A MILP Formulation and a Metaheuristic Approach for the Scheduling of Drone Landings and Payload Changes on an Automatic Platform

Elena Ausonio, Patrizia Bagnerini, and Mauro Gaggero

Abstract We present a mixed-integer linear programming formulation and a metaheuristic approach based on direct search to schedule landings and payload changes of a set of unmanned aerial vehicles that cooperate to achieve given mission objectives. In more detail, such vehicles require landing on an automatic platform able to rapidly substitute batteries and switch the payload they are currently carrying with another one, if required by the mission at hand. Preliminary numerical results are presented to show the effectiveness of the metaheuristic algorithm as a compromise between accuracy of suboptimal solutions and computational effort.

Key words: Scheduling, unmanned aerial vehicles (UAVs), landing, payload change, automatic platform, drones

1 Introduction

In the last decades, unmanned aerial vehicles (UAVs), also called drones, have become extremely popular owing to their ability to perform various kinds of missions consisting of different tasks. Examples of application are environmental monitoring, geographic mapping, search and rescue, shipping and delivery, precision agriculture, inspection and surveillance, and fire suppression [5, 10, 13, 14]. In fact, UAVs are able to carry a variety of equipment, generally called payload, such as thermal and multispectral cameras, LiDAR, and other sensors. Despite the technological evolution of drones, there are still two major limitations preventing their systematic use: the limited operating autonomy of batteries that affect the area a UAV can cover [3], and the difficulty of performing complex missions requiring the coordination of

Elena Ausonio, Patrizia Bagnerini
University of Genoa, Genoa, Italy, e-mail: elena.ausonio@edu.unige.it, patrizia.bagnerini@unige.it

Mauro Gaggero (✉)
National Research Council of Italy, Genoa, Italy, e-mail: mauro.gaggero@cnr.it

multiple devices [20]. To overcome these limitations, besides the enhancement of cooperation algorithms (not considered in this work), a possibility is to automate drone management through a landing platform able to automatically change payloads or batteries and recharge exhausted ones [11, 23]. Several companies around the world are working towards the development of such automatic platforms (see, e.g., [1, 2]). The paradigm shift is evident: from being a tool requiring continuous human intervention, UAVs can be employed to perform activities even 24/7.

In this paper, we focus on a set of drones flying in a given area and performing different tasks by means of suitable payloads. After concluding the assigned task or due to the need of replacing exhausted batteries, UAVs have to land on the aforementioned automatic platforms, which perform a change of battery with a newly-charged one and switch the payload carried before landing. In more detail, the payloads that they were carrying before landing are let on the platforms, and new ones are assigned. After such operations, drones take off and continue their mission. The payloads left on the platforms can then be mounted on the next drones landing therein. If UAVs land on platforms where the requested payload is unavailable, we assume that an alternative payload can be assigned (for instance, a less performing sensor with respect to the required one, or another one with similar characteristics) in order to avoid aborting the mission. The full functionality and efficiency of this scenario requires to assign in an optimal way the shared resources (platforms and payloads) to drones. Hence, it is crucial to sequence landings and take-offs, as well as to assign a payload to each drone after landing according to requests and availability on the chosen landing platform. From now on we refer to such a problem as *drone scheduling problem* (DSP).

First, we formulate the DSP as a mixed-integer linear programming (MILP) problem. However, finding a solution may be complex in the case of a large number of drones. The DSP was first introduced in [4], together with a simple heuristic based on a greedy choice of landing times and payload assignments, which was not satisfactory for problems with a large number of drones. Thus, here we propose a metaheuristic approach able to find suboptimal solutions that are close to those computed by the optimization-based method also for large-dimensional problem instances, but with large savings on the computational time. Such an approach is based on direct-search optimization, and in particular on the generalized pattern search (GPS) algorithm. Numerical results obtained in a test case are reported and discussed to evaluate the effectiveness of the proposed approaches.

Although a large scientific literature exists for the management of military aircraft missions or scheduling landings at an airport [6, 9, 15], the application to UAVs has not yet received the same attention. In most cases, the considered setting belongs to the family of the so-called aircraft landing problem, which consists in minimizing the difference between actual and target aircraft landing times under suitable constraints. Reference [18] presents a simple, recent approach to the solution of this problem, based on the decomposition into a chain of smaller, easier-to-solve cases. Several heuristic or metaheuristic algorithms have been proposed as well to reduce the computational burden in finding suboptimal solutions (see, e.g., [12, 19, 22]). The main difference between the DSP and the aircraft landing problem is the need of considering, in the former, the availability of the desired payload on the landing platform together with the deviation from the target landing time. The general problem of scheduling unmanned vehicles, instead, has received a lot of attention from the research community. Several works exist that consider autonomous vehicles to deliver packages through logistics networks. Many companies (e.g., Amazon, DHL, and UPS) are currently exploring the practical use of drones for parcel delivery. In this context, one of the first works that employs drones for last-mile delivery in logistics operations is [17]. It is based on a variant of the traveling salesman problem, where UAVs operate along with traditional delivery trucks to distribute packages. MILP formulations for two delivery-by-drone problems are presented, together with two heuristic approaches for solving large-scale problems. Reference [8] investigates the scheduling of UAVs that take off from a delivery truck traveling on a route with a predetermined sequence of stops. The goal is to minimize the total duration of the delivery round, making sure that customers receive their deliveries. Finally, [21] investigates the delivery problem in relation to the battery charge level, and focuses on the impact of battery consumption on fleet scheduling.

The rest of this paper is organized as follows. Section 2 reports the MILP formulation of the problem. Section 3 presents the metaheuristic approach to find suboptimal solutions. Section 4 discusses numerical results. Conclusions are drawn in Section 5.

2 Problem Formulation

Let us focus on a DSP instance composed of N drones to be scheduled within the time interval $[0, \mathcal{T}]$, where $\mathcal{T}$ is a given horizon. We perform a time discretization of $[0, \mathcal{T}]$, i.e., we focus on the discrete instants $t = 0, \Delta t, 2\Delta t, \ldots, T_{\text{fin}}\Delta t$, where T_{fin} is the total number of time steps and $\Delta t := \mathcal{T}/T_{\text{fin}}$ is the sampling time. With a little abuse of notation, from now on we refer to the discrete time steps $t = 0, 1, \ldots, T_{\text{fin}}$. Without loss of generality, we consider missions that require a unique landing, payload switch, and take-off per drone, and we assume that UAVs can carry only one payload at a time. A total number of different payload types equal to K is considered, and the number of payloads for each type is equal to P_k, $k = 1, \ldots, K$. The type of payload carried by drones before landing is accounted for by the 0-1 parameter $B_{i,k}$, $i = 1 \ldots, N$, $k = 1, \ldots, K$, which is equal to 1 if drone i carries the payload type k before landing, while it is equal to 0 otherwise. After landing, each drone has a "desired" payload type D_i, $i = 1, \ldots, N$, that has to be assigned to it in order to continue the mission after take-off. Each drone is characterized by a target landing time T_i, $i = 1, \ldots, N$, and by a time interval $[E_i, L_i]$ for landing, where E_i is the earliest landing time and L_i is the latest landing time. To allow a correct setup of the automatic platform for payload switches, we assume that two consecutive landings must be separated of at least S time steps. Summarizing, we define an instance of the DSP as the set $\mathcal{I}$ of all the previous input parameters, i.e., $\mathcal{I} := \{N, T_{\text{fin}}, K, P_k, B_{i,k}, D_i, T_i, E_i, L_i, S, i = 1, \ldots, N, k = 1, \ldots, K\}$.

Two goals can be identified for the DSP: (i) schedule landings so that drones land as close as possible to target times, and (ii) after landing, assign a payload type to drones as close as possible to the desired one. Goals (i) and (ii) are pursued by formulating the DSP as an optimization problem. To this end, let us consider the following decision variables:

- $x_{i,t}$, $i = 1, \dots, N$, $t = 0, 1, \dots, T_{\text{fin}}$, is a binary variable equal to 1 if drone i lands at time t, otherwise it is equal to 0;
- $p_{i,k,t}$, $i = 1, \dots, N$, $k = 1, \dots, K$, $t = 0, 1, \dots, T_{\text{fin}}$, is a binary variable equal to 1 if drone i carries a payload of type k at time t, otherwise it is equal to 0;
- $c_{i,k,t}$, $i = 1, \dots, N$, $k = 1, \dots, K$, $t = 0, 1, \dots, T_{\text{fin}}$, is an integer variable equal to the difference between the desired payload type and the actually-assigned one after drone landing; it is useful to linearize the cost and avoid absolute values;
- a_i, $i = 1, \dots, N$, is an integer variable representing the earliness in the landing of drone i with respect to the target time T_i;
- b_i, $i = 1, \dots, N$, is an integer variable accounting for the lateness in the landing of drone i with respect to the target time T_i;
- $y_{i,k,t}$, $i = 1, \dots, N$, $k = 1, \dots, K$, $t = 0, 1, \dots, T_{\text{fin}}$, is a binary variable that allow to write constraints on the payload types involving an absolute value in a linear way (see later for details).

We formulate the DSP as the following MILP problem.

$$\min \sum_{i=1}^{N} (a_i + b_i) + \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{t=0}^{T_{\text{fin}}-1} c_{i,k,t} \tag{1}$$

subject to

$$\sum_{t=0}^{T_{\text{fin}}} x_{i,t} = 1, \quad i = 1, \dots, N, \tag{2}$$

$$\sum_{i=1}^{N} x_{i,t} \leqslant 1, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{3}$$

$$E_i \leqslant \sum_{t=0}^{T_{\text{fin}}} t x_{i,t} \leqslant L_i, \quad i = 1, \dots, N, \tag{4}$$

$$\sum_{i=1}^{N} p_{i,k,t} \leqslant P_k, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{5}$$

$$\sum_{k=1}^{K} p_{i,k,t} = 1, \quad i = 1, \dots, N, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{6}$$

$$\sum_{l=t}^{t+S-1} \sum_{i=1}^{N} x_{i,l} \leqslant 1, \quad t = 0, 1, \dots, T_{\text{fin}} - S, \tag{7}$$

$$\sum_{t=0}^{T_{\text{fin}}} x_{i,t} (t - T_i) = b_i - a_i, \quad i = 1, \dots, N, \tag{8}$$

$$p_{i,k,0} = B_{i,k}, \quad i = 1, \dots, N, \quad k = 1, \dots, K, \tag{9}$$

$$p_{i,k,t+1} \geqslant p_{i,k,t} - Mx_{i,t}, \quad i=1,\dots,N, \quad k=1,\dots,K, \quad t=0,1,\dots,T_{\text{fin}}-1, \tag{10}$$

$$p_{i,k,t+1} \leqslant p_{i,k,t} + Mx_{i,t}, \quad i=1,\dots,N, \quad k=1,\dots,K, \quad t=0,1,\dots,T_{\text{fin}}-1, \tag{11}$$

$$c_{i,k,t} \leqslant Mx_{i,t}, \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{12}$$

$$p_{i,k,t+1}(k - D_i) + My_{i,k,t} \geqslant c_{i,k,t} - M(1 - x_{i,t}), \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}} - 1, \tag{13}$$

$$- p_{i,k,t+1}(k - D_i) + My_{i,k,t} \geqslant c_{i,k,t} - M(1 - x_{i,t}), \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}} - 1, \tag{14}$$

$$p_{i,k,t+1}(k - D_i) \leqslant c_{i,k,t} + M(1 - x_{i,t}), \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}} - 1, \tag{15}$$

$$- p_{i,k,t+1}(k - D_i) \leqslant c_{i,k,t} + M(1 - x_{i,t}), \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}} - 1, \tag{16}$$

$$x_{i,t} \in \{0, 1\}, \quad i = 1, \dots, N, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{17}$$

$$p_{i,k,t} \in \{0, 1\}, \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{18}$$

$$y_{i,k,t} \in \{0, 1\}, \quad i = 1, \dots, N, \quad k = 1, \dots, K, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{19}$$

$$c_{i,k,t} \geqslant 0, \quad i = 1, \dots, N, \quad t = 0, 1, \dots, T_{\text{fin}}, \tag{20}$$

$$a_i \geqslant 0, \quad i = 1, \dots, N, \tag{21}$$

$$b_i \geqslant 0, \quad i = 1, \dots, N, \tag{22}$$

where M is a very large positive constant. The cost function in (1) is made up of two terms that reflect the previously-introduced objectives (i) and (ii). The first term penalizes the landing of drones before or after the corresponding target times. Along with constraint (8), it is a linearized version of the absolute value of the difference between the actual landing time of drones and the corresponding target times, i.e., $\sum_{i=1}^{N} \sum_{t=0}^{T_{\text{fin}}} |t\, x_{i,t} - T_i|$. The second term penalizes the assignment to UAVs of payload types different that the desired ones after landing (see also constraints (12)–(16)).

Constraint (2) guarantees that each UAV lands only once, whereas (3) ensures that, for each time t, at most one drone can land. Equation (4) imposes that the landing time of UAVs lies in the interval between the corresponding earliest and latest times. Constraint (5) enforces that, at each time step, the number of assigned payloads of each type does not exceed the overall number of payloads of the same type, while (6) guarantees that each drone carries one and only one payload. Formula (7) separates two consecutive landings of at least S time instants, whereas (8) connects the decision variables $x_{i,t}$ to a_i and b_i, for all $i = 1, \dots, N$ and $t = 0, 1, \dots, T_{\text{fin}}$. Equation (9) provides initialization of the decision variable $p_{i,k,0}$ at $t = 0$ with the information on the carried payload type at the beginning of the mission, contained in the input parameter $B_{i,k}$. Constraints (10) and (11) are equivalent to the following, and establish a dynamics for the type of payloads carried by the various UAVs:

$$p_{i,k,t+1} = \begin{cases} p_{i,k,t} & \text{if } x_{i,t} = 0, \\ \text{unconstrained} & \text{if } x_{i,t} = 1, \end{cases} \quad i=1,\dots,N, k=1,\dots,K, t=0,1,\dots,T_{\text{fin}}-1.$$

More specifically, the payload of a drone i not landing at time t (i.e., such that $x_{i,t} = 0$) does not change from time t to $t+1$. Otherwise, the constraints are trivially satisfied due to the presence of the large positive constant M, i.e., $p_{i,k,t+1}$ is unconstrained. Similarly, constraints (12)–(16) are equivalent to the following:

$$c_{i,k,t} = \begin{cases} 0 & \text{if } x_{i,t} = 0, \\ |p_{i,k,t+1}(k - D_i)| & \text{if } x_{i,t} = 1, \end{cases} \quad i = 1, \ldots, N, k = 1, \ldots, K, t = 0, 1, \ldots, T_{\text{fin}} - 1.$$

In particular, a penalty is paid in the second term of the cost function (1) if a payload type different than the desired one is assigned to a drone landing at time t. The absolute value is then removed by using standard arguments, from which (12)–(16) are derived. Lastly, (17)–(22) define the decision variables.

3 A Metaheuristic Algorithm Based on Direct Search

The MILP formulation described in Section 2 is too complex to be solved in real time for problem instances characterized by a large number of drones. Thus, in this section, we propose a metaheuristic approach based on direct-search optimization to find suboptimal solutions in a reduced amount of time.

The goal of the metaheuristic approach is to find the optimal values of drone landing times $\tau_i := \sum_{t=0}^{T_{\text{fin}}} t\, x_{i,t}$ and payload type assignments after landing $\pi_i := \sum_{k=1}^{K} \sum_{t=0}^{T_{\text{fin}}-1} k\, p_{i,k,t+1}\, x_{i,t}$, for all $i = 1, \ldots, N$. To this end, let $\rho_i := \sum_{k=1}^{K} k B_{i,k}$, $i = 1, \ldots, N$, be the payload type carried by drone i before landing, and let $G_k := P_k - \sum_{i=1}^{N} B_{i,k}$, $k = 1, \ldots, K$, be the number of payloads of type k that are available on the landing platform at $t = 0$ and that are not carried by drones. The latter quantity has to be updated after each landing to track the number of payloads per each type that are actually carried by drones, as detailed in the following.

Optimization is done by using the GPS algorithm, which performs a local search on a given mesh around the current solution to reduce the cost at each iteration [7]. In more detail, we focus on the minimization of the following cost, which is a slightly revised version of (1) written in terms of the variables τ_i and π_i and with the introduction of the function $f : \mathbb{N} \to \mathbb{N}$ in the second term to account for constraints on the availability of payloads, as detailed later on:

$$J(X) := \sum_{i=1}^{N} |\tau_i - T_i| + \sum_{i=1}^{N} |f(\pi_i) - D_i|, \qquad (23)$$

where $X := (\tau_i, \pi_i, i = 1, \ldots, N) \in \mathbb{R}^{2N}$, and

$$f(\pi_i) := \begin{cases} \pi_i & \text{if } G_{\pi_i} > 0, \\ \min_{\pi \in \{1,\ldots,K\}} (|\pi - \pi_i| : G_\pi > 0) & \text{otherwise.} \end{cases}$$

As in (1), the first term of (23) penalizes deviations from target landing times, while the second one penalizes the assignment of payloads different than the desired ones.

Let X^k be the solution at iteration k of the GPS algorithm, obtained with a mesh size $\Delta X^k > 0$. Then, the function $J(X)$ is evaluated in the points $X'^k := X^k \pm \Delta X^k e_j$, $j = 1, \ldots, 2N$, around the current solution by discarding unfeasible ones (constraints are described in the next paragraph). The set of points X'^k is called pattern, the generation of the pattern is called polling, and the set of vectors e_j is a spanning set of $\mathbb{R}^{2N}$ (the simplest choice is given by the versors of the coordinate axes). After the construction of the pattern, we look for the point $X^{\circ k} := \arg\min J(X'^k)$, such that $J(X^{\circ k}) < J(X^k)$. The polling is successful if $X^{\circ k}$ exists. In this case, we generate a new solution point by setting $X^{k+1} := X^{\circ k}$ and $\Delta X^{k+1} := 2\Delta X^k$, i.e., the mesh size is increased to look for possible better points far from the current solution. Otherwise, the polling is unsuccessful, and we let $X^{k+1} := X^k$ along with $\Delta X^{k+1} := \Delta X^k/2$, i.e., the mesh size is reduced to allow a finer search around the current best solution. The procedure is iterated until ΔX^k is smaller than a certain tolerance value, and a maximum number of iterations or cost function evaluations is reached [16].

Constraints (5), (6), and (9)–(16) on the availability and dynamics of the various payload types are taken into account in the following way by means of the function f and the update of G_k at each landing. To fix ideas, let us focus on a generic drone i. First, the quantity G_{ρ_i} is increased of one unit to account for the payload type ρ_i that drone i was carrying before landing. Then, it is checked whether the payload type π_i selected by the GPS algorithm is available on the platform, i.e., if $G_{\pi_i} > 0$. If the payload type π_i is available, then it is actually assigned to drone i, i.e., $f(\pi_i) = \pi_i$, and the quantity G_{π_i} is reduced of one unit to model the pick up of the payload from the platform. Otherwise, an alternative payload type has to be assigned to drone i since the choice of π_i does not satisfy the availability constraints. The payload type actually assigned to drone i is chosen as the one that is closest to π_i among those that are available on the platform, i.e., $f(\pi_i) = \min_{\pi \in \{1,\ldots,K\}} (|\pi - \pi_i| : G_\pi > 0)$. The aforesaid revealed to be more effective than directly managing constraints on the positivity of G_k, $k = 1, \ldots, K$, after each landing. Indeed, the number of constraints in the latter case increases with the number of drones and payload types, thus complicating finding a solution. Constraints (2)–(4) and (7) related to landing times were taken into account by imposing proper bounds for the generation of the pattern points, while (8) was implicitly considered through the definition of the cost (23).

4 Numerical Results

We report in this section preliminary numerical results to verify the effectiveness of the MILP formulation and of the metaheuristic approach to find a solution to the DSP. Tests were performed using a personal computer equipped with an Intel Core i9 processor having a clock frequency equal to 3.6 GHz and 64 GB of RAM.

We considered 15 instances of the DSP, each one corresponding to a fixed number of drones, from 10 to 150. In particular, the l-th instance $\mathcal{I}^{(l)} := \{N^{(l)}, T_{\text{fin}}^{(l)}, K^{(l)}, P_k^{(l)}, B_{i,k}^{(l)}, D_i^{(l)}, T_i^{(l)}, E_i^{(l)}, L_i^{(l)}, S^{(l)}, i = 1, \ldots, N^{(l)}, k = 1, \ldots, K^{(l)}\}$ had a number of drones $N^{(l)} := 10\,l$, where $l = 1, \ldots, 15$. All the instances were characterized by $K^{(l)} := 5$ different payload types, and by a total number of payloads $\sum_{k=1}^{K^{(l)}} P_k^{(l)}$ equal to $N^{(l)} + 3$. The 3 exceeding payloads with respect to the number of drones were randomly assigned to the various payload types via random extractions from discrete uniform distributions in the range $[1, K^{(l)}]$. The parameter $S^{(l)}$ was fixed to 2 for all $l = 1, \ldots, 15$. For each instance l, the payload types $B_{i,k}^{(l)}$ carried by drones before landing, and the desired ones after landing $D_i^{(l)}$, were randomly extracted from discrete uniform distributions in the ranges $[0, 1]$ and $[1, K^{(l)}]$, respectively. The earliest landing times $E_i^{(l)}$ were again randomly extracted from discrete uniform distributions in the range $[E_{i-1}^{(l)}, 5(N^{(l)} - 1)]$, with $E_0^{(l)} := 0$ for all l. Instead, the latest landing times $L_i^{(l)}$ were randomly drawn from discrete uniform distributions in the range $[E_i^{(l)} + 5, 5(N^{(l)} - 1)]$, with $L_i^{(l)} := 5(N^{(l)} - 1)$ if $E_i^{(l)} + 5 > 5(N^{(l)} - 1)$. Lastly, the target landing times $T_i^{(l)}$ were randomly drawn from discrete uniform distributions in the range $[E_i^{(l)}, L_i^{(l)}]$. The total number of time steps $T_{\text{fin}}^{(l)}$ was set to $\max_i\{L_i^{(l)}\}$ for each instance $l = 1, \ldots, 15$, with a sampling time Δt equal to 1 minute. In this way, the largest mission duration for the instances characterized by 150 drones was equal to about 12 hours.

We solved the DSP both using the CPLEX solver applied to the MILP formulation (1)–(22), and the metaheuristic approach described in Section 3. Concerning CPLEX, we set a maximum time limit equal to 1 hour and a maximum relative optimality gap equal to 10^{-4}. As regards the metaheuristic approach, we relied on the Matlab function *patternsearch*, which provides an implementation of the GPS algorithm described in Section 3, with tolerance values for the stopping criteria equal to 10^{-6} (including the mesh size), and a maximum number of iterations equal to $200\,N^{(l)}$ for $l = 1, \ldots, 15$.

Since all scenarios involve the generation of random numbers, we repeated 10 times the random extractions to have statistical significance of results. Thus, each instance $\mathcal{I}^{(l)}$ was run a total of 10 times with different values for the random quantities. Performances were evaluated by computing the averages of the following indicators over the aforementioned 10 runs:

- the *total cost* (TC) is given by the cost function in (1); it measures the overall trade-off between the objectives (i) and (ii);
- the *total time cost* (TTC) is given by the first term of the cost function in (1); it accounts for the deviation of drone landing times from target times;
- the *total payload cost* (TPC) is given by the second term of the cost function in (1); it quantifies the difference between the payload type desired by drones after landing and the actually-assigned one.
- the *CPU time* (CPU) measures the time required to find a solution.

Table 1 reports the average results, where “OPT” denotes the solution of the MILP formulation and “META” the one of the metaheuristic method. The column “gap” indicates the percentage relative difference between the results provided by the metaheuristic and by the optimization-based approach. We observe that the larger the number of drones, the higher the average values of the performance indicators. Thus,

Table 1 Averages of the performance indicators over 10 simulation runs.

	TC			TTC			TPC			CPU (s)		
N	OPT	META	gap (%)	OPT	META	gap (%)	OPT	META	gap (%)	OPT	META	gap (%)
10	6.40	7.20	0.11	2.40	2.60	0.08	4.00	4.60	0.13	0.09	0.10	0.14
20	17.83	18.17	0.02	5.00	5.00	0.00	12.83	13.17	0.03	0.47	0.28	-0.68
30	24.33	25.00	0.03	6.17	6.67	0.07	18.17	18.33	0.01	1.21	0.52	-1.30
40	30.80	32.60	0.06	8.80	9.40	0.06	22.00	23.20	0.05	3.39	0.98	-2.47
50	39.00	41.40	0.06	9.80	11.60	0.16	29.20	29.80	0.02	5.52	1.60	-2.45
60	47.00	48.20	0.02	11.60	10.00	-0.16	35.40	38.20	0.07	7.13	1.85	-2.86
70	54.60	55.40	0.01	13.20	11.40	-0.16	41.40	44.00	0.06	20.40	2.39	-7.52
80	57.67	62.00	0.07	15.00	17.00	0.12	42.67	45.00	0.05	34.11	3.67	-8.31
90	63.00	67.67	0.07	16.00	18.67	0.14	47.00	49.00	0.04	65.42	4.78	-12.68
100	70.67	75.33	0.06	17.33	19.67	0.12	53.33	55.67	0.04	105.39	5.72	-17.41
110	79.00	83.67	0.06	19.33	22.00	0.12	59.67	61.67	0.03	459.32	7.36	-61.45
120	86.00	91.33	0.06	20.67	24.00	0.14	65.33	67.33	0.03	197.83	8.58	-22.05
130	91.00	97.00	0.06	22.67	25.67	0.12	68.33	71.33	0.04	1131.07	10.10	-110.95
140	98.67	106.00	0.07	24.33	27.67	0.12	74.33	78.33	0.05	1671.26	11.88	-139.72
150	105.00	113.00	0.07	27.00	30.33	0.11	78.00	82.67	0.06	2454.94	13.45	-181.56
Avg	58.06	61.60	0.06	14.62	16.11	0.09	43.44	45.49	0.04	410.50	4.88	-83.05

the difficulty in finding a solution to the DSP increases with N, as expected, and also the CPU time required to find a solution grows with N. An almost linear increase of the average TC, TTC, and TPC can be observed for both the optimization-based approach and the metaheuristic one. Instead, the computational requirements of the optimization-based method grow exponentially with the number of drones, whereas the growth is much more limited and linear for the metaheuristic. The accuracy of the metaheuristic method is satisfactory if compared to the optimization-based approach. In fact, the relative gap in terms of the TC is less than 7%, and almost independent of the number of drones characterizing the problem instance. Similar values for the relative gap are experienced for the TPC, while an increase up to a maximum of 14% can be measured for the TTC. In this case, for 60 and 70 UAVs, the metaheuristic provides a solution with drone landing times closer to targets as compared to the optimization-based method. However, such an advantage in the TTC is compensated by greater values of the TPC. Concerning computational time, the metaheuristic approach requires on the average about 13 seconds to find a solution in the largest instance of the DSP corresponding to 150 drones, as compared to $2.5{\cdot}10^3$ seconds of the optimization-based method. Thus, the proposed metaheuristic method represents a good compromise between accuracy of suboptimal solutions and required computational effort. On the average, a gap of 6% on the total cost and a saving of two orders of magnitude in the computational time are observed.

5 Conclusions

We have presented a MILP formulation of the problem of scheduling landings on automatic platforms and payload switches of a set of UAVs performing given missions. Since the optimization problem may be difficult to be solved for large numbers of drones, we have devised a metaheuristic approach based on a direct search method, which has revealed to be effective for either low- and high-dimensional problem instances. Future efforts will be devoted to develop *ad-hoc* heuristic algorithms and multi-objective optimization approaches. Moreover, we plan to extend the approach to the case of multiple landing platforms and the possibility for drones to carry more than one payload and perform multiple landings.

References

1. Airobotics company. https://www.airoboticsdrones.com. Accessed: 2022-04-05.
2. Inspire company. https://www.inspire.flights. Accessed: 2022-04-05.
3. J. Apeland, D. Pavlou, and T. Hemmingsen. Suitability analysis of implementing a fuel cell on a multirotor drone. *J. Aerosp. Technol. Manag.*, 12:1–14, 2020.
4. E. Ausonio, P. Bagnerini, and M. Gaggero. Scheduling landing and payload switch of unmanned aerial vehicles on a single automatic platform. *IEEE Conf. Autom. Sci. Eng.*, 2022, submitted.
5. E. Ausonio, P. Bagnerini, and M. Ghio. Drone swarms in fire suppression activities: A conceptual framework. *Drones*, 5:1–22, 2021. art. no. 17.
6. J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, and D. Abramson. Scheduling aircraft landings - The static case. *Transp. Sci.*, 34:180–197, 2000.
7. C. Bogani, M.G. Gasparo, and A. Papini. Generalized pattern search methods for a class of nonsmooth optimization problems with structure. *J. Comput. Appl. Math.*, 229:283–293, 2009.
8. N. Boysen, D. Briskorn, S. Fedtke, and S. Schwerdfeger. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, 72:506–527, 2018.
9. A. Faye. Solving the aircraft landing problem with time discretization approach. *Eur. J. Oper. Res.*, 242:1028–1038, 2015.
10. D. Floreano and R.J. Wood. Science, technology and the future of small autonomous drones. *Nature*, 521:460–466, 2015.
11. M. Ghio. Methods and apparatus for the employment of drones in firefighting activities. *US Patent*, (US11104436B2), 2017.
12. B. Girish. An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem. *Appl. Soft Comput.*, 44:200–221, 2016.
13. H. Gonzalez-Jorge, J. Martinez-Sanchez, M. Bueno, and P. Arias. Unmanned aerial systems for civil applications: A review. *Drones*, 1:1–19, 2017. art. no. 2.
14. M. Hassanalian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress Aerosp. Sci.*, 91:99–131, 2017.
15. S. Ikli, C. Mancel, M. Mongeau, X. Olive, and E. Rachelson. The aircraft runway scheduling problem: A survey. *Comp. Oper. Res.*, 132:1–20, 2021. art. no. 105336.
16. R.M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM J. Optim.*, 10:917–941, 2000.
17. C.C. Murray and A.G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C: Emerging Technologies*, 54:86–109, 2015.
18. A. Salehipour. An algorithm for single- and multiple-runway aircraft landing problem. *Math. Comput. Simul.*, 175:179–191, 2020.
19. A. Salehipour, M. Modarres, and L.M. Naeni. An efficient hybrid meta-heuristic for aircraft landing problem. *Comp. Oper. Res.*, 40:207–213, 2013.
20. A. Sargolzaei, A. Abbaspour, and C. Crane. *Control of Cooperative Unmanned Aerial Vehicles: Review of Applications, Challenges, and Algorithms*, pages 229–255. Springer, Cham, 2020.
21. M. Torabbeigi, G.J. Lim, and S.J. Kim. Drone delivery scheduling optimization considering payload-induced battery consumption rates. *J. Intell. Rob. Syst.: Th. Appl.*, 97:471–487, 2020.
22. S. Vadlamani and S. Hosseini. A novel heuristic approach for solving aircraft landing problem with single runway. *J. Air Transp. Manag.*, 40:144–148, 2014.
23. M. Wang and X. Chen. System and method for managing unmanned aerial vehicles. *US Patent*, (US20170190260A1), 2017.