2022 IEEE 18th International Conference on Automation
Science and Engineering (CASE)
August 20-24, 2022. Mexico City, Mexico

2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-9042-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CASE49997.2022.9926507

Scheduling Landing and Payload Switch of Unmanned Aerial Vehicles on a Single Automatic Platform

Elena Ausonio, Patrizia Bagnerini, Mauro Gaggero

Abstract— **We focus on the problem of optimally managing a set of unmanned aerial vehicles performing given missions that require to land on an automatic platform, unmount the currently-carried payload, and take off with another payload to complete mission objectives. Such a problem often arises when swarms of drones cooperate to complete monitoring applications or other tasks requiring an efficient schedule of landings and payload switches in a resource-constrained environment. First, the problem is formulated as a mixed-integer linear programming one, which, however, may be complex to be solved for a large number of drones. Thus, we also propose a heuristic algorithm able to find suboptimal solutions with a reduced computational effort. Preliminary simulation results are reported and discussed.**

I. Introduction

Unmanned Aerial Vehicles (UAVs), also known as drones, are flying devices remotely controlled by humans or computers that are able to fly autonomously. During the Twentieth Century, UAVs were almost exclusively used for military applications, whereas recently their use in civil applications has increased exponentially. As a consequence, both research institutions and industrial companies have shown increasing interest in investigating this technology. UAVs are of benefit in many applications since they can handle situations that are too dangerous for humans or manned vehicles [1]. Collaborative drones can be applied in different, socially-relevant contexts. Eamples are the placement of sensors in emergency situations (such as landslides, earthquakes, fires), land monitoring, geographic mapping, missing persons search and survivor identification, shipping and delivery, spraying of fertilizers on hard-to-reach crop fields, restoration of broken radio links, as well as inspection of potentially dangerous places and under difficult environmental conditions (see, among others, [2], [3]). The role of UAVs may be also crucial for difficult-to-control forest fires that would quickly put the safety of pilots and rescue teams at risk [4]–[6]. To accomplish all the aforementioned tasks, drones can be equipped with many different types of payloads, such as sensors of various kinds, thermal and multispectral cameras, LiDAR, etc.

As compared to a single device, multiple UAVs can perform complex tasks with more safety and efficiency, and also guarantee redundancy and robustness of the entire system [7]. Several companies, such as Inspire [8] and Airobotics [9], are therefore developing solutions to manage a large number of UAVs in an automated way. The basic idea is to automate drone management through a landing platform able to automatically change payloads or batteries and recharge exhausted ones [10], [11]. If the number of drones is large, the landing and take-off on the platform becomes a very difficult operation, similar to the management of aircraft landings on an aircraft carrier. In this context, the purpose of this paper is to study the feasibility of such types of platforms from the perspective of managing the real-time scheduling of drone landing, payload switch, and take-off in order to prevent the platform itself from being a bottleneck. In more detail, we assume that the drone platform has the following requirements: (i) it has to handle drones and payloads of different types through a flange that adapts to the various types of payloads; (ii) it has to be able to automatically replace exhausted batteries and insert them into a charging circuit, given the limited battery duration of current drones; (iii) it has to be capable of automatically changing the payload of UAVs.

Given the novelty of introducing platforms for managing drone swarms in an automated manner, to the best of the authors' knowledge, there are no algorithms in the literature for scheduling drone landings and payload switches on such platforms. Instead, there exist several papers concerning the management of military aircraft missions or the scheduling of landing at airports [12]–[14]. In most cases, the considered setting belongs to the family of the aircraft landing problem, a classic optimization problem that consists in determining the optimal sequence of aircraft landings such that the deviation between target and actual landing times is minimized. The choice of the landing sequence must also take into account constraints such as avoiding more than one landing on the same airport runway at the same time, as well as considering aircraft maneuver times, which are usually known. Reference [15] presents a simple, recent approach to the solution of this problem, by decomposing it into a chain of smaller, easier-to-solve, cases. Several heuristic algorithms have been proposed as well to reduce the computational burden in finding suboptimal solutions. The interested reader is referred, among others, to [16], [17] for a discussion.

The main difference between aircraft and drone scheduling is the need of considering, in the latter, also the availability of the desired payload on the landing platform together with the deviation from the target time, as well as constraints due to residual battery life. Concerning scheduling of unmanned vehicles, a number of works exist that consider the context of ground vehicles to deliver packages through logistics networks [18], [19]. These studies address the logistics of

E. Ausonio and P. Bagnerini are with the University of Genoa, Genoa, Italy (e-mail: elena.ausonio@edu.unige.it, patrizia.bagnerini@unige.it).

M. Gaggero is with the National Research Council of Italy, Genoa, Italy (e-mail: mauro.gaggero@cnr.it).

drones, i.e., how to distribute them across an area according to demand, but do not focus on battery and payload changes or the choice of the landing sequences on the platforms.

In this paper, we consider the availability of an automatic platform that meets the previously-introduced requirements (i), (ii), and (iii). We focus on a swarm of drones that fly in a given area and perform various kinds of tasks by means of a set of assigned payloads. After concluding their task or due to the need of changing their battery, drones land on some platforms where they obtain battery replacement and, if required by the type of mission, leave the payload they were previously carrying and get a new one. After these operations, drones take off to continue their mission. The payloads left on the platforms can then be mounted upon request on the next drones landing therein. 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 decide priorities for 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. If drones land on a platform 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). For the sake of brevity, from now on we refer to such a problem as *drone scheduling problem*, or DSP for short.

We first formulate the DSP as a mixed-integer linear programming (MILP) problem, by defining proper objectives and constraints. However, finding a solution to this problem may be complex in the case of a large number of drones, which may prevent the real-time application of the proposed approach. Thus, we present also a heuristic algorithm based on a greedy choice of the landing sequence of UAVs and payload assignments, which requires much less computational effort as compared to the solution of the optimization problem. Preliminary simulation results are reported and discussed to evaluate the effectiveness of managing multiple devices on a single automatic platform.

The rest of this paper is organized as follows. Section II reports the formulation of the problem as an optimization one, Section III discusses a solution approach based on a greedy heuristic, and Section IV presents preliminary simulation results. Lastly, Section V concludes this paper.

II. Problem Formulation

We consider an instance of the DSP made up of N drones over a time interval $[0, \mathcal{T}]$, where $\mathcal{T}$ is a fixed horizon. More specifically, we focus on the case of a single landing platform and on missions characterized by a unique landing, payload switch, and take-off per drone. The time interval $[0, \mathcal{T}]$ is discretized into time instants $0, \Delta t, 2\Delta t, \ldots, T_{\text{fin}}\Delta t$, where $\Delta t = \mathcal{T}/T_{\text{fin}}$ and T_{fin} are the sampling time and the total number of time steps, respectively. For the sake of notation compactness, from now on we only refer to the discrete time steps $t = 0, 1, \ldots, T_{\text{fin}}$. We assume that UAVs can carry one payload at a time among K different types, and that the number of available payloads for each type is equal to P_k, $k = 1, \ldots, K$. Before landing, drones are assumed to carry a certain type of payload, while, after landing, they require to mount another payload in order to take off again and continue their mission. Toward this end, we introduce the 0-1 parameter $B_{i,k}$, $i = 1 \ldots, N$, $k = 1, \ldots, K$, denoting whether drone i carries the payload of type k before landing. In particular, $B_{i,k} = 1$ denotes that drone i is carrying a payload of type k, and $B_{i,k} = 0$ otherwise. The type of payload desired after landing by drone i is denoted by D_i, $i = 1, \ldots, N$. Each UAV has a target landing time T_i, $i = 1, \ldots, N$, and a time interval $[E_i, L_i]$ to perform landing, where E_i and L_i are the earliest and latest landing times, respectively. For safety reasons, two consecutive landings must be separated of at least S time instants.

Summarizing, we define an instance of the DSP as a 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 main objectives are considered for the DSP:

(i) organize landings so that UAVs can land as close as possible to their target times;
(ii) after landing, assign to drones a type of payload as close as possible to the desired one.

The aforementioned objectives are taken into account by devising an optimization problem that consists in minimizing a certain cost function under suitable constraints. Toward this end, we introduce the following decision variables.

- $x_{i,t}$, $i = 1, \ldots, N$, $t = 0, 1, \ldots, T_{\text{fin}}$, is a binary variable indicating the landing time of UAVs. In more detail, $x_{i,t} = 1$ if drone i lands at time t, and $x_{i,t} = 0$ otherwise;
- $p_{i,k,t}$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, $t = 0, 1, \ldots, T_{\text{fin}}$, is a binary variable representing the type of payload carried by UAVs. In particular, $p_{i,k,t} = 1$ if drone i carries a payload of type k at time t, and $p_{i,k,t} = 0$ otherwise;
- $c_{i,k,t}$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, $t = 0, 1, \ldots, T_{\text{fin}}$, is an integer variable given by the difference between the desired payload type and the one actually assigned after landing. It is used to linearize the cost function and avoid using absolute values (see later on for details);
- a_i, $i = 1, \ldots, N$, is an integer variable denoting the earliness in the landing of drone i with respect to the target time T_i;
- b_i, $i = 1, \ldots, N$, is an integer variable denoting the lateness in the landing of drone i with respect to the target time T_i;
- $y_{i,k,t}$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, $t = 0, 1, \ldots, T_{\text{fin}}$, is an auxiliary binary variable that is used to manage constraints on the types of payload and involving an absolute value, as it will be discussed later on.

Based on the previous definitions, we formulate the DSP as a MILP problem as follows. First of all, let us define the cost function, which is given by

$$\min\ W_1 \sum_{i=1}^{N}(a_i+b_i) + W_2 \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{t=0}^{T_{\text{fin}}} c_{i,k,t}, \qquad (1)$$

where W_1 and W_2 are positive coefficients weighting the two terms of the cost, which reflect the two previously-introduced objectives (i) and (ii). The first term minimizes the landing of drones before or after the corresponding target times. It is a linearized version of the absolute value of the difference between the actual landing times 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. In fact, as previously pointed out, if a drone lands on a platform where the requested payload is unavailable, we assume that an alternative payload can be assigned.

The minimization in (1) is pursued subject to several constraints. First, we have to guarantee that each UAV lands only once, and that, for each time t, at most one drone can land. This is obtained by the following constraints:

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

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

Moreover, we impose that the landing time of UAVs is between the corresponding earliest and latest times, i.e.,

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

The following two constraints enforce 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, and that each drone carries one and only one payload:

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

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

Due to safety reasons, two consecutive landings must be separated of at least S time instants, i.e., we impose

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

The decision variables $x_{i,t}$ are connected to a_i and b_i, for all $i = 1,\dots,N$ and $t = 0,1,\dots,T_{\text{fin}}$, through the constraints

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

The next equation 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 $B_{i,k}$:

$$p_{i,k,0} = B_{i,k}, \quad i = 1,\dots,N, k = 1,\dots,K. \qquad (9)$$

Moreover, we have to establish the dynamics for the type of payloads carried by UAVs. Since the payload of a drone not landing at time t does not change from t to $t+1$, we have

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

The aforesaid is taken into account via the following linear constraints, where M is a very large positive constant:

$$p_{i,k,t+1} \geqslant p_{i,k,t} - M x_{i,t},$$
$$i = 1,\dots,N, k = 1,\dots,K, t = 0,1,\dots,T_{\text{fin}} - 1, \qquad (10)$$

$$p_{i,k,t+1} \leqslant p_{i,k,t} + M x_{i,t},$$
$$i = 1,\dots,N, k = 1,\dots,K, t = 0,1,\dots,T_{\text{fin}} - 1. \qquad (11)$$

If $x_{i,t} = 1$, the constraints are trivially satisfied due to the presence of M, i.e., $p_{i,k,t+1}$ is unconstrained.

Similarly, we introduce constraints on the variables $c_{i,k,t}$, $i = 1,\dots,N$, $k = 1,\dots,K$, $t = 0,\dots,T_{\text{fin}}$ to pay a penalty 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. In more detail, the cost to pay for the assignment of a payload type different than the desired one is given by the difference between the indexes corresponding to the involved types of payloads. Thus, we impose

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

The absolute value in the previous equation can be removed by using standard arguments, which allow to derive the following linear constraints:

$$c_{i,k,t} \leqslant M x_{i,t},$$
$$i = 1,\dots,N, k = 1,\dots,K, t = 0,1,\dots,T_{\text{fin}}, \qquad (12)$$

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

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

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

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

Lastly, the following equations define the decision variables:

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

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

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

Algorithm 1 Greedy heuristic for the DSP

Inputs: An instance $\mathcal{I}$ of the DSP
Outputs: Landing times τ_j and payload assignments after landing π_l for all drones $j, l = 1, \ldots, N$

// Choice of landing times
1: Sort drones according to non-decreasing target times
2: Let $j = 1, \ldots, N$ be the indexes of sorted drones
3: Let T'_j, E'_j, L'_j be the sorted versions of T_i, E_i, L_i
4: $\tau_1 \leftarrow T'_1$
5: **for** $j = 2$ **to** N **do**
6: $\quad \tau_j \leftarrow \min_{\tau \in \mathbb{N}} \big(|\tau - T'_j| : \tau \in [E'_j, L'_j]; |\tau - \tau_n| \geqslant S, n=1,\ldots,j\big)$
7: **end for**
// Choice of payload types assigned after landing
8: Let G_k be the number of payloads of type k available ↳ on the landing platform for all $k = 1, \ldots, K$
9: Sort drones according to non-decreasing landing times
10: Let $l = 1, \ldots, N$ be the indexes of sorted drones
11: Let D''_l be the sorted version of D_i
12: **for** $l = 1$ **to** N **do**
13: $\quad$ Let ρ_l be the type of payload carried by drone l ↳ before landing
14: $\quad G_{\rho_l} \leftarrow G_{\rho_l} + 1$
15: $\quad \pi_l \leftarrow \min_{\pi \in \{1,\ldots,K\}} \big(|\pi - D''_l| : G_\pi > 0\big)$
16: $\quad G_{\pi_l} \leftarrow G_{\pi_l} - 1$
17: **end for**
18: **return** τ_j and π_l for all $j, l = 1, \ldots, N$

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

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

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

III. Greedy Heuristic Algorithm

The optimization problem (1)–(22) defined in Section II may be too difficult to be solved for a large number of drones, thus undermining the possibility of finding an optimal schedule in real time. This motivates the development of a heuristic approach that is able to find a suboptimal solution to the DSP with a reduced computational effort. Among the various possibilities, in this paper we focus on a greedy heuristic, which consists in taking the locally optimal decision for each drone. More specifically, for each UAV, the choice of the landing time that is nearest to the target one is performed, together with the assignment of the payload type that is closest to the requested one among those available on the platform. The pseudo-code of the proposed heuristic is reported in Algorithm 1.

Algorithm 1 first determines landing times for all drones, and then computes payload assignments. Concerning landing times, UAVs are sorted according to non-decreasing target times (lines 1–3). Then, the algorithm determines the landing times τ_j for each drone $j = 1, \ldots, N$ at lines 4–7, where j is the index referring to sorted drones (see line 2). In more detail, the landing time of the drone with the lowest target time is set equal to the target time itself (line 4). The landing times of the remaining UAVs are chosen one at a time as the integer numbers minimizing the difference with respect to the target times (line 6). Clearly, the choice of the landing times must take into account constraints equivalent to (4) and (7) of the MILP formulation reported in Section II. In other words, the landing time of drone j must be in the interval $[E'_j, L'_j]$, where E'_j and L'_j are the sorted versions of E_i and L_i, respectively, and the distance between two consecutive landings has to be not less than S time steps. The optimization problem at line 6 of Algorithm 1 for the choice of landing times is a scalar one, and it can be solved with any one-dimensional integer optimization method. If the interval $[E'_j, L'_j]$ is not too large, it can be solved also by enumerating all the integer numbers within the interval, discarding those violating constraints, and then computing the cost for all the feasible ones. The best landing time is the instant corresponding to the minimum cost.

After the computation of all the landing times, Algorithm 1 determines the payload types assigned to UAVs after landing (lines 8–17). First, the quantity G_k, $k = 1, \ldots, K$, is defined as the number of payloads of type k that are available on the platform, i.e., those that are not mounted on UAVs. Formally, we have $G_k = P_k - \sum_{l=1}^{N} B_{l,k}$. Then, payloads are assigned once again one at a time by considering drones in non-decreasing order with respect to landing times (resorting is performed at line 9). In particular, the payload type π_l assigned to drone l, where l is the index referring to drones sorted according to non-decreasing landing times (see line 10), is the payload type that is closest to the desired one among those available on the platform (line 15). The number G_k of payloads of the various types available on the platform is updated at each payload switch at lines 14 and 16, according to the payload type $\rho_l = \sum_{k=1}^{K} k B_{l,k}$ that drone l was carrying before landing and the type π_l assigned after landing, respectively. Lastly, the computed landing times and payload assignments after landing are returned at line 18.

IV. Simulation Results

In this section, we report the results of preliminary simulations to evaluate the effectiveness of the proposed solution approaches. All the tests were performed using an Intel Core i9 processor with clock frequency equal to 3.6 GHz and 64 GB of RAM. In particular, we were interested in evaluating the computational requirements and effectiveness of the optimal solution approach and of the heuristic one for increasing number of drones, in order to assess the scalability of the proposed methods. Moreover, we aimed at estimating the capability of the greedy heuristic to find suboptimal solutions. Toward this end, we first solved the DSP (1)–(22) with the MILP solver available in CPLEX, by imposing a maximum time limit equal to 1 hour and default values for the optimality gap. The large positive constant M was fixed to 1000. Then, we applied the greedy heuristic algorithm described in Section III and compared the obtained results.

We focused on a scenario characterized by an increasing number of drones from 2 up to 150, and hence set up 149 instances $\mathcal{I}^{(l)} = \{N^{(l)}, T_{\text{fin}}^{(l)}, K^{(l)}, P_k^{(l)}, B_{i,k}^{(l)},$

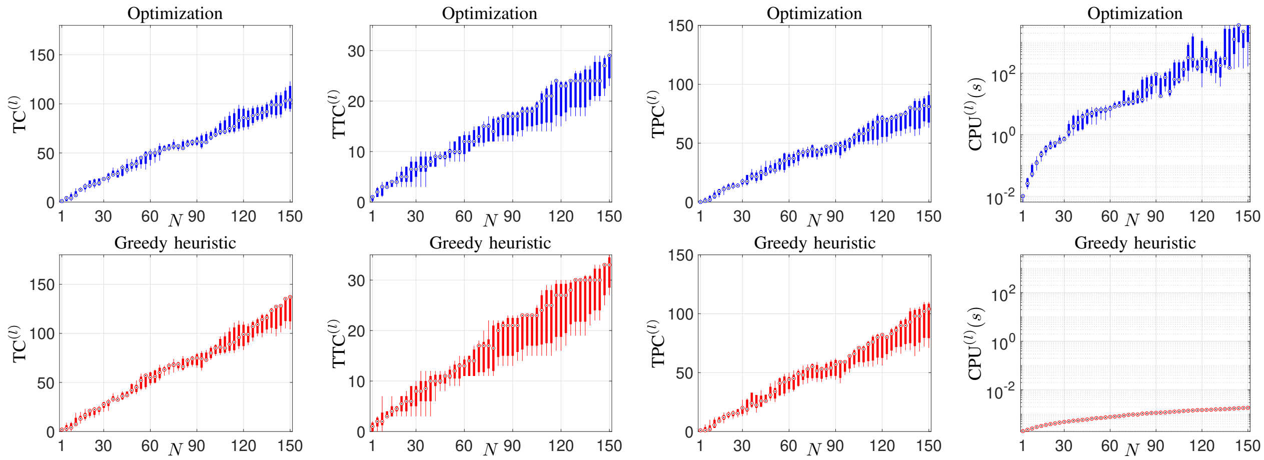

Fig. 1. Boxplots of the performance metrics obtained in 10 realizations of the instances $\mathcal{I}^{(l)}$, $l = 1, \dots, 149$, for the optimization-based approach (in blue, first row) and the greedy heuristic (in red, second row).

$D_i^{(l)}, T_i^{(l)}, E_i^{(l)}, L_i^{(l)}, S^{(l)}, i = 1, \dots, N^{(l)}, k = 1, \dots, K^{(l)}\}$, for $l = 1, \dots, 149$, with $N^{(l)} = l + 1$. Each instance $\mathcal{I}^{(l)}$ was built in such a way that $B_{i,k}^{(l)}$, $D_i^{(l)}$, $T_i^{(l)}$, $E_i^{(l)}$, $L_i^{(l)}$ for $i = 1, \dots, N^{(l)}$ were equal to $B_{i,k}^{(l+1)}$, $D_i^{(l+1)}$, $T_i^{(l+1)}$, $E_i^{(l+1)}$, $L_i^{(l+1)}$ for $i = 1, \dots, N^{(l+1)} - 1$. In other words, $\mathcal{I}^{(l)}$ and $\mathcal{I}^{(l+1)}$ differ for the new drone added to the latter. For all instances $l = 1, \dots, 149$, the number $K^{(l)}$ of payload types was fixed to 5, and the minimum separation $S^{(l)}$ between two consecutive landings was taken equal to 2. The payload types carried by drones before landing $B_{i,k}^{(l)}$, $i = 1, \dots, N^{(l)}$, $k = 1, \dots, K^{(l)}$, and the desired ones after landing $D_i^{(l)}$, $i = 1, \dots, N^{(l)}$, were drawn randomly from discrete uniform distributions in the ranges $[0, 1]$ and $[1, K^{(l)}]$, respectively. Without loss of generality, the overall number of available payloads was set equal to $N^{(l)} + 3$. Each of the 3 exceeding payloads with respect to the number $N^{(l)}$ of UAVs was assigned to the various payload types via a random extraction from a discrete uniform distribution in the range $[1, K^{(l)}]$ to obtain the value of the input parameter $P_k^{(l)}$, $k = 1, \dots, K$. Concerning landing times, the earliest times $E_i^{(l)}$, $i = 1, \dots, N^{(l)}$, for the various instances $l = 1, \dots, 149$ were drawn randomly from discrete uniform distributions in the range $[E_{i-1}^{(l)}, 5(N^{(l)} - 1)]$, with the fictitious $E_0^{(l)}$ set to 0 for all l. The latest times $L_i^{(l)}$, $i = 1, \dots, N^{(l)}$, $l = 1, \dots, 149$, were drawn randomly from discrete uniform distributions in the range $[E_i^{(l)} + 5, 5(N^{(l)} - 1)]$, with $L_i^{(l)} = 5(N^{(l)} - 1)$ if $E_i^{(l)} + 5 > 5(N^{(l)} - 1)$. Consequently, the target times $T_i^{(l)}$, $i = 1, \dots, N^{(l)}$, for each instance $l = 1, \dots, 149$, were randomly extracted from discrete uniform distributions in the range $[E_i^{(l)}, L_i^{(l)}]$. Lastly, the horizon $T_{\text{fin}}^{(l)}$ was set to $\max_i\{L_i^{(l)}\}$ for all $l = 1, \dots, 149$, i.e., it increased together with the number of drones. The sampling time Δt was chosen equal to 1 minute, which corresponds to considering a mission length of about 12 hours in the case of 150 UAVs. In all the instances of the DSP, the weights W_1 and W_2 were both fixed to 1.

The following metrics were adopted for performance measure of each instance $l = 1, \dots, 149$:

- the *total cost* $\text{TC}^{(l)}$ is given by the cost function in (1), and it quantifies the overall effectiveness in terms of trade-off between the two conflicting objectives (i) and (ii) discussed in Section II;
- the *total time cost* $\text{TTC}^{(l)}$ is given by the first term of the cost function in (1), and it measures the deviation of landing times from the corresponding target times;
- the *total payload cost* $\text{TPC}^{(l)}$ is given by the second term of the cost function in (1), and it accounts for the difference between the payload type desired by drones after landing and the actually assigned one.
- the *CPU time* $\text{CPU}^{(l)}$ measures the time required to find a solution.

To pursue statistical significance of results, we generated 10 random realizations for each instance $l = 1, \dots, 149$. Then, we computed the average values of the performance metrics over all the 10 realizations to have a compact view of results.

Figure 1 shows the boxplots of total cost, total time cost, total payload cost, and CPU time obtained over the 10 realizations of the instances $\mathcal{I}^{(l)}$, $l = 1, \dots, 149$, for increasing number of drones from 2 up to 150. Figure 2 reports the averages over l of the aforementioned performance metrics. In general, the larger the number of UAVs, the higher the values of all the performance indicators. In fact, the problem to be solved becomes more and more difficult if the number of drones increases, and therefore it is much more complex to satisfy target landing times and assign desired payload types after landing. As a consequence, the computational requirements grow with the number of drones. Both in the optimization-based approach and in the greedy heuristic, we observe an almost linear increase of the performance metrics TC, TTC, and TPC. The dispersion around the median is comparable for the two approaches, thus suggesting that it is due to the intrinsic variability of the problem instances rather than to the adopted solution method. Concerning computational requirements, we note a large dispersion around the median for the optimization-

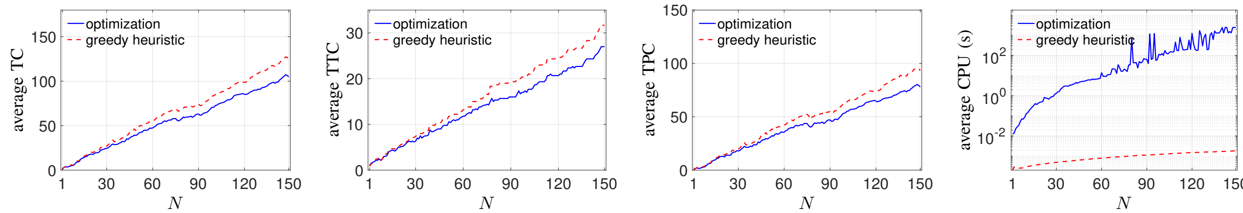

Fig. 2. Averages of the performance metrics obtained in 10 realizations of the instances $\mathcal{I}^{(l)}$, $l = 1, \ldots, 149$, for the optimization-based approach (continuous, blue line) and the greedy heuristic (dashed, red line).

based method, which is sometimes unable to find an optimal solution within the time limit of 1 hour for large N. On the contrary, the dispersion of the greedy heuristic is small and remains almost constant for all the numbers of drones.

As regards accuracy of solution, on the average the relative gap between the optimization-based method and the greedy heuristic is equal to about 13% for either the TC, TTC, and TPC. The two approaches almost coincide for small values of N, while we can observe an increase of the gap for a large number of drones, up to 16%. Such a difference is mainly due to the myopic nature of the greedy heuristic, which may have difficulties in choosing to delay the landing of a drone and anticipate the landing of another one that will leave on the platform the type of payload desired by the first one, especially for large values of N.

Concerning computational effort, the greedy heuristic requires on the average about six orders of magnitude less than the optimization-based method to find a solution. Sometimes, the optimal solution cannot be found by the optimization-based method within the limit of 1 hour for the MILP solver.

Summarizing, we conclude that there is room for improvement in the suboptimal solution of the DSP, even if the proposed greedy heuristic represents a valid starting point. This will be the subject of future research efforts to develop a novel heuristic method able to achieve a better trade-off between accuracy and required computational burden as compared to the optimization-based method.

V. Conclusions

This paper represents a first step in the investigation of scheduling landings and payload switches of multiple drones on a single automatic platform. For this purpose, we have first presented a MILP formulation of the problem, which, however, may be too difficult to be solved in real time for a large number of UAVs. Hence, we have also devised a heuristic approach based on a greedy choice of drone landing times and payload assignments. Such an approach has revealed to be effective in minimizing earliness or lateness in drone landing times and payload assignments for small values of the number of UAVs, while a small decay in effectiveness is experienced for a large number of drones.

Further studies are needed to address additional aspects, such as the effect on performance of the interval between two landings. In addition, we plan to extend the proposed approach to more general scenarios, such as the case of multiple landing platforms and the possibility for drones to carry more than one payload or perform multiple landings.

Acknowledgment

The authors thank Marco Ghio for discussions on the problem statement and Tommaso Lanza for a very early version of the heuristic.

References

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, 2017.

[2] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, pp. 460–466, 2015.

[3] H. Gonzalez-Jorge, J. Martinez-Sanchez, M. Bueno, and P. Arias, "Unmanned aerial systems for civil applications: A review," *Drones*, vol. 1, no. 1, pp. 1–19, 2017.

[4] E. Ausonio, P. Bagnerini, and M. Ghio, "Drone swarms in fire suppression activities: A conceptual framework," *Drones*, vol. 5, no. 1, pp. 1–22, 2021, art. no. 17.

[5] M. Innocente and P. Grasso, "Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems," *Journal of Computational Science*, vol. 34, pp. 80–101, 2019.

[6] M. Akhloufi, A. Couturier, and N. Castro, "Unmanned aerial vehicles for wildland fires: Sensing, perception, cooperation and assistance," *Drones*, vol. 5, no. 1, 2021.

[7] A. Nemra and N. Aouf, "Robust cooperative uav visual slam," *Proc. 9th Int. Conf. on Cybernetic Intelligent Systems*, 2010.

[8] "Inspire company," https://www.inspire.flights, accessed: 2022-06-20.

[9] "Airobotics company," https://www.airoboticsdrones.com, accessed: 2022-06-20.

[10] M. Ghio, "Methods and apparatus for the employment of drones in firefighting activities," *US Patent*, no. US11104436B2, 2017.

[11] M. Wang and X. Chen, "System and method for managing unmanned aerial vehicles," *US Patent*, no. US20170190260A1, 2017.

[12] J. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft landings - The static case," *Transportation Science*, vol. 34, no. 2, pp. 180–197, 2000.

[13] A. Faye, "Solving the aircraft landing problem with time discretization approach," *European Journal of Operational Research*, vol. 242, no. 3, pp. 1028–1038, 2015.

[14] S. Ikli, C. Mancel, M. Mongeau, X. Olive, and E. Rachelson, "The aircraft runway scheduling problem: A survey," *Computers & Operations Research*, vol. 132, pp. 1–20, 2021, art. no. 105336.

[15] A. Salehipour, "An algorithm for single- and multiple-runway aircraft landing problem," *Mathematics and Computers in Simulation*, vol. 175, pp. 179–191, 2020.

[16] A. Salehipour, M. Modarres, and L. M. Naeni, "An efficient hybrid meta-heuristic for aircraft landing problem," *Computers & Operations Research*, vol. 40, no. 1, pp. 207–213, 2013.

[17] B. Girish, "An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem," *Applied Soft Computing*, vol. 44, pp. 200–221, 2016.

[18] C. Murray and A. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[19] N. Boysen, D. Briskorn, S. Fedtke, and S. Schwerdfeger, "Drone delivery from trucks: Drone scheduling for given truck routes," *Networks*, vol. 72, no. 4, pp. 506–527, 2018.