are covered by a UAV at any time.
- **data gathering:** Data gathered by the UAVs should be transmitted by the UAVs to a base station. Since the distance between the UAVs and the base station can exceed the communication range of a UAV, the network of UAVs should be connected in a multi-hop fashion to the base station at any time.
- **objectives:** We want to minimize the number of UAVs for global cost minimization of ADCP, and for better management and coordination of the UAVs. We also consider the minimization of their respective altitudes and positions to keep them as close as possible to the base station in order to sustain the traffic. Minimizing the number of UAVs in real deployment is important, since it reduces the deployment cost. From a networking and communication point of view, reducing the number of UAVs also limits the possible communication interferences. It is important to notice that reducing the number of UAVs may affect robustness, but this point is left for future work.

3. Problem definition

From an optimization point of view, data collection from a set of nodes to a central entity has generally been modeled by two different approaches [32, 33]. First, the problem can be viewed as a *vehicle routing problem* in which the base station is the collecting point, and the goal is to compute the route of each vehicle, visiting all the customers once [34]. Second, the problem can be viewed as a *set covering problem* in which a subset of nodes is linked to all the nodes in a graph [35]. In our case, we choose to model our aerial data collection problem as a set covering problem since the UAVs evolve in the air with an associated altitude. Moreover, since the ground targets are mobile, we consider its dynamic version [36], minimizing deployment cost and data collection delay. We also use equations related to flow problems in order to ensure the connectivity and coverage for ADCP, as seen in the next sections.

3.1. ADCP assumptions

Each UAV u is located in the three-dimensional space and P is the set of possible 3D locations. Let $p = (x_u, y_u, h_u) \in P$ be the position (x_u, y_u) of UAV u in the 2D-plane and h_u its altitude. Time is discretized so that the target positions at each time are estimated and given by the successive sets N^t, for $t \in [0, T]$, where T is the monitoring time period length. Each target $n \in N^t$ is located on the ground and associated with two-dimensional coordinates (x_n^t, y_n^t).

We compute the observation radius r_u^h of UAV u depending on its altitude h_u and its directional antenna half beam-width, or visibility angle θ [5]. The coverage area of UAV u on the 2D-plane is represented by a disk of radius bounded by:

$$r_u^h \leq h_u \cdot \tan\left(\frac{\theta}{2}\right).$$

We assume that a UAV u deployed at location $p \in P$ covers target $n \in N^t$ if the euclidean distance between its projection on the 2D-plane and the target $d(u, n)$ is below the observation radius of the UAV:

$$d(u,n) = \sqrt{(x_u - x_n^t)^2 + (y_u - y_n^t)^2} \leq r_u^h.$$

Similarly, for air-to-air communications, a UAV u can communicate with another UAV v if their distance on the 3D plane

$$D_{uv} = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2 + (h_u - h_v)^2} \leq R_u,$$

where R_u is the communication range of UAV u. For efficient data collection, we enforce the deployed UAVs to be connected with each other (through possible multi-hop communications) and with the base station b located on a corner of the area.

3.2. Dynamic graph model

Given the target sets N^t, the coverage radius, and the communication range of the UAVs, one can derive a dynamic representation of the topology of our problem. Due to their dynamic and evolving characteristics, these networks are best represented by dynamic graph models [37]. For ADCP, we define a set of graphs $G^t = (V^t, E^t)$ in which the topology changes at each time $t \in [0, T]$.

Definition 1. *At time $t \in [0, T]$ of the observation time period, the current topology of ADCP is modeled by the directed graph $G^t = (V^t, E^t)$ where:*

- $V^t = \{b\} \cup P \cup N^t$, *and*
- $E^t = \{(b, p),\ p \in P \text{ and } (x_p, y_p) = \min_{q \in P} d(b, q)\} \cup \{(p, q),\ p, q \in P \text{ and } R_p \geq D_{pq}\} \cup \{(p, n),\ p \in P, n \in N^t \text{ and } r_p^h \geq d(p, n)\}$.

The set V^t contains the base station, all the positions for deploying drones, and the location of the targets at time t. Edges in E^t are defined to ensure (i) the connectivity of the backbone induced by the UAVs and the base station, and (ii) the coverage of all the targets at their current location. We assume that the base station has an edge to the positions p for which the projection on the 2D plane is the closest to the base station.

3.3. ADCP objective

In this section, we define a particular cost representation used in the objective function of ADCP involving considerations about cost, wireless communications, and delay. First, the management and coordination of the UAVs represent a specific difficult task to optimize. We thus seek to minimize the global cost of using a fleet of UAVs by minimizing the number of deployed UAVs over time. Moreover, wireless communication quality depends on various parameters such as environmental characteristics, wireless interferences, distance between devices, etc. Deploying the UAVs at low altitude maximizes the communication quality between the targets and the UAVs. However, it leads to more deployed UAVs, since the coverage

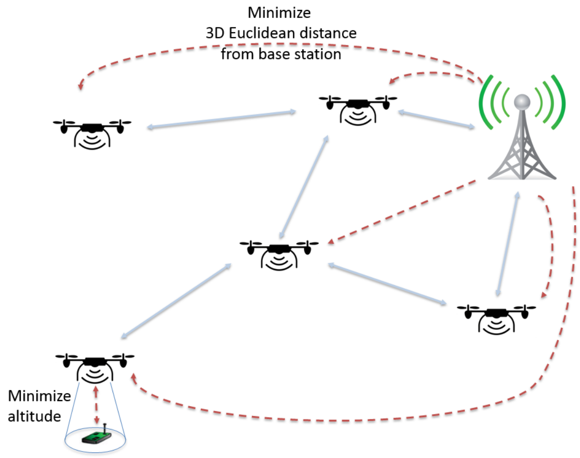


Figure 2: ADCP objectives.

area is smaller. The trade-off between these two antagonistic objectives has been studied in [38] for static targets.

In this paper, we define a cost parameter combining the different targeted optimization objectives:

Definition 2. *To each position $p \in P$, we associate a cost depending on the 3D-location:*

$$C_p = D_{bp} = \sqrt{(x_u - x_b)^2 + (y_u - y_b)^2 + (h_u)^2},$$

which is the distance in the 3D-plane between the position p and the base station b.

Using this cost in the objective function of our optimization model jointly optimizes the number of UAVs, their distances from the base station, and their altitudes (Figure 2). This allows for a better management of the fleet of flying drones, limiting their energy consumption over time and the delay of data collection. With the objective of determining a UAV positioning at each time step and minimizing the sum of their associated cost C_p, our optimization model of ACDP ensures that, (i) the UAVs stay close to the base station for efficient data collection, (ii) the UAVs are at the lowest altitude possible to ensure good connectivity with the sensors, and (iii) we use the minimum number of UAVs.

3.4. Scenario and tools description

Models developed in this paper have been implemented in the Java language using the JGraphT library[1], solved using IBM Cplex solver 12.7.1 on an Intel(R) Core(TM) i7-5500U CPU, 2.40 GHz, 16 Gb RAM machine, running Microsoft 8.1 Professional operating system.

Instances are deployed in a square area of size 100 m × 100 m. We consider 10 time slots for target mobility. The 2D-coordinates of the targets are initially chosen randomly in the monitored area. Their final destination and their velocity are also randomly chosen. At each time step, the next position of each target is computed based on their previous location, their velocity, and their direction. If a target reaches its destination coordinates before the last slot, it stays at its final location for the remaining slots.

We divide the monitored area into equal squares. One possible 2D-position of a UAV is located in the center of each square. In this way, the candidate sites for placing a UAV form a regular square grid. For each 2D-coordinate of point p, (x_p, y_p), we set the allowed altitudes to $\{10\,\text{m}, 25\,\text{m}, 45\,\text{m}\}$. The base station is placed on the bottom left corner at coordinates $(0.0, 0.0)$ with an altitude of 0 m.

Our instances contain between 5 and 50 targets and between 75 and 300 possible UAV locations (i.e., from 5×5 2D-coordinates $\times 3$ altitudes to 10×10 2D-coordinates $\times 3$ altitudes). The UAV visibility angle θ is set to 60 degrees, and their communication range to 30 m. Table 2 summarizes the parameters used in our scenario.

	Properties
Number of Sensors $\lvert N^t \rvert$	[5;50]
Sensors' mobilities	random
Terrain size	100 m×100 m
2D-positions (grid)	[5×5] - [10×10]
Possible altitudes	{10 m, 25 m, 45 m}
Base station coordinates	(0,0) - alt. 0m
Observation angle θ	60 degree
Air-to-air communication range R_u	30 m
Number of time slots	10

Table 2: Summary of instance generation.

4. Classical ADCP formulation

4.1. Mixed integer linear programming formulation (MILP)

Let z_p^t be a binary variable stating if position $p \in P$ is selected at time t to deploy a UAV. Classical ADCP can be modeled as a flow problem, given the temporal graphs of Definition 1. The existence of a flow in the graph G^t from a source node, the base station b, to destination nodes, the targets of N^t, ensures that we obtain a connected set of flying drones with the base station, and such that the selected locations for the UAVs cover all the targets at time t. Indeed, there exists an edge between a UAV position and a target if and only if the target is located within the coverage area of the UAV. We thus define $f_{uv}^t \in \mathbb{R}$ as the amount of flow going through link $(u, v) \in E^t$ at time $t \in [0, T]$.

The linear formulation of ADCP, called ADCP_MILP (for ADCP Mixed Integer Linear Program), is the following :

$$\min \left(\sum_{t=0}^{T} \sum_{p \in P} C_p z_p^t + \max_{t \in [1,T]} \left| \sum_{p \in P} C_p z_p^t - \sum_{q \in P} C_q z_q^{t-1} \right| \right) \quad (1)$$

$$\sum_{q \in V^t, q \neq p} f_{pq}^t - \sum_{q \in V^t, q \neq p} f_{qp}^t = \begin{cases} |N^t| & \text{if } p = b \\ 0 & \text{if } p \in P \\ -1 & \text{if } p \in N^t \end{cases}, \quad \forall t \in [0, T], p \in V^t \quad (2)$$

$$f_{pq}^t \leq z_p^t \cdot |N^t|, \ \forall t \in [0, T], (p, q) \in E^t \text{ with } p \in P \quad (3)$$

$$f_{pq}^t \in \mathbb{R}, z_p^t \in \{0, 1\} \quad (4)$$

[1] http://jgrapht.org/

Objective (1) not only minimizes the global deployment cost over time, but also the changes between two consecutive time slots. The metric used here is the one introduced in Definition 2. Recalling Section 3.3, the goal is to maintain a minimum number of UAVs to monitor the mobile targets, while placing them at a minimum distance from the base station. The first part of the objective function (1) ensures that we deploy the minimum number of UAVs as close to the base station as possible, ensuring the lowest altitude to cover the sensors and the fastest delay to gather the monitored information. The second part seeks to minimize the changes of selected positions. We want to minimize the difference in terms of cost (Def. 2) for two consecutive time slots, in order to minimize the delay of data collection between the selected positions over time. If positions need to change due to target mobility, then we ensure that the UAVs remain the closest possible to the base station, corresponding to the departure location of the drones. Then, minimizing their deployment cost over time gives a solution with limited UAV moves over time.

To get rid of the absolute value in the objective function, we introduce a new variable $\lambda \geq 0$. Objective (1) becomes :

$$\min\left(\sum_{t=0}^{T}\sum_{p\in P} C_p z_p^t + \lambda\right), \tag{5}$$

and we add the following constraints to the program:

$$\lambda \geq \sum_{p\in P} C_p z_p^t - \sum_{q\in P} C_q z_q^{t-1},\ \forall t \in [1,T] \tag{6}$$

$$\lambda \geq \sum_{p\in P} C_p z_p^{t-1} - \sum_{q\in P} C_q z_q^{t},\ \forall t \in [1,T] \tag{7}$$

Minimizing λ thus gives the minimum absolute value of the cost evolution between two consecutive time slots.

Constraints (2) model the flow conservation. We want to ensure the existence of a flow of size $|N^t|$ sent by the base station b to the targets of N^t. The flow is forwarded by the UAVs until the targets receive one unit each. Since, a priori, we do not know the locations of the UAVs, we define constraints (3) ensuring that no flow can go through a link incident to an unchosen position. Consequently, if a location $p \in P$ is not chosen, i.e., $z_p^t = 0$, then no flow exists on its incident links.

4.2. Results and discussions

In Table 3, we can see that even for small topologies with 5 targets and 75 possible locations for the UAVs (i.e., 5×5 coordinates $\times 3$ altitudes), ADCP_MILP takes several thousands of seconds ($\sim$ 4500s) to compute the optimal solution. However, when the number of targets is doubled from 5 to 10, then resolution time is multiplied by 3 for $|P| = 75$. When the number of possible 3D-locations is increased from 75 to 108, the resolution time is multiplied by 30 for $|N^t| = 5$. Consequently, for larger numbers of targets or possible UAV positions, ADCP_MILP does not scale and is unable to give a solution, as the CPLEX solver runs out of memory.

Indeed, this mixed-integer formulation suffers from the number of binary variables of the deployment constraints and time period length. Even for small networks and considering few 3D positions and time slots, the program generates MILPs with a huge number of constraints and integer variables. This combinatorial hardness makes the problem intractable for topologies with more than one hundred 3D-positions. In order to tackle larger topologies, we introduce an approach based on a decomposition model in the following sections.

$\lvert N^t\rvert$ \ $\lvert P\rvert$	75	108
5	4575.593	129764.829
10	12543.451	

Table 3: Resolution time in seconds of ADCP_MILP.

5. Connected set (CS) formulation

We now present a new formulation of ADCP, called ADCP_CG_OPT (for ADCP Column Generation, with Optimal pricing policy), involving sets of selected locations for the UAVs. This formulation allows us to separate our problem into two parts: (i) a master problem in which we deal with time and ensure a complete coverage of the targets at each time step, and (ii) a pricing problem generating connected subsets of UAVs with the base station, covering the targets at a given time.

Variables of the master program do not depend on one specific UAV location, but on more sophisticated structures, i.e., subsets of deployed UAVs with specific characteristics. Therefore, we do not deal with possible positions for each UAV, but only seek to select one subset of connected UAVs at each time in order to cover the sensors. We give a formal definition of the connected set considered in this new model.

Definition 3 (Connected set (CS))**.** *Let $S \subseteq P$ be a subset of 3D-positions with the following properties :*

- *One deployed UAV is associated with each 3D-position $(x_p, y_p, h_p) \in S$;*
- *Nodes of S form a connected graph with the base station b, i.e., there exists a path between b and every $u \in S$.*

We thus adapt the cost associated with each connected set, using Definition 2, in the following:

Definition 4 (CS cost)**.** *For each subset of UAVs S, we compute an associated cost depending on the 3D-location of the UAVs:*

$$C_S = \sum_{u\in S} D_{bu},$$

where D_{bu} is the distance on the 3D-plane between UAV $u \in S$ and the base station b introduced in Definition 2.

5.1. Master program

Let $\mathcal{S}$ denote the set of all possible CS S. $\mathcal{S}$ is exponential in size. The goal here is to prevent the enumeration of all possible subsets of UAVs using a known technique of optimization called *column generation*. Let z_S^t be a binary variable indicating if CS S is selected at time t. We introduce another set of binary variables χ_n^t to model the coverage of target n at time t. The linear master program modeling ADCP with $\mathcal{S}$ is the following:

$$\min\left(\sum_{t=0}^{T}\sum_{S\in\mathcal{S}}(z_S^t\cdot C_S)+\lambda\right) \quad (8)$$

$$\sum_{S\in\mathcal{S}} z_S^t = 1, \forall t\in[0,T] \quad (9)$$

$$\chi_n^t \geq 1, \forall t\in[0,T], n\in N^t \quad (10)$$

$$\chi_n^t \leq \sum_{S\in\mathcal{S}}\left(z_S^t\cdot\sum_{p\in S}|\{(p,n)\}\cap E^t|\right), \forall t\in[0,T], n\in N^t \quad (11)$$

$$\lambda \geq \sum_{S\in\mathcal{S}}(z_S^t\cdot C_S)-\sum_{S'\in\mathcal{S}}(z_{S'}^{t-1}\cdot C_{S'}),\ \forall t\in[1,T] \quad (12)$$

$$\lambda \geq \sum_{S\in\mathcal{S}}(z_S^{t-1}\cdot C_S)-\sum_{S'\in\mathcal{S}}(z_{S'}^{t}\cdot C_{S'}),\ \forall t\in[1,T] \quad (13)$$

The objective function (8) is an adaptation of the previous equation (5) to deal with $\mathcal{S}$. Constraints (9) state that we must select exactly one subset S of UAVs at each time t. Constraints (10) and (11) ensure that all the targets are covered at each time t if they are within the coverage area of at least one UAV of set S. For each position $p\in S$, we verify if the link (p,n) exists in the set of edges E^t of graph G^t. Given Definition 1, it means that the target $n\in N^t$ is within the coverage area of a UAV located at position $p\in P$. Therefore, target $n\in N^t$ is covered at time t if set S is selected at time t and at least one point p of S covers n. Constraints (12) and (13) concern the absolute value in the original objective function as constraints (6) and (7) do in the classical formulation of Section 4.

5.2. Pricing program

The sub-problem, or *pricing problem*, aims at generating new CS fulfilling the property of forming a connected backbone of UAVs with the base station. It is based on the dual formulation of the master problem.

Given the master problem (8)-(13), let $\beta^{(i)}$ be the dual variables associated with constraints (i). For each $S\in\mathcal{S}, t\in[0,T]$, the associated dual constraint is of the form:

$$C_S\cdot(1-\gamma)-\sum_{n\in N^t}\beta_{nt}^{(11)}\sum_{p\in S}|\{(p,n)\}\cap E^t|\geq\beta_t^{(9)}$$

where γ is a coefficient involving dual variables $\beta^{(12)}$ and $\beta^{(13)}$ depending on the value of t (i.e., $\gamma=f(\beta^{(12)},\beta^{(13)})$). Indeed, constraints (12) and (13) are defined for $t\geq1$ and involve variables z_S^t and z_S^{t-1}. We get the following definition of γ:

$$\gamma=\begin{cases}\beta_{t+1}^{(12)}-\beta_{t+1}^{(13)} & \text{if } t=0\\ -\beta_t^{(12)}+\beta_t^{(13)} & \text{if } t=T\\ \beta_{t+1}^{(12)}-\beta_{t+1}^{(13)}-\beta_t^{(12)}+\beta_t^{(13)} & \text{if } 0<t<T\end{cases}$$

The pricing problem of ADCP thus seeks to compute new CS violating the dual constraints, to add to the set of columns of the master problem. In other words:

Definition 5 (Minimum Weighted Connected Subset (MWCS))**.** *Given weight functions $\beta^{(i)}$ and a time slot $t\in[0,T]$, the* Minimum Weighted Connected Subset (MWCS) *consists in finding a subset $S\in\mathcal{S}$ of UAVs for which*

$$C_S\cdot(1-\gamma)-\sum_{n\in N^t}\beta_{nt}^{(11)}\sum_{p\in S}|\{(p,n)\}\cap E^t|$$

is minimum.

Thus, a minimum weighted connected subset generation either gives a good candidate to add to the set of variables of the master problem, or proves that no such column exists. If the cost computed by the pricing problem is smaller than $\beta^{(9)}$, the generated CS is added to the set of considered connected sets $\mathcal{S}$, and the corresponding variables z_S^t are added for all t. Constraints of this sub-problem define the structure of the CS of UAVs, fulfilling Definition 3.

Given the set P of possible 3D-locations for the UAVs, the goal is to select a subset of locations to deploy UAVs, such that they form a connected backbone with the base station efficiently collecting data from the mobile sensors on the ground.

In order to improve the efficiency of the computed subset, we define a pricing problem for every $t\in[0,T]$, and we specify that the connected subset must cover the sensors of N^t.

An optimal linear formulation of MWCS is similar to the classical formulation presented in Section 4, except that we remove the subscript t on each variable and only look for a single flow between the base station and the current target set N^t. To do so, we introduce variables $y_p\in\{0,1\}$ determining if location $p\in P$ is selected to be included in the CS. The objective of the pricing program is to minimize the weighted connected subset cost depending on the dual constraint presented above. We ensure the existence of a flow between the base station and the targets $n\in N^t$ (Constraints (15)), and ensure that no flow can go through incident links of unchosen locations (Constraints (16)).

$$\min\sum_{p\in P}y_p\left(C_p(1-\gamma)-\sum_{n\in N^t}\beta_{nt}^{(11)}\sum_{p\in P}|\{(p,n)\}\cap E^t|\right) \quad (14)$$

$$\sum_{q\in P,q\neq p}f_{pq}-\sum_{q\in P,q\neq p}f_{qp}=\begin{cases}|N^t| & \text{if } p=b\\ 0 & \text{if } p\in P\\ -1 & \text{if } p\in N^t\end{cases}, \quad \forall p\in V^t \quad (15)$$

$$f_{pq}\leq y_p\cdot|N^t|,\quad \forall(p,q)\in E^t \text{ with } p\in P \quad (16)$$

$$f_{pq}\in\mathbb{R}, y_p\in\{0,1\} \quad (17)$$

5.3. Resolution method

Column generation is a decomposition method that combines the resolution of a *restricted master problem*, i.e., the master problem with a limited number of variables/columns allowing the existence of at least one feasible solution of the linear relaxation of the problem, and a *pricing problem* or subproblem, generating new columns to add to the master problem in order to improve its objective value. When the pricing problem becomes infeasible, then by the *separation/optimization theorem* we know that the optimal value of the relaxed master problem has been reached.

We use column generation to optimally solve the linear relaxation of the master problem with a restricted set of initial variables. Indeed, we know that there exists an exponential number of CS as indicated in Section 5. At the beginning of our resolution process, we only generate one CS S_0 containing all the possible 2D-locations for UAVs, and placing them at the highest altitude available at these locations. This set S_0 is actually the most covering set, ensuring an initial coverage of the sensors at each time. S_0 is also guaranteed to be connected since there is one UAV placed at each possible position. However, this solution is not efficient in terms of objective value corresponding to the deployment cost. Placing a UAV at every possible location is very costly, and assigning them the highest possible altitude degrades the air-to-ground communication quality. Consequently, the column generation process iteratively solves the master problem and the pricing problem for every $t \in [0, T]$, generating new CSs S of UAVs with lower associated cost C_S, optimizing the MWCS objective involving the dual values obtained from the master problem resolution.

After each resolution of the master program, we get the dual costs associated with columns z_S^t and solve the pricing program t times to find new improving CSs. When a new CS S has been found by the pricing program for a specific time, we choose to generate t new variables z_S^t to add to the master program in order to be able to reuse the CS for other time slots than the one for which it has been found. This method allows us to accelerate the resolution of ADCP_CG_OPT as presented in the next section.

At the end of the column generation process, we thus obtain the optimal solution of the linear relaxation of the master program, giving a lower bound for our problem. We finally run the master program again, with the set of CSs found by the column generation process, but with binary variables z and χ. This gives us an effective integer solution of ADCP that we evaluate in the following.

5.4. Results for ADCP_CG_OPT

We use here the same scenario as presented in Section 3.4.

5.4.1. Performance of the model

Table 4 summarizes the performance of ADCP_CG_OPT. We present results for the tested topologies with different numbers of mobile sensors $|N^t|$ and different number of 3D-locations for the UAVs $|P|$, listed in columns ① and ②. Note here that, $|P| = 147$ means (7×7) 2D-locations $\times 3$ altitudes.

Column ③ presents the number of generated connected subsets of positions during the column generation process. We remark that this number is very low and always below 10 CSs. This means that the number of generated columns needed to reach the optimal value of the relaxed master problem is small compared to the exponential size of the set of possible CSs $\mathcal{S}$. In our model, when we run the pricing program for one particular value of t, the new subset found involves adding t columns in the master program, one for each time slot t. Consequently, a new subset of UAVs found for a specific time t can be reused for other time slots in the master program. This greatly improves the column generation process. However, the pricing program is ran for each time slot t in order to determine if there still exists a new subset to consider in the master program. This condition degrades the total resolution time as we will see in the next section.

Table 4: Computational results of ADCP_CG_OPT.

① $\lvert N^t \rvert$	② $\lvert P \rvert$	③ # generated CSs	④ z^*_{LP}	⑤ $\bar{z}_{ILP}$	⑥ ϵ	⑦ Time (s)
5	75	5	3424.389	3424.389	0	25.668
10	-	6	3905.826	3927.719	5.6×10^{-3}	31.473
15	-	8	5657.524	5707.063	8.7×10^{-3}	136.312
20	-	9	6096.653	6146.192	8.1×10^{-3}	249.439
25	-	10	6468.669	6484.211	2.4×10^{-3}	233.053
30	-	10	6574.307	6598.206	3.6×10^{-3}	276.054
35	-	9	7081.05	7081.05	0	537.511
40	-	8	7184.666	7203.332	2.6×10^{-3}	371.639
45	-	10	7501.873	7536.078	4.6×10^{-3}	517.278
50	-	10	7705.093	7718.27	1.7×10^{-3}	650.059
5	108	6	3227.604	3235.296	2.4×10^{-3}	144.652
10	-	9	3522.636	3545.914	6.6×10^{-3}	120.476
15	-	8	4958.595	5046.899	1.8×10^{-2}	935.043
20	-	7	5183.688	5183.688	0	888.504
25	-	8	5312.243	5330.563	3.4×10^{-3}	745.25
30	-	9	5345.651	5403.246	1.1×10^{-2}	769.817
35	-	8	5885.56	5917.416	5.4×10^{-3}	1315.132
40	-	10	6106.151	6188.343	1.3×10^{-2}	1150.683
45	-	10	6360.619	6376.655	2.5×10^{-3}	911.651
50	-	10	6559.157	6588.797	4.5×10^{-3}	1303.261
5	147	6	2889.468	2889.468	0	365.313
10	-	7	3128.349	3128.349	0	351.032
15	-	8	4606.988	4606.988	0	3553.59
20	-	8	4769.846	4774.719	1.0×10^{-3}	6670.563
25	-	9	5031.934	5047.334	3.1×10^{-3}	6415.417
30	-	9	5088.588	5092.94	8.6×10^{-4}	6440.275
35	-	8	5472.193	5517.763	8.3×10^{-3}	8352.025
40	-	9	5638.079	5674.783	6.5×10^{-3}	7682.315
45	-	10	5899.808	5922.879	3.9×10^{-3}	11361.689
50	-	9	5973.393	5996.662	3.9×10^{-3}	8723.237
5	192	6	2782.009	2782.009	0	2000.798
10	-	8	3174.343	3184.558	3.2×10^{-3}	1938.628
15	-	10	4774.253	4789.456	3.2×10^{-3}	52511.092
20	-	10	4879.655	4906.987	5.6×10^{-3}	69763.808
25	-	10	5077.339	5095.287	3.5×10^{-3}	68535.967
5	243	6	2652.332	2690.736	1.4×10^{-2}	11764.966
10	-	8	3001.447	3010.627	3.0×10^{-3}	9415.008
5	300	8	2576.889	2576.889	0	32964.699
10	-	8	2761.192	2776.259	5.4×10^{-3}	15286.829

Column ④ gives the optimal value of the linear relaxation of the master program (z^*_{LP}) obtained at the end of the column generation process. Column ⑤ gives the value of the integer master program with the set of variables of the last iteration of the column generation ($\bar{z}_{ILP}$). Column ⑥ provides the accuracy of the integer solution ϵ which is computed as $\epsilon = 1-(z^*_{LP}/\bar{z}_{ILP})$. ϵ can be seen as the worst-case distance between $\bar{z}_{ILP}$ and the optimal integer value of ADCP. A small value of ϵ thus means that $\bar{z}_{ILP}$ is very close to the optimal integer value. When $\epsilon = 0$, then the found integer solution is optimal for our problem. This optimum is reached for different numbers of sensors (5 to 20, and 35) and for different numbers of possible positions (75 to 300). We can see from column ⑥ of Table 4 that our solution is always at most at 2% away from the optimal value, validating our approach.

5.4.2. Quality of the solution computed with ADCP_CG_OPT

We investigate the number of deployed UAVs for different numbers of mobile sensors. It is important to note here that the number of deployed UAVs depends on the evolving position of the sensors on the ground and thus, depends on time. We can see from Figure 3 that the number of deployed UAVs increases with the number of sensors to cover, for all numbers of 3D-positions $|P|$. We present the mean number of deployed UAVs over time by a bar inside each boxplot in Figure 3. We can remark that the number of UAVs is less than or equal to the number of sensors. Indeed, a UAV can cover more than one sensor, optimizing the deployment cost in an effective way.

Figure 3 also depicts that the number of UAVs increases or is reduced on average by 1 unit over time (except for $|N^t| = 35$ and $|P| = 108$). This information is given by the size of the boxes in the boxplot. For some scenarios, the number of UAVs remains constant over time, especially for $|P| = 147$, or when the number of sensors increases ($|N^t| = 45$ or 50 for $|P| = 75$). This behavior is interesting for real deployment because it limits the use of new UAVs from the base station and enforces re-deployment of already placed UAVs.

In Figure 4, we depict the evolution of the cost of the selected sets over time. We seek to keep the cost as constant as possible due to the second part of the objective function of our optimization problem. However, the sensors' mobility enforces the UAVs to often change positions. In particular, when the number of sensors increases, then a configuration when several sensors are located close to each other can drastically change the cost of the subsets. For instance, the case for $|N^t| = 40$ and $|P| = 108$ has a drop of the cost at time $t = 1$. This is due to the removal of a UAV located far away from the base station. The UAV is then needed again, making the cost become higher again. Moreover, the evolution of cost is optimized only for consecutive time slots. Due to the mobility pattern of the sensors, small changes are needed at each time, either by adjusting the altitude of the UAVs, or by changing the location of the most remote UAV.

5.4.3. Resolution time

Table 4 provides raw value of ADCP_CG_OPT computation times in column ⑦. We report in Figure 5 the evolution of the computation time depending on both the number of sensors and the number of possible 3D-positions. We remark that it takes at most 1303.261s (~ 20 minutes) to compute solutions for instances with less than 147 possible 3D-positions for the UAVs, combined with any number of sensors. From Figure 5, we can see that the resolution time increases exponentially with the number of positions beyond 147 possible 3D-positions. When the number of sensors increases, computational time also increases to several hours of resolution, for example with 15 sensors and 192 possible 3D-positions for which the resolution time is 52511s (~ 15h). This is due to the complexity of the pricing program which is NP-hard since it is related to a *Steiner tree problem*. Moreover, we have here the additional constraint that we do not know a priori the set of nodes to include in the tree. It is worth noting that when the number of time slots is increased, the time for the column generation process also increases since we need more pricing resolutions.

Compared to ADCP_MILP, ADCP_CG_OPT shows better computational performance and provides solution quality close to the optimal. However, when both the number of 3D-positions and sensors increases, the computation time becomes unrealistic for a reactive system as the one we describe. It is important to find a trade-off between the computation time and the solution quality. This trade-off is studied in the following sections.

6. Heuristic pricing policy

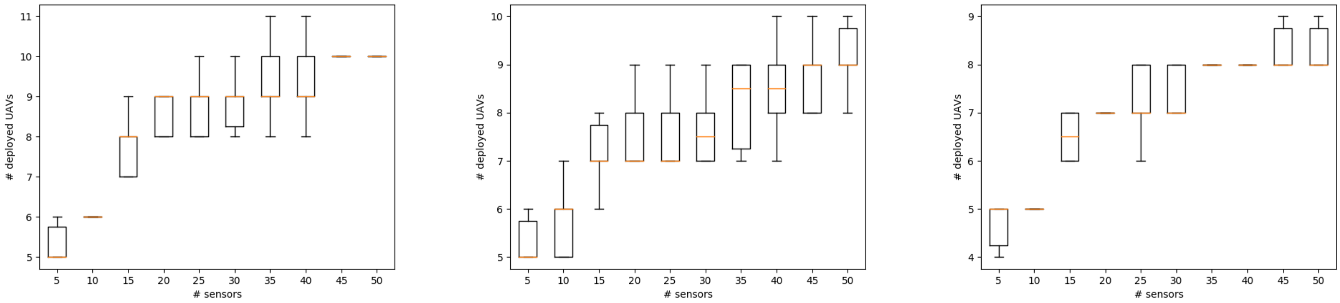
In order to improve the pricing program resolution and thus, reduce the computation time, we change our pricing program policy and propose a fast heuristic algorithm instead of the optimal pricing computation presented in Section 5.2 for MWCS optimization. The algorithm is presented in the next section, and we analyze the results found on our tested scenarios in Section 6.2.

6.1. Heuristic MWCS algorithm

Algorithm 1 Heuristic MWCS

Require: Time t, $G^t = (V^t, E^t)$, dual costs $\beta^{(i)}$
  $G = G^t, N = N^t, S = \emptyset$
  for all $e = (p, q) \in E(G)$ **do**
    if $q \in N^t$ **then**
      $cost_e = 0$
    else
      $cost_e = D_{bq} \cdot (1 - \gamma) - \sum_{n \in N^t} \beta^{(11)}_{nt} |\{(q, n)\} \cap E(G)|$
  $s = b$: source node (at the beginning it is the base station)
  while $N \neq \emptyset$ **do**
    $\mathcal{L} = \{L = (s, l_1, ..., l_k, n)$ shortest paths from s to $n \in N\}$
    Select L^* s.t. $l^*_1 = \max_{L \in \mathcal{L}} h_{l_1}$ and $cost_{L^*} = \max_{L \ni l^*_1} \sum_{e \in L} cost_e$
    if $length(L^*) = 1$ **then**
      $N = N \setminus \{n\}$: path to n has been found
    else if $\sum_{e \in L^*} cost_e = 0$ **then**
      $S = S \cup \{l^*_1, ..., l^*_k\}$ and $N = N \setminus \{n\}$
    else
      $S = S \cup \{l^*_1, ..., l^*_k\}$ and $N = N \setminus \{n^* = l^*_k\}$
      for $e \in L^*$ **do**
        $cost_e = 0$: encourage using same intermediate nodes
      $s = l^*_1$: new source for shortest paths
  return new CS $S \subset P$ covering N^t and connected with b

We use Dijkstra shortest paths algorithm (ADCP_CG_DIJ) to find a new CS (see Algorithm 1) as the pricing program in our column generation process. We iteratively compute the weighted shortest paths from the base station to the targets in the graph G^t for time t, in which the weight of link $(p, q) \in E^t$ corresponds to the dual cost associated with Definition 5. We run $|N^t|$ times the Dijsktra algorithm. At each step, the longest path found is selected and we record the positions of the UAVs



(a) $|P| = 75$. (b) $|P| = 108$. (c) $|P| = 147$.

Figure 3: Distribution of the size of selected subsets of UAV positions over time.

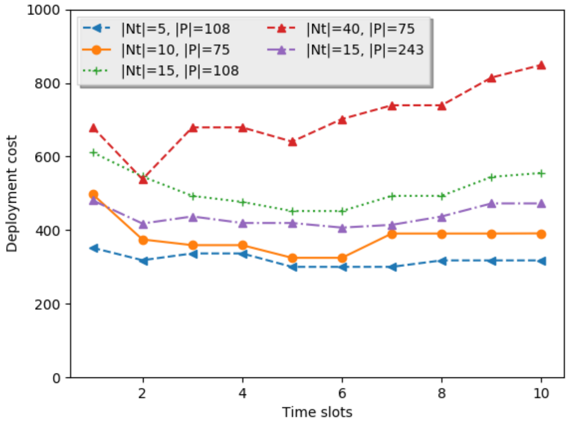


Figure 4: Evolution of deployment cost over time.

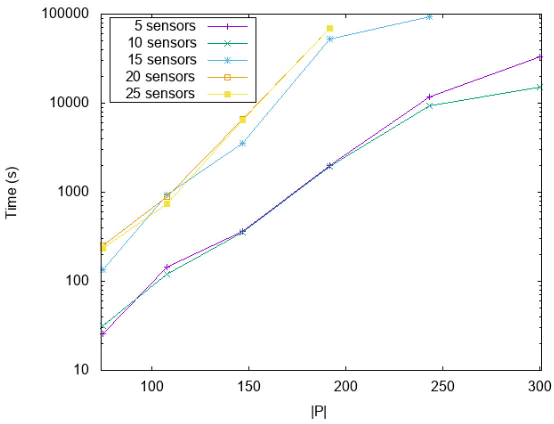


Figure 5: Resolution time in function of the number of 3D-positions for ADCP_CG_OPT. (y axis is in log scale)

of the path. The link costs of the selected path are set to 0 in order to encourage the other computed paths to use the same edges. Therefore, we obtain a CS fulfilling the required properties and check if it violates the dual constraint to add it to the master program.

6.2. *Results for ADCP_CG_DIJ*

6.2.1. *Performance of the model*

Table 5: Computational results of ADCP_CG_DIJ.

① $\lvert N^t\rvert$	② $\lvert P\rvert$	③ # generated sets	④ z^*_{LP}	⑤ $\bar{z}_{ILP}$	⑥ ϵ	⑦ Time (s)
5	75	5	3424.389	3909.936	1.2×10^{-1}	0.204
10		5	3905.826	4510.616	1.3×10^{-1}	0.159
15		6	5657.524	6339.536	1.0×10^{-1}	0.122
20		6	6096.653	6415.276	4.9×10^{-2}	0.186
25		6	6468.669	6763.646	4.3×10^{-2}	0.255
30		5	6574.307	6866.951	4.2×10^{-2}	0.258
35		6	7081.05	7434.808	4.7×10^{-2}	0.292
40		8	7184.666	8000.568	1.0×10^{-1}	0.313
45		8	7501.873	8413.125	1.0×10^{-1}	0.372
50		9	7705.093	8536.661	9.7×10^{-2}	0.431
5	108	5	3227.604	3878.835	1.6×10^{-1}	0.215
10		8	3522.636	4463.87	2.1×10^{-1}	0.246
15		10	4958.595	5970.774	1.6×10^{-1}	0.361
20		10	5183.688	5799.398	1.0×10^{-1}	0.407
25		8	5312.243	6240.422	1.4×10^{-1}	0.485
30		7	5345.651	6368.844	1.6×10^{-1}	0.528
35		9	5885.56	7606.585	2.2×10^{-1}	0.539
40		10	6106.151	7608.826	1.9×10^{-1}	0.563
45		9	6360.619	8085.365	2.1×10^{-1}	0.641
50		10	6559.157	8518.769	2.3×10^{-1}	0.715
5	147	8	2889.468	3205.161	9.8×10^{-2}	0.280
10		7	3128.349	3369.58	7.1×10^{-2}	0.450
15		9	4606.988	5553.485	1.7×10^{-1}	0.510
20		8	4769.846	5526.248	1.3×10^{-1}	0.588
25		8	5031.934	5756.901	1.2×10^{-1}	0.919
30		9	5088.588	6080.205	1.6×10^{-1}	0.868
35		10	5472.193	6344.368	1.3×10^{-1}	1.135
40		9	5638.079	6320.016	1.0×10^{-1}	1.144
45		10	5899.808	6743.387	1.2×10^{-1}	1.147
50		9	5973.393	6907.604	1.3×10^{-1}	1.218
5	192	8	2782.009	3359.266	1.7×10^{-1}	0.799
10		10	3174.343	3979.738	2.0×10^{-1}	0.800
15		9	4774.253	5719.449	1.6×10^{-1}	1.103
20		9	4879.655	5815.745	1.6×10^{-1}	1.244
25		9	5077.339	6416.000	2.0×10^{-1}	1.206
5	243	9	2652.332	3371.896	2.1×10^{-1}	1.396
10		10	3001.447	3584.85	1.6×10^{-1}	1.211
5	300	10	2576.889	3059.354	1.5×10^{-1}	1.412
10		10	2761.192	3663.003	2.4×10^{-1}	1.542

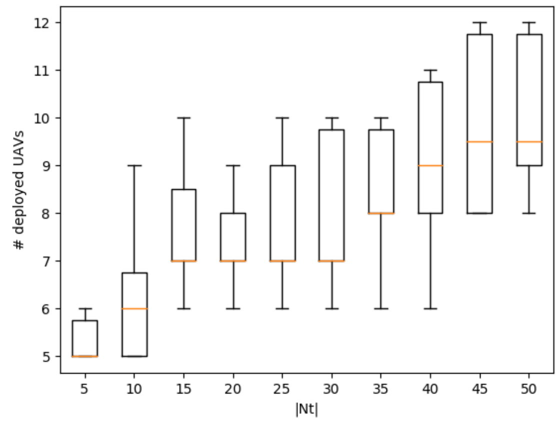
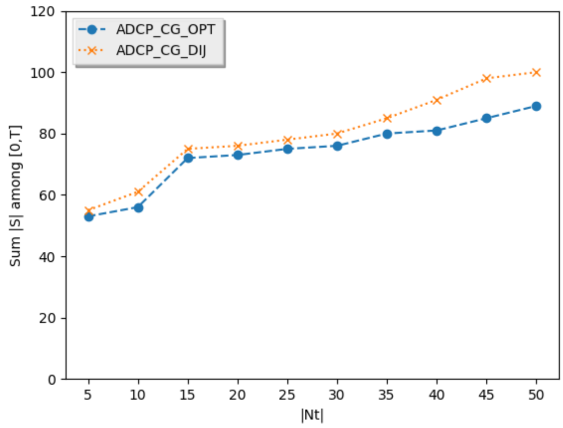
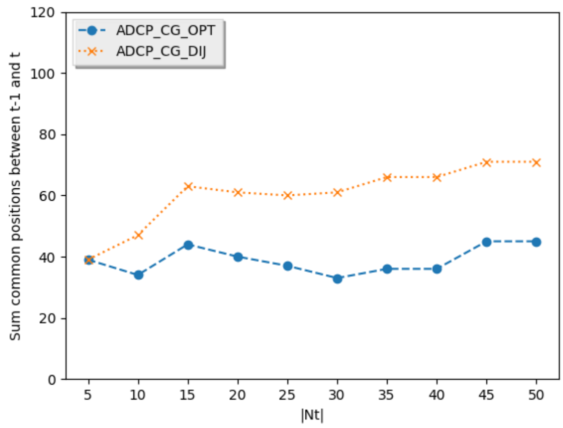


Figure 6: Distribution of the size of selected CSs over t for $|P| = 75$ with ADCP_CG_DIJ.



(a) Sum of $|S|$.



(b) Common positions.

Figure 7: Evaluation of Dijkstra pricing for $|P| = 75$.

The performance of ADCP_CG_DIJ is summarized in Table 5. Topology characteristics are described in columns ① and ②: it shows the number of mobile targets $|N^t|$ and number of possible 3D-locations for the UAVs $|P|$. Recall for example that when $|P| = 147$ it means 49 possible 2D-locations and 3 possible altitudes. Columns ③, ④, ⑤, and ⑥ are similar to Table 4 and describe here the results obtained for the column generation model with the heuristic pricing policy (ADCP_CG_DIJ). First, we can remark that the number of generated subsets during the column generation process (Column ③ of Table 5) is very low and always below 10 as has already been noticed for ADCP_CG_OPT. In columns ④, ⑤, and ⑥, as for ADCP_CG_OPT, we recall the optimal value of the linear relaxation of the master program (z^*_{LP}) and the value of the integer master program with the set of variables of the last iteration of the column generation ($\bar{z}_{ILP}$) together with the accuracy ϵ. Solutions for ADCP_CG_DIJ are less accurate than ADCP_CG_OPT since their accuracy has values between 8 and 26% away from optimal (see column ⑥), however it generates less subsets for small topologies to obtain very efficient solutions in a fast resolution scheme.

6.2.2. *Resolution time*

The time needed to compute the solution for ADCP_CG_DIJ is below ~ 1.6s for all instances with any combination of different numbers of 3D-positions $|P|$ and sensors $|N^t|$. The raw data for the computation time of ADCP_CG_DIJ are given in Table 5 column ⑦. This validates our policy to improve the computational time of ADCP.

6.2.3. *Quality of the solution computed with ADCP_CG_DIJ*

In Figure 6, we investigate the number of deployed UAVs over time for different numbers of mobile targets. We present the mean value over time by a bar inside each boxplot. We can see that the mean size of selected subsets is similar to ADCP_CG_OPT (see Figure 3a), and this mean value can also be lower than the one obtained with the optimal pricing policy, for example when $|P| = 75$ and $|N^t| = 20$. However, for a given topology, the evolution of the size during the time period changes more than ADCP_CG_OPT. For each topology, the variation of the CS size stays usually bounded by 3. And for a large number of sensors, we see that ADCP_CG_DIJ generates bigger CSs than ADCP_CG_OPT (12 UAVs needed for the heuristic while the optimal pricing uses only 10 UAVs for $|N^t| = 50$).

We investigate more specifically the solutions of ADCP_CG_DIJ in Figure 7. They globally use more UAVs than ADCP_CG_OPT during the time period (see Figure 7a), and the placed UAVs are located at higher altitudes due to Algorithm 1. However, the selected sets of ADCP_CG_DIJ maintain the positions more efficiently than ADCP_CG_OPT (see Figure 7b). This is an advantage for maintaining data collection and maximizing the aerial network lifetime.

7. Discussion and future works

In this section, we discuss the limitations and possible extensions of our work. Especially, we discuss the assumptions made in our model and the possible usage of the results.

- Application: This work will be used to solve, design and benchmark a real application that will be developed for wild animal tracking designed by biologists for marine turtle tracking. The turtles are tracked by a fleet of flying drones when they are in a bay area close to the base station. Related to the assumptions given by the biologists, the turtles are equipped with sensors (temperature, cardiac frequency, etc.), a GPS, and two communication interfaces. One interface (satellite) is used to send the position of the turtle to the base station when they move to the surface in the bay. This position is used as a trigger to deploy the fleet of drones. The second interface (not decided so far) will be used to transmit data from the turtles to the drones.

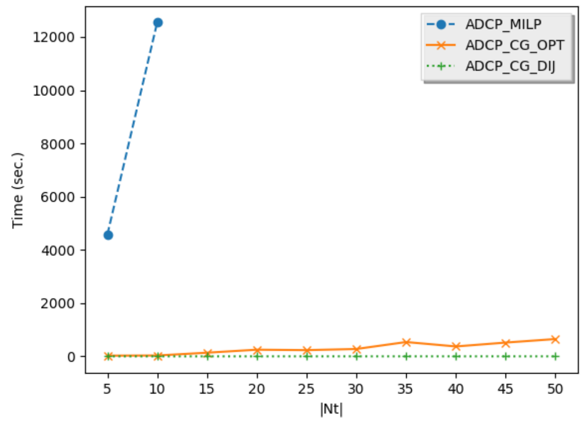
- Computation time: Figure 8 compares the resolution time of the three models. In Figure 8a the number of 3D-positions is fixed with $|P| = 75$. In Figure 8b the number of sensors is fixed with $|N^t| = 5$. These figures show the scalability of ADCP_CG_DIJ compared to the other methods. However, it is important to notice that choosing a method such as ADCP_CG_DIJ reduces the accuracy of the solution compared to the optimal pricing policy.

- Uplink and Downlink transmissions: In this paper, we focus on uplink transmissions from sensors to sink. However, the model can be used for data collection but also software updates. The permanent connectivity provided by the deployment of the UAV network allows such a behavior.

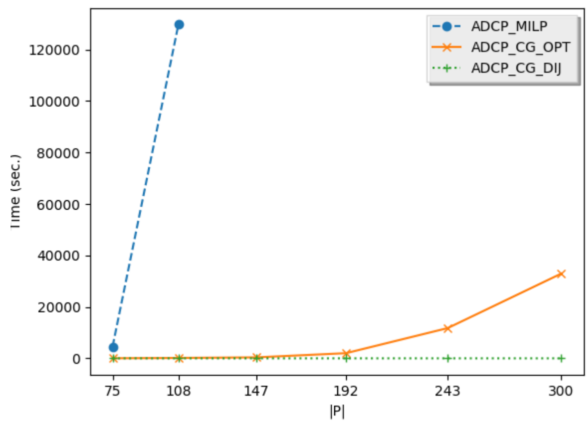
- Connectivity: In this paper, we focus on permanent connectivity of the network composed by UAVs. This constraint is strong and could be relaxed for applications where data from sensors are not required to be sent directly to the base station. An intermittent connectivity could be sufficient. For example, network connectivity should be established at regular periods and for a given duration. This can be solved by modifying our model, by relaxing the connected set problem at specific time slots. However, in our case, for the purpose of observation, data have to be sent in real time since different data are produced when the turtles are in the bay.

- Scalability: Depending on the size of the problem, the number of sensors, the size of the area to be observed, and the number of drones, the solution computed by the model may not be fast enough to be directly used to drive the drones. We are already working on some heuristics and some distributed models that could scale. In this case, the model described here will be used as a reference. However, based on the application description, the model could be used as is.

- Path planning: In this paper, we do not optimize the path of each drone but only provide successive positions that should be occupied. In future work, we will include an optimal path computation for each drone. Moreover, we will also include drone properties such as angle of curvature, etc.



(a) Evolution of resolution time in function of the number of sensors. $|P| = 75$



(b) Evolution of resolution time in function of the number of 3D-positions. $|N^t| = 5$

Figure 8

- Communication protocol: The next step of this paper is to design or adapt communication protocols above the UAV network or to use existing protocols and modify the topology of the UAV network to obtain optimal performance from the network protocols. This second case is interesting and is left to future work since protocol properties can be injected into the model.

- Practical considerations: It is important to link the results from this paper to practical implementation. In this work, we try to be generic enough to be able to easily modify the model to fit any technology. We think that the model presented in this paper could be associated with communication technology such as WiFi for the air-to-air communication, LoRa for the ground-to-air communication and that off-the-shelves drone technologies fit the assumptions of our model. However, further investigations should be made regarding this subject.

8. Conclusion

In this paper, we address the aerial data collection problem from a set of mobile targets using a fleet of UAVs. We present linear programming formulations: a MILP and a column generation approach with two different pricing policies. We show that ADCP_MILP does not scale for large instances, while ADCP_CG_OPT solves ADCP almost exactly with an NP-hard optimal pricing problem that becomes hard to solve with larger possible 3D-positions. We thus propose a heuristic pricing program generating larger subsets of UAVs but solving ADCP in less than 2 seconds with a small number of columns. Taking into account the work of the literature, this is the first model that scales with an increasing number of 3D-positions and targets, while considering mobility and connectivity.

Bibliography

[1] V. Dyo, S. Ellwood, D. Macdonald, A. Markham, N. Trigoni, R. Wohlers, C. Mascolo, B. Pasztor, S. Scellato, K. Yousef, Wildsensing: Design and deployment of a sustainable sensor network for wildlife monitoring, ACM Transactions on Sensor Networks.
URL `https://www.cl.cam.ac.uk/ cm542/papers/tosn2011.pdf`

[2] G. Guido, V. Gallelli, D. Rogano, A. Vitale, Evaluating the accuracy of vehicle tracking data obtained from unmanned aerial vehicles, International Journal of Transportation Science and Technology 5 (3) (2016) 136 – 151, unmanned Aerial Vehicles and Remote Sensing. doi:https://doi.org/10.1016/j.ijtst.2016.12.001.

[3] S. Hayat, E. Yanmaz, R. Muzaffar, Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint, IEEE Communications Surveys Tutorials 18 (4) (2016) 2624–2661. doi:10.1109/COMST.2016.2560343.

[4] C. Cambra, S. Sendra, J. Lloret, L. Parra, Ad hoc network for emergency rescue system based on unmanned aerial vehicles, Network Protocols and Algorithms 7 (4) (2015) 72–89. doi:10.5296/npa.v7i4.8816.

[5] A. Al-Hourani, S. Kandeepan, A. Jamalipour, Modeling air-to-ground path loss for low altitude platforms in urban environments, in: 2014 IEEE Global Communications Conference, 2014, pp. 2898–2904. doi:10.1109/GLOCOM.2014.7037248.

[6] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage, IEEE Comm. Lett. 20 (8) (2016) 1647–1650. doi:10.1109/LCOMM.2016.2578312.

[7] M. M. Zavlanos, M. B. Egerstedt, G. J. Pappas, Graph-theoretic connectivity control of mobile robot networks, Proceedings of the IEEE 99 (9) (2011) 1525–1540. doi:10.1109/JPROC.2011.2157884.

[8] E. Yanmaz, Connectivity versus area coverage in unmanned aerial vehicle networks, in: 2012 IEEE International Conference on Communications (ICC), 2012, pp. 719–723. doi:10.1109/ICC.2012.6364585.

[9] C. Caillouet, T. Razafindralambo, Efficient deployment of connected unmanned aerial vehicles for optimal target coverage, in: IEEE GIIS, 2017.

[10] H. Huang, A. V. Savkin, Viable path planning for data collection robots in a sensing field with obstacles, Computer Communications 111 (2017) 84 – 96. doi:https://doi.org/10.1016/j.comcom.2017.07.010.

[11] D. Zorbas, L. Di Puglia Pugliese, T. Razafindralambo, F. Guerriero, Optimal drone placement and cost-efficient target coverage, Journal of Network and Computer Applications 75 (2016) 16 – 31. doi:http://dx.doi.org/10.1016/j.jnca.2016.08.009.

[12] D. G. Macharet, J. W. Monteiro, G. R. Mateus, M. F. Campos, Bi-objective data gathering path planning for vehicles with bounded curvature, Computers & Operations Research 84 (2017) 195 – 204. doi:https://doi.org/10.1016/j.cor.2016.07.004.

[13] L. Chen, S. Lin, H. Huang, Charge me if you can: Charging path optimization and scheduling in mobile networks, in: Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '16, 2016, pp. 101–110. doi:10.1145/2942358.2942364.

[14] R. C. Shah, S. Roy, S. Jain, W. Brunette, Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks, Ad Hoc Networks 1 (2) (2003) 215 – 233, sensor Network Protocols and Applications. doi:https://doi.org/10.1016/S1570-8705(03)00003-9.

[15] E. B. Hamida, G. Chelius, A line-based data dissemination protocol for wireless sensor networks with mobile sink, in: Proceedings of IEEE International Conference on Communications, ICC 2008, Beijing, China, 19-23 May 2008, 2008, pp. 2201–2205. doi:10.1109/ICC.2008.420.

[16] S. Gandham, M. Dawande, R. Prakash, S. Venkatesan, Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in: Proceedings of the Global Telecommunications Conference, 2003. GLOBECOM '03, San Francisco, CA, USA, 1-5 December 2003, 2003, pp. 377–381. doi:10.1109/GLOCOM.2003.1258265.

[17] S. Basagni, A. Carosi, C. Petrioli, C. A. Phillips, Moving multiple sinks through wireless sensor networks for lifetime maximization, in: IEEE 5th International Conference on Mobile Adhoc and Sensor Systems, MASS 2008, 29 September - 2 October 2008, Atlanta, Georgia, USA, 2008, pp. 523–526.

[18] J. Luo, J. Hubaux, Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility, IEEE/ACM Trans. Netw. 18 (3) (2010) 871–884.

[19] A. Wichmann, J. Chester, T. Korkmaz, Smooth path construction for data mule tours in wireless sensor networks, in: 2012 IEEE Global Communications Conference, GLOBECOM 2012, Anaheim, CA, USA, December 3-7, 2012, 2012, pp. 86–92.

[20] T. Razafindralambo, D. Simplot-Ryl, Connectivity preservation and coverage schemes for wireless sensor networks, IEEE Trans. Automat. Contr. 56 (10) (2011) 2418–2428.

[21] Z. Becvar, M. Vondra, P. Mach, J. Plachy, D. Gesbert, Performance of mobile networks with uavs: Can flying base stations substitute ultra-dense small cells?, in: 23th European Wireless Conference, 2017.

[22] T. Razafindralambo, M. Erdelj, D. Zorbas, E. Natalizio, Spread and shrink: Point of interest discovery and coverage with mobile wireless sensors, Journal of Parallel and Distributed Computing 102 (Supplement C) (2017) 16 – 27. doi:https://doi.org/10.1016/j.jpdc.2016.09.003.

[23] J. L. Ny, E. Feron, E. Frazzoli, On the dubins traveling salesman problem, IEEE Trans. Automat. Contr. 57 (1) (2012) 265–270.

[24] C. Dixon, E. W. Frew, Optimizing cascaded chains of unmanned aircraft acting as communication relays, IEEE JSAC 30 (5) (2012) 883–898. doi:10.1109/JSAC.2012.120605.

[25] L. He, J. Pan, J. Xu, A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements, IEEE Trans. Mob. Comput. 12 (7) (2013) 1308–1320.

[26] F. Alduraibi, N. Lasla, M. Younis, Coverage-based node placement optimization in wireless sensor network with linear topology, in: IEEE ICC, 2016. doi:10.1109/ICC.2016.7511421.

[27] J. Chen, D. Gesbert, Optimal positioning of flying relays for wireless networks: A los map approach, in: IEEE ICC, 2017. doi:10.1109/ICC.2017.7996921.

[28] S. Quintero, F. Papi, D. Klein, L. Chisci, J. Hespanha, Optimal uav coordination for target tracking using dynamic programming, in: IEEE CDC, 2010, pp. 4541 –4546.

[29] K. Kar, S. Banerjee, Node placement for connected coverage in sensor networks, in: WiOpt, 2003.

[30] L. D. P. Pugliese, F. Guerriero, D. Zorbas, T. Razafindralambo, Modelling the mobile target covering problem using flying drones, Optimization Letters 10 (5) (2016) 1021–1052.

[31] C. Caillouet, F. Giroire, T. Razafindralambo, Optimization of mobile sensor coverage with uavs, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2018, Honolulu, HI, USA, April 15-19, 2018, 2018, pp. 622–627.

[32] A. A. Somasundara, A. Ramamoorthy, M. B. Srivastava, Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines, in: 25th IEEE International Real-Time Systems Symposium, 2004, pp. 296–305. doi:10.1109/REAL.2004.31.

[33] C. Hung, W. Peng, W. Lee, Energy-aware set-covering approaches for

approximate data collection in wireless sensor networks, IEEE Transactions on Knowledge and Data Engineering 24 (11) (2012) 1993–2007. doi:10.1109/TKDE.2011.224.
[34] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, S. L. Smith, Dynamic vehicle routing for robotic systems, IEEE proceedings 99 (9) (2011) 1482–1504.
[35] M. Popescu, H. Rivano, O. Simonin, Multirobot patrolling in wireless sensor network using bounded cycle coverage, in: ICTAI, 2016. doi:10.1109/ICTAI.2016.0035.
[36] J. W. Chrissis, R. P. Davis, D. M. Miller, The dynamic set covering prblem, Applied Mathematical Modelling 6 (1) (1982) 2 – 6.
[37] D. H. A. Zaki, M. Attia, S. Amin, Comprehensive survey on dynamic graph models, IJACSA 7 (2) (2016) 573–582.
[38] C. Caillouet, T. Razafindralambo, Efficient Deployment of Connected Unmanned Aerial Vehicles for Optimal Target Coverage, in: IEEE GIIS, 2017.