Computers & Operations Research 168 (2024) 106678

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Autonomous data detection and inspection with a fleet of UAVs

Tiziana Calamoneri [a], Federico Corò [b], Simona Mancini [c,d,*]

[a] Università di Roma La Sapienza, Roma, Italy
[b] University of Padova, Padova, Italy
[c] University of Palermo, Palermo, Italy
[d] University of Klagenfurt, Klagenfurt, Austria

ARTICLE INFO

Keywords:
Data detection
Data inspection
Matheuristic
Unmanned aerial vehicles (UAVs)

ABSTRACT

Consider an area of interest A, where a set of n *sites* lie. Two kinds of information can be captured from each site: light and heavy information. A fleet of m homogeneous UAVs, each one equipped with a battery B, is available at a common depot, where the flight mission of each UAV starts and finishes. The problem we consider focuses on a single flight of the fleet of UAVs and aims at collecting their light information from all sites (that can be retrieved, not necessarily passing over each site, but simply "close" to it). At the same time, the fleet will have to select a limited number of sites from which to collect their heavy information. Flying among sites and acquiring information from them (both light and heavy) has a battery cost. On the other hand, a profit is associated with the action of acquiring heavy information from a site. We refer to the extraction of light and heavy information from a site as to weakly or strongly cover the site. The aim of the problem consists of retrieving light information from all sites while maximizing the overall profit, keeping the battery consumption of each UAV within B. In this paper, we model this real-life situation as a new combinatorial optimization problem that we call m3DIP, for which we provide a mixed integer programming model. Given the high degree of complexity of the problem, in this way we are not able to provide a solution in a reasonable time. To address larger instances we propose a matheuristic in which we exploit a path-based algorithm filled with only a subset of feasible cycles (paths) provided by different heuristics. The output indicates which path to select and the set of nodes to be strongly and weakly covered by each trip. We compare our matheuristic with the results obtained by every single heuristic on a large set of instances, showing that the matheuristic strongly outperforms them. An interesting insight is that even paths provided by a heuristic with very bad performances can be useful if combined with paths provided by other heuristics and if the coverage decisions are reoptimized by the matheuristic. We also show the benefit of adding fictitious additional points that UAVs can visit to weakly cover a subset of sites, without actually visiting none of them.

1. Introduction

Unmanned aerial vehicles (UAVs) are aircraft whose flights can be fully autonomous without any provision for human intervention. UAVs were originally developed for military applications, but now that control technologies have improved and costs have decreased, their use has found a wide range of applications in many civilian and commercial sectors, such as weather monitoring, forest fire detection, traffic control, plant disease detection, cargo transport, patrolling, emergency search and rescue, etc. (*e.g.* see Di Gennaro et al. (2016), Liang et al. (2019), Sharifi et al. (2014), Thiels et al. (2015) and Valavanis and Vachtsevanos (2014)).

At a very high level, data detection is a generalized monitoring that detects some anomalies, while data inspection requires a deep verification of the situation. This paper focuses on applicative scenarios in which data detection is always required while data inspection is necessary only under certain conditions. In such real-life scenarios, UAVs are preferable with respect to ground robots because flying devices move faster and are not affected by eventual obstacles on the terrain.

In this paper, we introduce a new combinatorial optimization graph problem, which we call m3DIP, arising from some applicative scenarios (Section 2) and consisting of determining a number of cycles covering all the assigned cycles fulfilling certain constraints. In Section 3, we observe similarities and differences between it and a couple of very well-known problems, and this justifies the study of m3DIP as a new problem. So, in Section 4, we express m3DIP as an integer linear programming problem. Since in this way we are not able to provide a solution in a reasonable time, we propose a matheuristic (Section 5);

* Corresponding author at: University of Palermo, Palermo, Italy.
E-mail addresses: calamo@di.uniroma1.it (T. Calamoneri), federico.coro@unipd.it (F. Corò), simona.mancini@aau.at, simona.mancini@unipa.it (S. Mancini).

https://doi.org/10.1016/j.cor.2024.106678
Received 1 August 2023; Received in revised form 17 April 2024; Accepted 22 April 2024
Available online 22 May 2024

the idea is to give as an additional input an opportune subset of cycles so that the search space is reduced only to this subset instead of the entire set of all possible cycles. In Section 6, we propose some heuristics serving a twofold purpose: on the one hand, they generate the set of cycles given in input to the matheuristic; on the other hand, they can be seen as benchmark heuristics, revealing that the matheuristic largely outperforms each of them. Section 7 is devoted to showing the results of a large set of experiments, validating our matheuristic, and showing that, even if each heuristic alone is not able to achieve a good result, nevertheless, it actively contributes with some cycles to the input of the matheuristic; instead, the matheuristic – that exploits cycles collected from all the heuristics – largely guarantees the best profit.

2. Problem definition

In this section, we formally define m3DIP and propose some practical applications that can be modeled through it.

Consider an area of interest A, where a set $V = \{v_1, \dots, v_n\}$ of n *sites* is located. The largest distance between two sites is identified as d_{MAX}.

These sites can be either nodes with possibly multiple sensors from which collect data, or locations to be monitored with different precision so that two kinds of information can be captured for each of them: a piece of *light information* (for example, some simple sensed data, such as ground temperature or humidity, or medium resolution pictures) and a piece of *heavy information* (*e.g.*, a high-resolution video).

A fleet of m homogeneous UAVs $U = \{u_1, \dots, u_m\}$ is available at a depot v_0. The flight mission of each UAV starts and finishes at the depot. We denote by V^+ the set $V \cup \{v_0\}$.

Each UAV u_i is equipped with a *battery* B. The problem we consider focuses on a flight of the fleet of UAVs between two periods to recharge batteries and aims at collecting from *all* sites their light information that can be retrieved, not necessarily passing over each site, but simply "close" to it. Specifically, a UAV passes "close" to a site if it passes at a distance upper bounded by a parameter R from it; in other words, if there exists an instant in its flight when the site falls inside a circle centered at the UAV itself with radius R. We define γ_i as the number of sites located within a radius R from the site v_i. At the same time, the fleet will have to select a limited number of sites from which to collect their heavy information.

We say that a site has been *covered* when a UAV has captured a piece of information from it. Each site can be either weakly or strongly covered: a UAV *strongly covers* a site when it hovers over the site, acquiring heavy information about it. In contrast, the UAV *weakly covers* the site when it flies close to the site and acquires light information about it. Finally, a UAV can even decide to fly over a site without acquiring its information.

Flying between two sites v_i and v_j has a cost in terms of battery consumption, so we introduce a function q, indicating the battery spent in moving between two sites (regardless of the direction) and is typically (but not necessarily) proportional to the covered distance.

Also, acquiring information (both light and heavy) from a site has a cost for a UAV, so we introduce the two cost functions $b^w : V \to \mathbb{R}^+$ and $b^s : V \to \mathbb{R}^+$, representing the battery cost to respectively weakly and strongly cover site.

It is worth noticing that, although heavy information typically occupies a huge memory, we decide not to take care of memory cost (as done instead *e.g.* in Sorbelli et al. (2022)) because the current technology allows us to store even very large information easily. Indeed, an hour of 4K video uses about 45 GB of storage space, and commercial UAVs can easily be equipped with memory cards that range in storage between 64 and 512 GB.[1]

To drive the choice of which sites to visit to collect their heavy information, sites are labeled by a relevance function, and the objective of the whole fleet is to maximize the overall captured relevance in such a way that every UAV does not overrun its battery. The relevance of sites w.r.t. strong coverage is modeled by associating a profit to each site: the higher the profit, the more relevant the heavy data acquisition from that site. So, for each site, we also define a profit function $p : V \to \mathbb{R}^+$ that is gained by the fleet whenever a UAV strongly covers the site. The profit on each site is defined on the basis of some external information, *e.g.*, the time elapsed since the last accurate monitoring, some probability function, etc., and can be updated before each fleet flight.

Some possible applications of this model are the following:

- wind turbine inspection: sites are wind turbines; a thermal camera detects any energy loss and performance issues even from a relatively far location (light information) while, where necessary, *i.e.*, where profit is higher, a vision camera allows one to accurately view the fine details of the wind turbine (heavy information);
- animal species monitoring: identifying sites with lairs positioned on a slope, a thermal camera monitors all of them (light information); then, only holes with a higher probability of animal detection (and hence with a higher profit) will be more closely guarded with a high-performance camera (heavy information);
- cellular antennas or solar panels inspection/diagnostics: antennas are prone to bird's nests, lightning strikes, rust/corrosion, and damaged bolts, and they usually lie over high buildings and are often not easily accessible; panels are prone to delamination and corrosion, micro-cracks, PID (Potential Induced Degradation) effect, degrading their efficiency and performance. Moreover, both antennas and panels are in huge numbers, and UAV inspection eliminates the need to put people in harm's way and reduces person-hours and labor costs by automating inspections with low maintenance costs. Each antenna/panel is a site, light information corresponds to a rough visual inspection to evaluate at a high level any damage or potential problems, and a more accurate inspection (heavy information) is done only on some opportunely chosen antennas on the basis of their profits, computed keeping into account external factors such as time elapsed since the previous visit and the favorable position for bird nidification;
- smart agriculture: a number of sites can be identified in a field with extensive crop (*e.g.*, the points at integer coordinates w.r.t. opportune Cartesian coordinates); UAVs can be equipped with a variety of sensors that facilitate the analysis of a range of data: nitrogen levels, chlorophyll, biomass, humidity, water stress, etc. (light information); sites in special situations, as zones where a parasite has recently spread among plants or slopes where sun's rays are most direct, will receive a higher relevance and will be better analyzed (heavy information); such an approach should reduce the amount of used water and pesticides, and promote sustainable and rational agriculture.

To model the situation, we construct a complete node- and edge-weighted graph $G = (V, E, q, b^w, b^s, p)$, where:

$q : E \to \mathbb{R}^+$ is an edge-weight: $q(\{v_i, v_j\}) = q_{ij}$ represents the battery consumption required by the movement of the UAV along edge $\{v_i, v_j\}$;

$b^w : V^+ \to \mathbb{R}^+$ and $b^s : V^+ \to \mathbb{R}^+$ are node-weights: $b^w(v_i) = b^w_i$ and $b^s(v_i) = b^s_i$ represent respectively the battery cost to acquire light (weak coverage) and heavy (strong coverage) information from v_i, if $i > 0$, and are null if $i = 0$;

Finally, $p : V^+ \to \mathbb{R}^+$ is a node-weight representing the profit; hence, $p(v_i) = p_i$ is equal to the relevance for the sites and is null for the depot.

Given the set of sites V and a fleet of UAVs U all positioned at the depot, our problem consists of determining a cycle for each UAV such that it has enough battery to complete the traversal of its cycle, in the meantime weakly covering *all* sites and strongly covering some of them.

Given a solution Sol of this problem, by extension, we denote by $p(Sol)$ its profit, that is, the sum of the profits associated with the

[1] See *e.g.* https://www.dji.com/it/mini-3-pro/specs.

Table 1
Table of symbols used in this paper.

Symbol	Description
$V = \{v_1, \dots, v_n\}$	Set of sites
q_{ij}	Battery cost to fly between sites v_i and v_j
b_i^w	Battery cost to weakly cover site v_i
b_i^s	Battery cost to strongly cover site v_i
p_i	Profit gained by the fleet whenever v_i is strongly covered by a UAV
d_{MAX}	Largest distance between two sites
v_0	Depot
V^+	$V \cup \{v_0\}$
γ_i	No. of sites located within a radius R from node v_i
$U = \{u_1, \dots, u_m\}$	Set of UAVs
B	Battery given to each UAV
R	Radius for weak coverage

only sites heavily covered. The final aim is choosing the solution of maximum profit among all the feasible ones.

We conclude this section by observing that we implicitly assume that fleet U is equipped to guarantee the feasibility of the problem; in other words, we assume the batteries of the UAVs are powerful enough to guarantee that at least the weak coverage is always possible (see Table 1).

3. m3DIP vs. some known related problems

In this section, we review the literature of some very well-studied problems that have similitudes with m3DIP, for most of them we try to propose how they could be extended in order to define light and heavy information, and we clarify why their known solutions cannot be exploited in our case. All the definitions will be rephrased according to the terminology used in this paper.

TOP

There are m hikers initially located at v_0, and each one gains a profit $p(v_i)$ if visits a still unvisited site v_i. The hikers must complete their tour within a predetermined time B. So, the *Team Orienteering Problem* (TOP) (Chao et al., 1996) consists in determining a set of m cycles, each passing through v_0 and respecting the time constraint such that each node is visited at most once and the total profit collected is maximized.

It is clear that, in general, TOP omits to visit some nodes due to the B constraint, and can be considered as a special case of m3DIP, when $R = \infty$, that trivially guarantees the weak coverage.

Variants of TSP

In the *multiple Traveling Salesperson Problem* (mTSP) (Gorenstein, 1970), m salespersons leave their base station v_0 and have to visit n sites most cheaply, so m cycles must be found such that all nodes are included in at least one cycle, and the goal is to keep the overall traveled distance as low as possible. They could have to simply visit all the customers (light information) but have some longer meetings with a few of them (heavy information).

The *Close Enough m-Traveling Salesperson Problem* (CEmTSP) (Gulczynski et al., 2006) is a variant of mTSP, where the salespersons do not need to visit the exact location of each site. Instead, for each of them, a region of the plane containing it and possibly other sites considered as its neighborhood set is specified, and the goal is to find m cycles, all starting from v_0 and intersecting all of these neighborhood sets at least once in the shortest possible overall traveling distance.

CEmTSP and m3DIP are somehow similar if all p_is are null, but even in this case, a solution for m3DIP would be a feasible solution for CEmTSP but not necessarily the optimum one, because CEmTSP requires minimizing the traveled distance (*i.e.*, the overall battery consumption) instead of the completion time.

It is worth noting that some researchers solving this problem introduce some additional dummy points to better cover all sites. We also adopted this strategy, which will be discussed in Section 7.

A problem closer to ours is the generalized CETSP recently introduced by Di Placido et al. (2023). In that paper, the authors address a variant of the CEmTSP, where each customer is associated with a set of disks with different radii and rewards. If the disk dimensions are two, they could be exploited to define light and heavy information. The reward collected corresponds to the one related to the innermost circle traversed. The objective is to maximize the difference between the total collected reward and the route length. In contrast, our problem differs from theirs in several aspects: we have a team of UAVs instead of a single one; we do not need to minimize the route length, and we have a battery constraint that limits the flight duration of each UAV.

The *Prize Collecting TSP* (Balas, 1989) is another well-known extension of the TSP in which the goal is to find the optimal tour through a set of sites, each associated with a profit, which maximizes the difference between the collected profit and the traveling cost (prize). These prizes could be categorized as light or heavy based on the level of effort or the time needed to collect them. It has many applications in different fields, and it has also been extended to the multi-vehicle case. The main difference with m3DIP is that, whereas the Prize Collecting TSP actually considers the travel cost in its objective function, in our problem, the travel cost, expressed in terms of battery usage, impacts only the feasibility of the routes. In addition, we consider two types of profits whose one is the possibility of collecting the light information passing in the nearby of a site without explicitly visiting it.

Coverage Problems

When solving the *Sweep Coverage Problem* (SCP) (Li et al., 2011), only periodic patrol inspections are sufficient for a certain set of sites instead of continuous monitoring, like in traditional coverage; in particular, a site is said to be *t-sweep covered* whenever at least one UAV visits the site within every t time period, and t is an input parameter.

A variant of SCP is the *Cooperative Sweep Coverage Problem* (CSCP) (Gao et al., 2020), which allows the deployment of multiple UAVs on the same trajectory to further reduce the sweep period or detection delay. UAVS could need to guarantee complete periodic patrolling (light information) and check for some more accurate (heavy) information within longer time intervals.

The objectives are to minimize either the number of necessary UAVs to guarantee t-sweep coverage for all the sites or the maximum sweep period t given the number of UAVs. The different objective function makes SCP and m3DIP rather different.

The *Maximum Coverage Problem* (MCP) consists of selecting k sites from a set in order to maximize the portion of demand covered, where a site is considered covered if at least one node in its covering radius is selected (Downs and Camm, 1996). A well-established version of MCP is the Maximum Coverage Location Problem (MCLP) (Berman and Krass, 2002), in which the goal is to locate k facilities in order to maximize the portion of demand covered. Both problems could be extended by considering light and heavy coverage, with different covering radii. The goal would become to offer a light coverage to all the sites while maximizing the portion of demand covered also by the heavy service. This could find application, for example, in telecommunications, where the light service could be represented by 2G coverage, which has a broader communication range, while the heavy service is exemplified by the faster data rates and lower latency of a 5G network but with a shorter communication range and a greater cost. This problem does not have vehicles moving around but the concepts of light and heavy information appear very naturally.

Routing Problems

The *Vehicle Routing Problem* (VRP) (Braekers et al., 2016) is one of the most studied combinatorial optimization problems. Its goal is to visit a set of customers, starting from a depot, with a set of homogeneous vehicles, in such a way that the total traveled distance (or cost) is minimized. A classical extension is the Capacitated VRP (CVRP), in which vehicles have a maximum loading capacity, and each customer is associated with a demand. Vehicles could collect light information when passing near a customer (*e.g.*, visual check on the state) and heavy information when stopping at a customer (*e.g.*, performing an equipment check or maintenance). Despite some similarities, in m3DIP the goal is to maximize the total collected profit instead of minimizing the total traveled distance.

The *Capacitated Arc Routing Problem* (CARP) differs from the CVRP in the demand, associated with arcs rather than nodes (Golden and Wong, 1981). The different kinds of visits could differ from the type of service supplied by visiting the arc. For instance, in waste collection, a light service could correspond to just emptying the waste bins along certain streets, while a heavy one to clean the bins and the streets themselves. While the first must be performed, the second one could be optional and generate a profit. Our problem is rather different with respect to this one for a twofold reason: first, we do not give any importance to the arcs, and secondly, in the CARP it is not possible to get light information passing simply close to a site but it is anyway necessary to visit the sites.

WRP

The *Watchperson Route Problem* (WRP) is a well-known problem in computational geometry (Ntafos, 1992). The goal is to define the shortest tour for the watchperson within a polygonal area in order to be able to watch at least once every point in the area from a maximum distance of d. A common application lies in the design of routes for surveillants in a closed area, such as, for instance, a museum. The concept of light and heavy services could be applied also in this case. The latter would represent a deeper inspection from a smaller distance. The goal could become to maximize the area covered with a heavy service, in a given amount of time, while ensuring a total light coverage for the area. Clearly, this problem behaves differently from m3DIP in view of the different objective function and the absence of battery constraints.

4. Mathematical model

In this section, we express m3DIP as a mixed integer linear programming problem arc-based model (AM).

We define the following binary decision variables:

- x_{ij} assuming value equal to 1 if site v_j is visited immediately after site v_i, and 0 otherwise;
- w_{ij} assuming value equal to 1 if site v_i is weakly covered by site v_j, and 0 otherwise;
- s_i assuming value equal to 1 if site v_i is strongly covered, and 0 otherwise;
- Q_j is the residual energy of the UAV while flying over site v_j. By definition, $Q_0 = B$.

These variables are subject to the following constraints:

$$\max \sum_{v_i \in V} p_i s_i \tag{1}$$

$$\sum_{v_j \in V^+} x_{ij} = \sum_{j \in V^*} x_{ji} \quad \forall i \in V^+ \tag{2}$$

$$\sum_{v_j \in V^+} x_{ij} \geq s_i \quad \forall v_i \in V \tag{3}$$

$$\sum_{v_j \in V} w_{ji} \leq \sum_{v_j \in V^+} \gamma_i \, x_{ji} \quad \forall v_i \in V \tag{4}$$

$$\sum_{v_j \in V} x_{0j} \leq h \tag{5}$$

$$\sum_{v_j \in V} w_{ij} = 1 \quad \forall v_i \in V \tag{6}$$

$$w_{ij} \leq 1 - \frac{d_{ij} - R}{d_{MAX}} \quad \forall v_i \in V \;\; \forall v_j \in V \tag{7}$$

$$Q_j \leq Q_i - q_{ij} \, x_{ij} - b_i^s \, s_i - \sum_{v_l \in V} b_l^w \, w_{li} + B(1 - x_{ij}) \quad \forall v_i \in V \;\; \forall v_j \in V \tag{8}$$

$$b_i^s \, s_i + \sum_{v_l \in V} b_l^w \, w_{li} + q_{i0} \leq Q_i \leq B \quad \forall v_i \in V \tag{9}$$

$$Q_0 = B \tag{10}$$

The goal of the problem is to maximize the total collected profit as expressed in (1). Constraints (2) ensure route continuity imposing that the number of existing arcs must equal the number of entering arcs for all the nodes in the network. A site can be strongly covered and can weakly cover other sites only if it is visited, as stated by constraints (3) and (4), respectively. The number of UAVs used cannot exceed the number of available UAVs or, equivalently, the number of produced cycles is at most equal to the number of UAVs, as expressed in Constraints (5). Each site must be weakly covered, as imposed by Constraints (6). Constraints (7) imply that a site can be weakly covered only by sites located within a radius R from it. Constraints (8) allow keeping a trace of the currently available battery when reaching a site. At each site, the battery level must be always sufficient to allow the eventual strong and weak planned coverage, as imposed by Constraints (9). Finally, constraints (10) set the available battery at the starting depot equal to the prefixed value, respectively. All the decision variables are binary.

In the following, we call *AM-solver* an algorithm exactly solving the problem exploiting the above AM based model.

5. Matheuristic $\mathcal{MH}$

Unfortunately, the AM-solver cannot provide a solution in a reasonable time, even for very small instances of our problem. So, here we provide a matheuristic that we call $\mathcal{MH}$, to distinguish from other heuristics that will be described in the following.

$\mathcal{MH}$ consists on solving a path-based model (PM) of m3DIP, in which variables represent complete cycles. Such a model provides a solution if fed with all the feasible cycles. Since their number grows with the number of sites n following a factorial law, the problem cannot be solved with the current state-of-the-art solvers, even for small instances.

The idea is to give the PM only a subset of cycles, all passing through the depot, so that the search space is reduced only to this subset instead of the whole set of all possible cycles.

To highlight this behavior, we show the results of an experiment on a single random instance with 20 sites, showing how the execution times and the objective function vary when the number of cycles increases. From it, it clearly appears that while the time quickly grows with the number of cycles (see Fig. 1(a)), the optimal solution is approached already with a small number of cycles in input (see Fig. 1(b)). This justifies our choice of consistently reducing the number of cycles among which choosing the solution.

Each cycle, which is defined as an ordered sequence of sites starting and ending in v_0, will be passed in input as a characteristic vector (it traverses site v_j if and only if its characteristic vector has a 1 in correspondence of index j) plus a value indicating which is the traversed distance that depends on the order in which the traversed sites are visited.

While selecting the cycles that will enter the solution, among the ones given in input, the $\mathcal{MH}$ decides which sites traversed by these cycles are weakly and strongly covered, always respecting the battery constraint B, paying attention to weakly cover all sites and trying to keep as higher as possible the profit.

Since the output solution is optimum when limited to the given set of cycles, the quality of the solution will strongly depend on how we select the input set of cycles.

Here, we detail how the matheuristic is designed and postpone to the next section the long discussion on selecting suitable cycles.

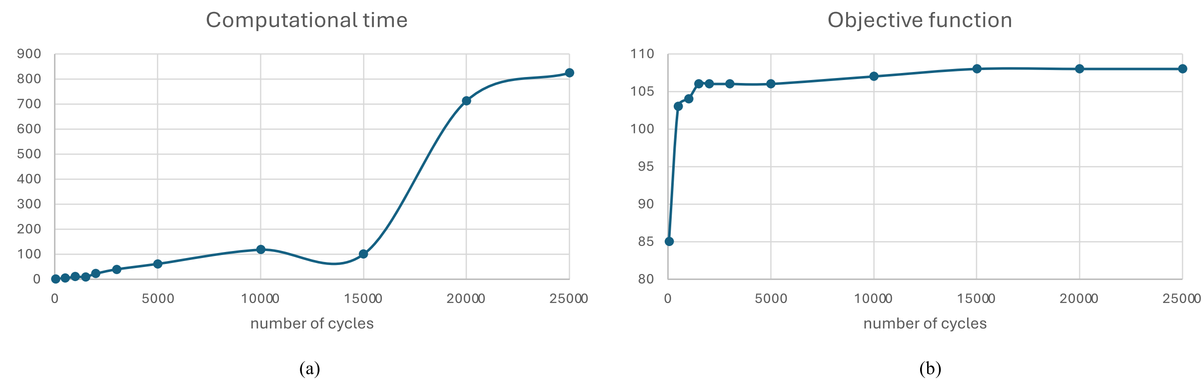

Fig. 1. Variation of: (a) computational time and (b) objective function with the number of cycles given in input to $\mathcal{MH}$.

So, we assume we have already generated a set C of $c = |C|$ cycles with the heuristics described in Section 6, we pass them to the following algorithm which assembles the optimal solution obtainable as the extraction of m cycles among the given c ones. We recall that a cycle is only identified by the subset of sites visited, while the PM takes decisions on strong and weak coverage.

We define the following binary decision variables, which are used in the PM.

- c_k: assuming value equal to 1 if cycle k is selected and 0 otherwise;
- W_{ik}: assuming value equal to 1 if site v_i is weakly covered by cycle k and 0 otherwise;
- S_{ik}: assuming value equal to 1 if site v_i is strongly covered by cycle k and 0 otherwise

We also introduce the following constants.

- α_{ik}: equal to 1 if site v_i belongs to cycle k (*i.e.*, if v_i can be strongly covered by cycle k) and 0 otherwise;
- β_{ik}: equal to 1 if site v_i is located within the weakly coverage radius of a node visited by cycle k (*i.e.*, if v_i can be weakly covered by cycle k, even if it possibly do not belong to it) and 0 otherwise;
- $\hat{B}_k$: residual battery to dedicate for coverage operations for cycle k; it is equal to B decreased by the energy necessary to fly over the whole cycle, that is the battery amount that can be spent for weak and strong coverage.

The mathematical PM can be formulated as follows:

$$\max \sum_{k\in C} \sum_{v_i \in V} p_i S_{ik} \tag{11}$$

$$S_{ik} \le \alpha_{ik}\, c_k \quad \forall\, v_i \in V \;\; \forall k \in C \tag{12}$$

$$W_{ik} \le \beta_{ik}\, c_k \quad \forall\, v_i \in V \;\; \forall k \in C \tag{13}$$

$$\sum_{v_i \in V} b_i^s\, S_{ik} + \sum_{v_i \in V} b_i^w\, W_{ik} \le \hat{B}_k \quad \forall\, k \in C \tag{14}$$

$$\sum_{k\in C} W_{ik} = 1 \quad \forall v_i \in V^+ \tag{15}$$

$$\sum_{k\in C} S_{ik} \le 1 \quad \forall v_i \in V^+ \tag{16}$$

$$\sum_{k\in C} c_k \le h \tag{17}$$

The objective function consists of the maximization of the reward collected by strongly covering the nodes, as expressed in (11). A node can be strongly covered by a cycle only if this cycle has been selected and passes through this node, as stated by constraints (12). Constraints (13) impose that a site can be weakly covered by a cycle only if this cycle has been selected and the site is located within the coverage radius of at least one of the nodes in the cycle. The quantity of battery spent in weak and strong coverage operations cannot exceed the quantity available, as imposed by constraints (14). Every site must be weakly covered by exactly one cycle (constraints (15)) and can be strongly covered by at most one cycle (constraints (16)). Finally, constraints (17) impose that at most h cycles are used, one for each UAV.

6. Generation of input cycles and benchmark heuristics

In this section, we describe some heuristics, all obtained as simple modifications of algorithms originally designed to solve problems different from m3DIP, although with something in common with it. These heuristics serve a twofold purpose: on the one hand, they generate the set C of cycles given in input to $\mathcal{MH}$; on the other hand, they can be seen as benchmark heuristics, revealing that $\mathcal{MH}$ largely outperforms each of them. In each heuristic we introduce a degree of randomness, in order to run them many times and to acquire a large number of cycles; on the other hand, we compared the best profit computed over all the runnings with the profit achieved avoiding randomness, and they coincide.

Preliminarily, observe that, in order to construct a suitable set of cycles to be given in input to $\mathcal{MH}$, we would like to choose cycles with different characteristics; *e.g.*, some of them could pass through many sites with the aim of better contributing to the weak coverage, while others could pass through sites with a high profit, in order to improve the value of the profit of the whole solution.

In Section 3, we have already discussed that the solutions to the two problems TOP and CEmTSP address only a part of the objectives addressed by m3DIP. Namely, TOP is focused on strong coverage (and neither guarantees a feasible solution to m3DIP), while CEmTSP is on weak one. So, we cannot exploit the solutions of these two problems to deduce good solutions to m3DIP; nevertheless, the cycles constituting their solutions seem very good candidates to contribute to generating cycles to be given in input to $\mathcal{MH}$.

Heuristic $\mathcal{H}_{gTOP}$

Unfortunately, TOP is NP-hard and APX-hard (Blum et al., 2007). So, there is a wide literature solving the problem either optimally or with good approximations (see *e.g.* the survey Gavalas et al., 2014). Nevertheless, even these latter algorithms require very long times, though theoretically polynomial. For this reason, we run a simple greedy heuristic to handle TOP, which we call $\mathcal{H}_{gTOP}$.

More in detail, $\mathcal{H}_{gTOP}$ generates m cycles one by one by adding one node at a time, at each step randomly selecting one node among the three ones with the highest ratio between the profit of that node and its cost, given as the sum of the battery consumption to reach it plus the battery cost to strongly cover it.

It is worth noting that in the literature there is a very simple and fast heuristic for TOP (Vansteenwegen et al., 2009); nevertheless, we

Table 2
Average weak coverage percentage.

Battery B	7.5 MJ			1 MJ			1.5 MJ		
No. of nodes n	20	50	70	20	50	70	20	50	70
$\mathcal{MH}$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$\mathcal{H}_{gTOP}$	82.9	68.8	67.0	94.2	83.6	81.8	100.0	96.7	95.4
$\mathcal{H}_{gTOP+10\%}$	87.2	74.6	71.3	97.4	88.0	86.0	100.0	98.0	97.4
$\mathcal{H}_{gTOP+20\%}$	90.6	79.4	77.4	99.6	90.8	89.3	100.0	99.4	98.3
$\mathcal{H}_{gTOP+30\%}$	93.6	83.4	80.8	100.0	93.8	92.6	100.0	99.8	98.9
$\mathcal{H}_{gTOP+40\%}$	96.1	86.0	84.1	100.0	95.8	93.8	100.0	99.9	99.3
$\mathcal{H}_{gTOP+50\%}$	97.9	89.4	88.3	100.0	96.7	95.4	100.0	100.0	99.6
$\mathcal{H}_{greedy}$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$\mathcal{H}_{TSPN}$	98.4	95.4	91.6	100.0	99.9	99.1	100.0	100.0	100.0
$\mathcal{H}_{TSPN-10\%}$	96.7	90.2	87.1	99.6	98.8	97.2	100.0	100.0	100.0
$\mathcal{H}_{TSPN-20\%}$	90.4	80.8	77.9	98.8	96.4	94.0	100.0	100.0	100.0
$\mathcal{H}_{TSPN-30\%}$	82.8	70.2	67.6	96.2	90.8	88.9	100.0	100.0	99.5
$\mathcal{H}_{TSPN-40\%}$	70.5	58.8	54.9	90.4	80.8	77.9	99.6	98.8	97.2
$\mathcal{H}_{TSPN-50\%}$	62.7	51.0	47.1	79.0	65.6	63.0	98.4	95.4	91.6
$\mathcal{H}_{\alpha\beta}$	87.6	82.2	80.3	98.3	95.9	95.2	100.0	100.0	100.0
$\mathcal{H}_{01}$	89.9	83.6	81.0	99.2	97.0	96.3	100.0	100.0	100.0
$\mathcal{H}_{\frac{1}{4}\frac{3}{4}}$	87.6	80.6	79.6	97.9	94.3	94.2	100.0	99.9	99.8
$\mathcal{H}_{\frac{1}{2}\frac{1}{2}}$	84.3	75.6	75.5	96.6	90.4	90.0	100.0	98.4	99.2
$\mathcal{H}_{\frac{3}{4}\frac{1}{4}}$	81.4	71.0	69.4	94.6	86.4	85.2	100.0	97.2	96.9
$\mathcal{H}_{10}$	83.4	67.4	67.7	94.2	84.2	81.0	100.0	96.9	95.7

experimentally compared it with $\mathcal{H}_{gTOP}$ verifying that the latter one guarantees better profits; for this reason, we adopt the greedy approach to solve TOP.

Of course, any solution output by $\mathcal{H}_{gTOP}$ is in general not feasible for m3DIP, because it does not guarantee weak coverage of all sites (see Table 2).

Heuristics $\mathcal{H}_{greedy}$ and $\mathcal{H}_{TSPN}$

For what concerns CEmTSP, it is clearly NP-hard as not easier than TSP (Miller, 2013), so we determine feasible solutions by means of two different heuristics. Although CEmTSP does not require respect for any battery constraint, our heuristics do, in order to guarantee the cycles' feasibility.

The first heuristic is inspired by an algorithm in Calamoneri et al. (2022) that solves a different problem but involves anyway UAVs with a battery constraint, so we adapt it to this context. It follows a greedy approach, and hence it is denoted $\mathcal{H}_{greedy}$.

$\mathcal{H}_{greedy}$ constructs all the m cycles at the same time; at each step, considers one by one all the m current portions of cycles and, for each of them, starting from the node selected last v_{last} (at the beginning v_0), selects the next one, v_{next}, randomly choosing among the three sites with the highest ratio between the number of sites weakly covered by v_{next} and the sum given by the cost to reach v_{next} plus the cost of weak covering all these nodes. Clearly, we have to guarantee that the whole cycle with the addition of v_{next} can be flown over within battery B.

Note that the solutions output by $\mathcal{H}_{greedy}$ are feasible for m3DIP, because they guarantee weak coverage of all sites within battery B, as shown in Table 2.

The second heuristic is inspired by an algorithm called m-TSPN (multiple-Traveling Salesman Problem with Neighborhood) (Kim et al., 2014, 2017) and hence denoted $\mathcal{H}_{TSPN}$. The original is an approximation algorithm with a provable constant approximation ratio. Nevertheless, it provides no battery constraints, so we need to modify the algorithm to let it work correctly in our setting. Unfortunately, in this way, the algorithm loses the guarantee of complete weak coverage, although it is reached in most of the cases (see Table 2).

$\mathcal{H}_{TSPN}$ gives as part of the input m sites (one for each UAV) and uses them as children of the root of a minimum spanning tree rooted at v_0; the m sub-trees of this minimum spanning tree are then transformed, using the Christofides's approximation algorithm for TSP (Christofides, 1976), into m cycles covering all sites and intersecting only at the root; finally, some operations are executed in order to equalize the weight.

Heuristics $\mathcal{H}_{gTOP+x\%}$ and $\mathcal{H}_{TSPN-y\%}$

The reason why we use approaches tackling different problems is that the cycles they produce have diverse properties and hence are somehow complementary. In particular, as far as the two problems TOP and CEmTSP are defined, $\mathcal{H}_{gTOP}$ tends to produce cycles very focused on strong coverage (and hence containing few sites with high profit), while $\mathcal{H}_{greedy}$ and $\mathcal{H}_{TSPN}$ produce cycles very focused on weak coverage (and hence passing through as many sites as possible).

In this way, the produced cycles are very suitable to address only either weak or strong coverage, but not both together. So, besides the previously described algorithms, we propose some modifications that attenuate the main objective of each approach (either weak or strong coverage) to introduce also the other one.

Namely, concerning $\mathcal{H}_{gTOP}$, we propose a simple variant: we run the same algorithm with an increased battery $B + x\%$ instead of simply B, for certain fixed values of x; in this way, we obtain cycles that contain, in general, more sites than any solution of the original algorithm; then, we run a post-processing phase in which we keep only the cycles that can be flown over within battery B guaranteeing the largest possible weak coverage, but possibly renouncing to the strong coverage of some sites. We call this heuristic $\mathcal{H}_{gTOP+x\%}$ and, as expected, it increases the weak coverage (see Table 2), at the expense of the strong coverage (*i.e.*, of the profit), as highlighted in Table 3.

For $\mathcal{H}_{TSPN}$, we propose a symmetric variant: we run the same algorithm with a decreased battery $B - y\%$ instead of simply B, for certain fixed values of y; in this way, we obtain cycles that contain, in general, fewer sites to be weakly covered (as shown in Table 2) but a higher residual battery; this can be exploited by a post-processing phase to strongly cover some sites more (see Table 3). We call this heuristic $\mathcal{H}_{TSPN-y\%}$, and we expect that it increases the strong coverage, possibly at the expense of the weak coverage.

Heuristic $\mathcal{H}_{\alpha\beta}$

Here we introduce a new simple greedy heuristic addressing from the beginning both weak and strong coverage. Calling $\gamma'(v_i)$ the set of sites located within a radius R from v_i that have not been weakly covered by any other site, this heuristic chooses the next site v_{next} to be included in the current cycle after the last one v_{last} as the one maximizing the following value:

$$\frac{\alpha \cdot p(v_{next}) + \beta \cdot |\gamma'(v_{next})|}{q_{lastnext} + b^s_{next} + \sum_{v \in \gamma'(v_{next})} b^w_i}$$

where α and β are two parameters such that $0 \leq \alpha, \beta \leq 1$ and $\beta = 1 - \alpha$.

Clearly, varying the values of α and β we expect to give a different importance to the two kinds of coverage. Indeed, a large value of α corresponds to favoring strong coverage, while a large value of β implies strengthening the requirement of weak coverage. This is confirmed by Tables 2 and 3.

It is worth noting that the heuristic corresponding to $\alpha = 1$ is different from $\mathcal{H}_{gTOP}$, although following the same philosophy, because the function optimized by the greedy approach includes anyway information about weak coverage. Analogously, the heuristic corresponding to $\beta = 1$ follows the same approach as the ones solving CEmTSP although is different from both of them; in particular, due to the optimized function, it chooses sites even with respect to their possible strong coverage, while $\mathcal{H}_{greedy}$ and $\mathcal{H}_{TSPN}$ base their choices only on weak coverage. In all three heuristics, it is decided by a post-processing phase (a heuristic for the knapsack problem) which sites to strongly cover.

In the following, we consider some variants of $\mathcal{H}_{\alpha\beta}$. Namely: when α and β are not given, they are randomly chosen, and many iterations are executed on the same instance, saving as profit the best value over all their runs; then, some special values for α and β are given, that are $\alpha = 0$ and $\beta = 1$; $\alpha = 1/4$ and $\beta = 3/4$; $\alpha = \beta = 1/2$; $\alpha = 3/4$ and $\beta = 1/4$; $\alpha = 1$ and $\beta = 0$.

7. Experimental results

In this section, we run some experiments, setting all the parameters to some realistic values, as follows:

- battery B: we perform three series of experiments, with battery 0.75 MJ, 1 MJ and 1, 5 MJ, respectively, in agreement with some UAVs sold at moment (*e.g.*, 0.75 MJ is close to the battery of DJI Matrice 200 while 1.5 MJ is about the single battery of DJI Matrice 300[2]); in this way, we aim to understand whether $\mathcal{MH}$ and the other heuristics change their behavior when UAVs have poor, medium, and high batteries.
- UAV energy consumption for traveling and hovering can vary significantly based on several factors, including the model, payload, and environmental conditions such as weather, and particularly wind speeds. We chose the following values, commonly utilized in the literature, *e.g.*, Khochare et al. (2021), Sorbelli et al. (2024), and aligned with the ones found in recent publications such as Alyassi et al. (2022) and Baek et al. (2019), where rigorous experiments to measure drone energy consumption were conducted:
  - unit battery cost q': 200 J/m (Joule *per* meter); hence $q_{ij} = q' \cdot dist(v_i, v_j)$, where $dist$ represents the distance in meters between v_i and v_j;
  - hovering cost h of 700 J/s (Joule *per* second); we set b_i^s as h multiplied by a random number of seconds ranging between 100 and 300; instead, b^w is set for all sites to 700 J;
- R is set to 150 m;
- the area of interest is 1 km × 1 km, and in the same dimension we randomly allocate sites; we consider three different values of their number n: 20, 50, and 70, in order to see how the matheuristic may change its behavior when sites are more or less densely distributed.

All our experiments have been performed on a computer equipped with an Intel(R) Core(TM) i5-1135G7 CPU (8 cores clocked at 2.4 GHz) and 16 GB RAM. All simulation runs have been repeated 10 times with different seeds, the tables show mean values. For each simulation, to generate the cycles, we run each heuristic 100 times with different seeds.

With the aim of allowing people the reproducibility of the experiments, we provide the code of the matheuristics and the instances here: github.com/ulisse91/LightHeavyDataCollectionCycleGen, while the solvers for AM and PM straightforwardly follow from the constraints in Sections 4 and 5.

In order to analyze the performance of the considered heuristics, we compare different measures.

First, we compare the weak coverage reached by each heuristic (see Table 2). As already discussed in Section 6, only our matheuristic and $\mathcal{H}_{greedy}$ guarantee full weak coverage; heuristics $\mathcal{H}_{TSPN}$, originally solving a problem very similar to CEmTSP, and $\mathcal{H}_{01}$, completely focused on the weak coverage, are very close to complete weak coverage; also $\mathcal{H}_{gTOP+50\%}$ achieves a very good result w.r.t. this parameter because the battery is artificially increased by a quantity large enough to guarantee almost total weak coverage at the expense of strong coverage. Finally, clearly, when the battery is very large (1.5 MJ), almost all the heuristics are able to achieve full weak coverage, so they produce a feasible solution for m3DIP and can be compared w.r.t. the profit they gain.

Second, we compare the profit achieved by every heuristic (see Table 3).

Note that it is meaningful to compare the results of $\mathcal{MH}$ only with those of the heuristics that are able to guarantee full weak coverage (and that are highlighted with white background in Table 3). In this way, it is clear that our matheuristic highly outperforms $\mathcal{H}_{greedy}$ and all the other heuristics, whenever they are able to reach full weak coverage.

Besides the performance of the heuristics, we studied the distribution of the cycles selected by $\mathcal{MH}$. More precisely, we checked whether there are some heuristics that contribute less to the solution provided by $\mathcal{MH}$, in the sense that their cycles are rarely chosen.

Preliminarily, it must be said that a few cycles are generated exactly the same by more than one heuristic; in this case, we keep only one copy but store the different heuristics that generated them. (The number of these cycles largely changes with n and B.)

In Table 4, an "x" on the row corresponding to heuristic $\mathcal{H}$ means that at least one cycle generated exclusively by $\mathcal{H}$ has been selected by the matheuristic, somehow meaning that excluding $\mathcal{H}$ would damage the performance of the matheuristic. Vice versa, a "o" means that at least one cycle generated by $\mathcal{H}$ was selected by the matheuristic, but it was also generated by another heuristic, and no other cycles produced by $\mathcal{H}$ have been exploited. More in detail, we assign an "x" to the heuristic with the largest number of used cycles, remove the heuristic and all its cycles, and repeat.

It turns out that $\mathcal{H}_{TSPN}$ and all its variants do not seem very useful to $\mathcal{MH}$. Nevertheless, we decided to keep them, because $\mathcal{H}_{TSPN}$ is not a heuristic designed by us, but it derives from a well-established algorithm in the literature.

7.1. Adding grid-points to improve the solution of m3DIP

In agreement with certain works dealing with CEmTSP (*e.g.*, Carrabs et al., 2017; Di Placido et al., 2023) we now take into account the possibility of weakly covering a site from a point that does not belong to the set of sites but is simply close to it; so, in our model, we introduce some additional dummy points that may be close to more than one site and can be exploited to collect light information from all its neighborhood. Clearly, these dummy nodes do not need to be covered (neither weakly nor strongly), and hence the value of functions b^w and b^s is null on them. For the sake of simplicity, in this setting, we keep the same notation, just extending its meaning. So, with V^+ we now mean the set of sites plus the depot and the newly introduced dummy nodes; moreover, by γ_i we intend the number of sites that are located within a radius R from i (where i is either a site or a dummy node) and by q_{ij} the battery cost of flying between i and j, where i and j are now either sites or dummy nodes.

As dummy nodes, we introduce points at integer coordinates, *i.e.* the cross points of a grid. We execute experiments considering grids of different dimensions: the unit is set to $\frac{R}{4}$, $\frac{R}{2}$, $\frac{3}{4}R$, and R (see Table 5),

[2] See https://www.dji.com/it/matrice-200-series-v2 and https://www.aprflytech.it/dji-matrice-300-rtk.

Table 3
Average profit due to the strong coverage. The values corresponding to feasible solutions (*i.e.*, guaranteeing 100% weak coverage) are highlighted with white background; the performance of $\mathcal{MH}$ can be compared only with these values.

Battery B	7.5 MJ			1 MJ			1.5 MJ		
No. of nodes n	20	50	70	20	50	70	20	50	70
$\mathcal{MH}$	63.6	59.1	65.6	92	112.4	132.6	103.5	195.8	221.9
$\mathcal{H}_{gTOP}$	66.8	97.1	106.8	88.1	134.1	148.7	103.5	195.6	220.4
$\mathcal{H}_{gTOP+10\%}$	65.3	95.0	102.3	86.4	129.3	142.9	103.5	192.6	217.3
$\mathcal{H}_{gTOP+20\%}$	62.9	91.1	100.2	86.0	129.7	141.5	103.5	187.8	215.3
$\mathcal{H}_{gTOP+30\%}$	62.1	88.2	97.2	83.4	124.6	136.3	103.5	182.6	209.0
$\mathcal{H}_{gTOP+40\%}$	57.8	84.4	91.5	81.6	119.2	131.8	103.5	177.0	204.0
$\mathcal{H}_{gTOP+50\%}$	56.5	80.9	87.5	80.7	114.0	127.4	103.5	167.3	198.0
$\mathcal{H}_{greedy}$	50.9	55.9	52.5	68.0	88.0	93.2	72.2	117.6	128.2
$\mathcal{H}_{TSPN}$	42.7	49.3	49.8	50.2	60.7	60.9	59.6	78.3	78.9
$\mathcal{H}_{TSPN-10\%}$	50.3	64.1	60.1	56.8	79.1	80.4	66.3	103.5	108.8
$\mathcal{H}_{TSPN-20\%}$	55.1	66.6	70.9	63.8	90.1	94.3	74.0	128.3	132.3
$\mathcal{H}_{TSPN-30\%}$	54.1	66.5	68.1	69.3	97.8	99.6	80.4	143.7	148.4
$\mathcal{H}_{TSPN-40\%}$	49.2	64.2	65.0	68.7	95.4	100.2	85.4	151.6	162.1
$\mathcal{H}_{TSPN-50\%}$	40.9	54.1	57.0	59.4	76.4	84.5	88.7	147.2	153.9
$\mathcal{H}_{\alpha\beta}$	66.1	95.8	103.7	87.8	130.5	143.6	103.5	192.3	217.7
$\mathcal{H}_{01}$	53.8	71.3	74.1	69.2	93.0	100.0	72.3	114.8	130.5
$\mathcal{H}_{\frac{1}{4}\frac{3}{4}}$	64.8	88.1	92.9	86.2	123.3	131.9	103.5	183.9	207.2
$\mathcal{H}_{\frac{1}{2}\frac{1}{2}}$	66.3	95.1	103.0	87.9	130.1	143.4	103.5	192.5	216.1
$\mathcal{H}_{\frac{3}{4}\frac{1}{4}}$	66.3	97.3	107.1	88.4	134.4	147.8	103.5	195.4	220.0
$\mathcal{H}_{10}$	66.9	96.9	106.4	88.0	134.6	148.6	103.5	196.6	220.6

Table 4
Usage of cycles generated by heuristics: an "x" on the row corresponding to heuristic $\mathcal{H}$ means that at least one cycle generated exclusively by $\mathcal{H}$ has been selected by the matheuristic; a "o" means that $\mathcal{H}$ generated a cycle that was selected by the matheuristic, but another heuristic also generated it, and no other cycles produced by $\mathcal{H}$ have been exploited.

Battery B	7.5 MJ			1 MJ			1.5 MJ		
Number of nodes	20	50	70	20	50	70	20	50	70
$\mathcal{H}_{gTOP}$	o			o		x	x	x	x
$\mathcal{H}_{gTOP+10\%}$	o	o		o	x	x	o		x
$\mathcal{H}_{gTOP+20\%}$	x	x	x	x	x	x	x	x	
$\mathcal{H}_{gTOP+30\%}$	x	x		x	x	x	o	x	x
$\mathcal{H}_{gTOP+40\%}$	x	x	x	o	x	x	o		
$\mathcal{H}_{gTOP+50\%}$	x	x	x	x	x	x	o	x	x
$\mathcal{H}_{greedy}$	x	x	x	o	x	x	x		
$\mathcal{H}_{TSPN}$	x		x				x		
$\mathcal{H}_{TSPN-10\%}$	o		x				x		
$\mathcal{H}_{TSPN-20\%}$	o						x		
$\mathcal{H}_{TSPN-30\%}$							o		
$\mathcal{H}_{TSPN-40\%}$	o								
$\mathcal{H}_{TSPN-50\%}$			o						
$\mathcal{H}_{\alpha\beta}$	x			o	x	x		x	x
$\mathcal{H}_{01}$	o			o	x				
$\mathcal{H}_{\frac{1}{4}\frac{3}{4}}$	o			x		x	x	x	x
$\mathcal{H}_{\frac{1}{2}\frac{1}{2}}$	o	x		x	x		x		x
$\mathcal{H}_{\frac{3}{4}\frac{1}{4}}$	o	o	x	o		x	x	x	x
$\mathcal{H}_{10}$	o			o		o	o	x	x

in order to study possible differences in the performance of $\mathcal{MH}$. Of course, a grid unit larger than R is meaningless, and the larger the grid dimension, the smaller the number of grid points and vice-versa. We vary the battery dimension and the number of sites as in the previous experiments ($B = 7.5$ MJ, $B = 1$ MJ; $n = 20$, $n = 50$, and $n = 70$) in order to compare the resulting profits in the two settings (without and with dummy grid points). We did not perform experiments with the largest value of battery $B = 1.5$ MJ, since the battery level is so high that allows us to strongly cover almost all the sites, even without the insertion of grid-based additional points. Therefore, this set of interests is not relevant for this experiment.

Observe that adding grid points is always favorable for our problem, because:
- the lengths of the cycles are never longer, so saving battery;
- the cost of the overall weak coverage is the same, *i.e.*, $\sum_{v_i \in V} b_i^w$; so having more battery energy at disposal for strong coverage.

In Table 5, we focus only on the comparison of the unique heuristics able to always guarantee a feasible solution for m3DIP, that is $\mathcal{MH}$ and $\mathcal{H}_{greedy}$. In order to ease the reading of the table, we added the columns with $R = 0$ corresponding to the values without grid points (the same as in Table 3).

Looking at the results in general, it is very clear that, while $\mathcal{MH}$ takes advantage of the insertion of dummy grid nodes, instead $\mathcal{H}_{greedy}$ produces much worse solutions; the reason is that, as far as this heuristic works, it can only behave worse when grid point are added; indeed, first it aims at the weak coverage greedily trying to keep small the length of the produced cycles, so tending to choose grid points instead of sites; secondly it tries to strongly cover as many sites as possible with the residual battery, but it is not able to find enough of them because many grid points have been exploited.

It is worth noting that $\mathcal{MH}$ often takes less time to be executed when grid points are added (see Table 5); the reason is twofold:

Table 5
Experiments with grid-points. The values with $R = 0$ correspond to the experiments without grid-points (*i.e.*, the same as in Table 3).

Number of sites n	20					50					70				
Grid unit	0	$\frac{R}{4}$	$\frac{R}{2}$	$\frac{3}{4}R$	R	0	$\frac{R}{4}$	$\frac{R}{2}$	$\frac{3}{4}R$	R	0	$\frac{R}{4}$	$\frac{R}{2}$	$\frac{3}{4}R$	R
Battery $B = 7.5$ **MJ:**															
$\mathcal{MH}$	63.6	67.6	67.6	67	66.33	59.1	97.5	86.67	105.78	104.33	65.6	105.78	105.11	104.11	101.44
$\mathcal{H}_{greedy}$	50.9	6.8	16.5	26.2	32.1	55.9	20.7	29.9	28.4	39.4	52.5	20.4	32.9	52.5	31.6
Time of $\mathcal{MH}$ (s)	4.06	14.97	8.83	7.35	7.26	31.09	48.45	17.25	15.64	10.29	101.14	46.98	25.37	16.13	15.05
Battery $B = 1$ **MJ:**															
$\mathcal{MH}$	92	103.11	92.6	92.6	92.4	112.4	135.22	133.89	113.6	132.5	132.6	147.9	146.4	144.3	142.8
$\mathcal{H}_{greedy}$	68	6.8	14.4	29.4	32.2	88	22	35.4	33.8	48.5	93.2	22	41.2	78.7	44.8
Time of $\mathcal{MH}$ (s)	11	36.67	20.41	31.23	19.82	165.72	299.19	232.1	291.7	207.12	689.94	371.83	458.06	743.82	520.07

on the one hand, the larger number of nodes impacts the heuristics generating cycles (taking much more time), but not $\mathcal{MH}$; on the other hand, the most time-demanding issue faced by the matheuristic is to guarantee complete weak coverage, and it becomes easier in presence of dummy grid points. Note that we computed the running times of all the heuristics; nevertheless, we decided to report only the ones of $\mathcal{MH}$ because the times of the other ones always remain largely under 1 s

More in detail, when $B = 7.5$ MJ and n is either 20 or 70, $\mathcal{MH}$ reaches the best performance when the grid is finer (small values of the grid unit). Even the times are higher for the finer grid since there are many more points to be processed. The results are less regular when $n = 50$, but the trend is similar.

Even when $B = 1$ MJ the considerations for the profit are similar while it is more evident the gain in terms of time, especially for larger n.

8. Conclusions and future perspectives

In this paper, we introduced a novel problem, called m3DIP, based on a real-life situation involving a fleet of UAVs, consisting of guaranteeing complete weak coverage of a set of sites that, at the same time, maximizes the profit of a partial strong coverage. We expressed m3DIP as an integer linear programming problem and proposed a matheuristic $\mathcal{MH}$ that exploits several (mostly greedy) heuristics to give as an additional input an opportune subset of cycles so that the search space is reduced only to this subset instead of the whole set of all possible cycles. We showed the results of a large set of experiments, validating $\mathcal{MH}$ and showing that each heuristic exploited to produce a portion of input cycles alone is not able to achieve a good result but actively contributes with some cycles to the solution output by $\mathcal{MH}$; moreover, $\mathcal{MH}$ – that exploits cycles collected from all the heuristics – achieves the largely best profit. Then, we extended the setting with the possibility of using some dummy nodes for the weak coverage and showed that they are very useful for $\mathcal{MH}$ to get an even better solution.

We highlight two interesting open problems arising in the attempt to make the model more true to life.

First, cooperation among UAVs leads to autonomous decisions of the fleet, while the definition of our problem implicitly assumes that there is a central unit. Of course, introducing cooperation makes the model more powerful and interesting, but the problem is even more complicated.

Second, some papers (*e.g.*, Sorbelli et al., 2022) allow UAVs to fly at different heights from the floor, implying different radii for weak coverage; in particular, the higher the flight, the larger the radius. Nevertheless, if a UAV flies high, its ability to strongly cover sites decreases because precise information can be acquired only from very close. So, such a model would require balancing the advantage of augmenting the radius for better weak coverage and the disadvantage coming from the decreased ability to strongly cover. This would be a very challenging variant of our problem.

Finally, from a methodological point of view, it could be interesting to exploit the path-based model within a column generation framework, or in a branch-and-price algorithm, to provide an exact method to address m3DIP.

CRediT authorship contribution statement

Tiziana Calamoneri: Conceptualization, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing, Software. **Federico Corò:** Conceptualization, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Simona Mancini:** Conceptualization, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing, Software.

Data availability

Data will be made available on request.

References

Alyassi, R., Khonji, M., Karapetyan, A., Chau, S.C.-K., Elbassioni, K., Tseng, C.-M., 2022. Autonomous recharging and flight mission planning for battery-operated autonomous drones. IEEE Trans. Autom. Sci. Eng. 20 (2), 1034–1046.
Baek, D., Chen, Y., Bocca, A., Bottaccioli, L., Di Cataldo, S., Gatteschi, V., Pagliari, D.J., Patti, E., Urgese, G., Chang, N., et al., 2019. Battery-aware operation range estimation for terrestrial and aerial electric vehicles. IEEE Trans. Veh. Technol. 68 (6), 5471–5482.
Balas, E., 1989. The prize collecting traveling salesman problem. Networks 19 (6), 621–636.
Berman, O., Krass, D., 2002. The generalized maximal covering location problem. Comput. Oper. Res. 29 (6), 563–581.
Blum, A., Chawla, S., Karger, D.R., Lane, T., Meyerson, A., Minkoff, M., 2007. Approximation algorithms for orienteering and discounted-reward TSP. SIAM J. Comput. 37 (2), 653–670.
Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I., 2016. The vehicle routing problem: State of the art classification and review. Comput. Ind. Eng. 99, 300–313.
Calamoneri, T., Corò, F., Mancini, S., 2022. A realistic model to support rescue operations after an earthquake via UAVs. IEEE Access 10, 6109–6125. http://dx.doi.org/10.1109/ACCESS.2022.3141216.
Carrabs, F., Cerrone, C., Cerulli, R., Gaudioso, M., 2017. A novel discretization scheme for the close enough traveling salesman problem. Comput. Oper. Res. 78, 163–171.
Chao, I.-M., Golden, B.L., Wasil, E.A., 1996. The team orienteering problem. European J. Oper. Res. 88 (3), 464–474.
Christofides, N., 1976. Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. Technical Report, Carnegie-Mellon Univ..
Di Gennaro, S., Battiston, E., Di Marco, S., Facini, O., Matese, A., Nocentini, M., Palliotti, A., Mugnai, L., 2016. Unmanned Aerial Vehicle UAV-based remote sensing to monitor grapevine leaf stripe disease within a vineyard affected by esca complex. Phytopathol. Mediterr. 55 (2).
Di Placido, A., Archetti, C., Cerrone, C., Golden, B., 2023. The generalized close enough traveling salesman problem. European J. Oper. Res. 310 (3), 974–991.
Downs, B., Camm, J., 1996. An exact algorithm for the maximal covering problem. Nav. Res. Logstics 43 (3), 435–461.
Gao, X., Fan, J., Wu, F., Chen, G., 2020. Cooperative sweep coverage problem with mobile sensors. IEEE Trans. Mob. Comput. 21 (2), 480–494.
Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.E., 2014. A survey on algorithmic approaches for solving tourist trip design problems. J. Heuristics 20 (3), 291–328.
Golden, B., Wong, R., 1981. Capacitated arc routing problems. Networks 11 (3), 305–315.
Gorenstein, S., 1970. Printing press scheduling for multi-edition periodicals. Manage. Sci. 16 (6), B–373.
Gulczynski, D.J., Heath, J.W., Price, C.C., 2006. The close enough traveling salesman problem: A discussion of several heuristics. In: Perspectives in Operations Research: Papers in Honor of Saul Gass' 80 th Birthday. Springer, pp. 271–283.

Khochare, A., Simmhan, Y., Sorbelli, F.B., Das, S.K., 2021. Heuristic algorithms for co-scheduling of edge analytics and routes for uav fleet missions. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, pp. 1–10.

Kim, D., Uma, R.N., Abay, B.H., Wu, W., Wang, W., Tokuta, A.O., 2014. Minimum latency multiple data muletrajectory planning in wireless sensor networks. IEEE Trans. Mob. Comput. 13 (4), 838–851.

Kim, D., Xue, L., Li, D., Zhu, Y., Wang, W., Tokuta, A.O., 2017. On theoretical trajectory planning of multiple drones to minimize latency in search-and-reconnaissance operations. IEEE Trans. Mob. Comput. 16 (11), 3156–3166.

Li, M., Cheng, W., Liu, K., He, Y., Li, X., Liao, X., 2011. Sweep coverage with mobile sensors. IEEE Trans. Mob. Comput. 10 (11), 1534–1545. http://dx.doi.org/10.1109/TMC.2010.237.

Liang, J., Huang, X., Zhang, Z., 2019. Approximation algorithms for distance constraint sweep coverage with base stations. J. Comb. Optim. 37, 1111–1125.

Miller, R., 2013. Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20 22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department. Springer Science & Business Media.

Ntafos, S., 1992. Watchman routes under limited visibility. Comput. Geom. 1 (3), 149–170.

Sharifi, F., Zhang, Y., Aghdam, A.G., 2014. Forest fire monitoring and detection using a network of autonomous vehicles. In: Proceedings of International Conference on Intelligent Unmanned Systems. Vol. 10.

Sorbelli, F.B., Navarra, A., Palazzetti, L., Pinotti, C.M., Prencipe, G., 2022. Optimal and heuristic algorithms for data collection by using an energy- and storage-constrained drone. In: Erlebach, T., Segal, M. (Eds.), Algorithmics of Wireless Networks - 18th International Symposium on Algorithmics of Wireless Networks, ALGOSENSORS 2022, Potsdam, Germany, September 8-9, 2022, Proceedings. In: Lecture Notes in Computer Science, Vol. 13707, Springer, pp. 18–30. http://dx.doi.org/10.1007/978-3-031-22050-0_2.

Sorbelli, F.B., Navarra, A., Palazzetti, L., Pinotti, C.M., Prencipe, G., 2024. Wireless IoT sensors data collection reward maximization by leveraging multiple energy-and storage-constrained UAVs. J. Comput. System Sci. 139, 103475.

Thiels, C.A., Aho, J.M., Zietlow, S.P., Jenkins, D.H., 2015. Use of unmanned aerial vehicles for medical product transport. Air Med. J. 34 (2), 104–108.

Valavanis, K.P., Vachtsevanos, G.J., 2014. Handbook of Unmanned Aerial Vehicles. Springer Publishing Company, Incorporated.

Vansteenwegen, P., Souffriau, W., Berghe, G.V., Oudheusden, D.V., 2009. Metaheuristics for tourist trip planning. In: Sörensen, K., Sevaux, M., Habenicht, W., Geiger, M.J. (Eds.), Metaheuristics in the Service Industry. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 15–31.