

de la Vega, Jonathan et al.

Article

Helicopter recovery in an oil and gas industry: Model and solution approaches

EURO Journal on Transportation and Logistics (EJTL)

Provided in Cooperation with:

Association of European Operational Research Societies (EURO), Fribourg

Suggested Citation: de la Vega, Jonathan et al. (2022) : Helicopter recovery in an oil and gas industry: Model and solution approaches, EURO Journal on Transportation and Logistics (EJTL), ISSN 2192-4384, Elsevier, Amsterdam, Vol. 11, Iss. 1, pp. 1-17,
<https://doi.org/10.1016/j.ejtl.2022.100084>

This Version is available at:

<https://hdl.handle.net/10419/325160>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc-nd/4.0/>



Helicopter recovery in an oil and gas industry: Model and solution approaches

Jonathan De La Vega^a, Thiago Vieira^a, Mateus Santana^a, Vitória Pureza^a, Reinaldo Morabito^{a,*}, Roberto Tavares^a, Yan Bastos^b, Paulo César Ribas^b

^a Department of Production Engineering, Federal University of São Carlos, 13565-905, São Carlos-SP, Brazil

^b Research and Development Center, PETROBRAS, Av. Horacio de Macedo, 950, Ilha do Fundão, Rio de Janeiro - RJ, Brazil

ARTICLE INFO

MSC:
00-01
99-00

Keywords:

Aircraft recovery problem
Flight rescheduling
Mixed integer programming
Heuristics
Oil and gas industry

ABSTRACT

Although optimization techniques have been successfully applied for a number of airline operations, aircraft recovery remains a challenge for both practitioners and researchers due to its complexity and the usual need of a quick response in practical settings. In this paper, we consider the case of a Brazilian oil and gas company that uses a heterogeneous fleet of helicopters for passenger transportation from a few mainland aerodromes to maritime units. The major difficulties in rescheduling delayed flights are resource limitations and realistic features of the company combined with safety and management practices. The problem consists of determining joint daily flight reschedules for all aerodromes that satisfy operational constraints and recovers all pending flights, while minimizing flight transfers among aerodromes, usage of helicopters and overall flight delays. We propose a mixed integer programming (MIP) model, aiming to appropriately represent the problem, and MIP-based local search and two-phase heuristics to cope with larger realistic problem instances. Computational results obtained with instances collected in the case study show the potential of these approaches to deal with this real-life aircraft recovery problem.

1. Introduction

Airline companies often face the inherent difficulty of flight rerouting and rescheduling (also known as aircraft recovery) when departure delays, airport close-down, temporary aircraft unavailability and other unexpected events occur. This is particularly true in the case of companies committed to offer a large number of flights, as the costs required to maintain good service levels under such circumstances can be very high. Aircraft recovery makes use of reassignment aircrafts to flights, delaying and even canceling flights, which in turn imply in additional airport taxes and fuel expenses, personnel overtime pay and possibly extra aircraft use. In the case of passenger airlines, there are also the onuses of hotel accommodation, refunds and reallocation of passengers to other flights. Thus, the recovery operation targets schedules with an acceptable trade-off between minimum delay, cancellation and costs. Moreover, the reschedules must be attained within an acceptable time frame.

The aircraft recovery problem (ARP) has attracted the attention of the OR community since the 1980s. The seminal work by Teodorović and Guberinić (1984) applies a branch-and-bound procedure to

reschedule a homogeneous fleet of aircrafts with minimum total passenger delays. As expected, subsequent papers added more realistic features such as the option of flight cancellation and the inclusion of airport curfew constraints (Teodorović and Stojković, 1990), aircraft swapping and the use of spare aircrafts (Jarrah et al., 1993), crew related constraints (Teodorović and Stojković, 1995), the use of ferry flights (Cao and Kanafani, 1997; Yan and Yang, 1996), heterogeneous fleet (Yan and Tu, 1997) and aircraft balance and maintenance (Arguello, 1997). Solution methods include branch-and-bound and branch-and-cut algorithms (Brunner, 2014), dynamic programming and column generation (Eggenberg et al., 2010), and metaheuristics (Lin and Wang, 2018; Hu et al., 2016).

A successful approach is proposed in Bard et al. (2001). The problem is transformed into a time-based network in which the time horizon is partitioned into intervals and the activities in each airport are aggregated during each interval. The ARP is then modeled as a minimum cost network flow problem with side constraints, often generating integral solutions with standard linear programming methods. A current trend in the literature is the inclusion of passenger and/or crew recovery

* Corresponding author.

E-mail addresses: jonathan.delavega@simplit-solutions.com (J. De La Vega), thiagozevieira@gmail.com (T. Vieira), mateus.95.santana@gmail.com (M. Santana), vmmpureza@gmail.com (V. Pureza), morabito@ufscar.br (R. Morabito), tavares@dep.ufscar.br (R. Tavares), yan.bastos@petrobras.com.br (Y. Bastos), paulo.ribas@petrobras.com.br (P.C. Ribas).

<https://doi.org/10.1016/j.ejtl.2022.100084>

Received 8 July 2021; Received in revised form 22 May 2022; Accepted 13 June 2022

Available online 21 June 2022

2192-4376/© 2022 The Author(s). Published by Elsevier B.V. on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

decisions (Arkan et al., 2016; Sinclair et al., 2014). Zhang et al. (2016) propose a multi-phase hybrid method that combines mathematical programming and different heuristics for solving the integrated aircraft and passenger recovery problem. The approach's first two phases consist, respectively, of solving a multi-commodity network flow model with a tight lower bound to select flight and maintenance arcs, and a greedy heuristic that constructs feasible rotations for the aircrafts based on the outcome of the flow model. The authors report that their algorithm found nearly 72% of the best known solutions of realistic test instances.

In this paper, we consider the short-term rescheduling problem faced by a Brazilian company engaged in the exploration, production, transportation, and commercialization of crude and processed oils and natural gas. The extraction of these raw materials occur mostly offshore, making the company responsible for providing the transportation from and to the mainland to the workers assigned to oil rigs and other maritime units (MUs). Transportation occurs whenever work shifts are about to start or finish, and it is carried out by a heterogeneous fleet of helicopters based on a few aerodromes operated by the company. There are several maritime units and flights and the daily reschedule must take into account the original timetable of each aerodrome and the departure aerodrome of each flight, the number of runways at the aerodromes and maritime units, the fleet of helicopters available in each aerodrome, postponement and shift regulations, flight departure priorities, aerodromes and helicopter time windows, among others. In particular, this ARP also considers pending flights transferred from previous days due to unexpected events, such as bad weather or aircraft mechanical failures, and with different recovering priorities, for rescheduling on the current day.

The present study extends previous recent studies in Vieira et al. (2021) and De La Vega et al. (2022) for the same oil and gas company, but where the ARP does not consider a set of aerodromes and possible flight transfers between these aerodromes. The solution approaches of these studies, based on continuous-time and discrete-time MIP models and ad-hoc heuristics, consider the particular case in which there is a single aerodrome, isolated or independent from the others, and all flights must depart from and return to this aerodrome. In addition, the approaches in De La Vega et al. (2022) consider constraints concerning different lease contracts of the helicopters, which are particular of that aerodrome and imply in different durations of the daily working hours of helicopters and mandatory stops for the crew's lunch breaks.

In order to tackle the present problem, a continuous-time MIP model considering multiple aerodromes is proposed, based on the single aerodrome continuous-time network flow model in De La Vega et al. (2022), which closely describes the present company's ARP. Due to its complexity and the usual need of a quick response in practical settings, we also propose a MIP-based local search heuristic and a two-phase heuristic, capable of coping with larger problem instances within acceptable computer runtimes. The performance of solving the model with general-purpose optimization software and the heuristics is assessed by means of realistic problem instances with data collected in a case study conducted at the company. Our aim in this work is to contribute to the practice of aerial passenger transportation and shows improvements to the company's flight recovery plans, highlighting the potential of the proposed approaches. Despite the specificities of our model, we believe that this study can also contribute to deal with problems of other oil and gas companies that routinely transport work teams to maritime units, such as offshore production platforms, drilling rigs, service units and special vessels. Similar situations arise in oil and gas companies operating in the North sea, Gulf of Mexico, West of Africa and Australia, for example.

The remainder of this paper is organized as follows. Section 2 presents a more detailed problem description. The MIP formulation and the heuristics are described in Sections 3 and 4, respectively. Section 5 discusses and analyzes the computational experiments and results from applying the solution approaches based on the model and the heuristics in realistic problem instances, followed by concluding remarks and the next steps of the research in Section 6.

2. Problem description

As stated before, in this study we consider the recovery of helicopter flights from/to aerodromes to/from maritime units, motivated by the ARP faced by an oil and gas company. As can be seen in Fig. 1, each aerodrome has its own available fleet of helicopters and the helicopters are heterogeneous in terms of travel cruise speed, capacity of passengers, flying range, etc. The company programmers previously schedule several daily flights (timetables) for each aerodrome and for each helicopter at the beginning of the week and, in the absence of unexpected events, these timetables should be followed. Each flight has its scheduled departure time in the day and it is basically a round-trip between an aerodrome and a given maritime unit (i.e., aerodrome-maritime unit-aerodrome). There may be more than one flight from the same aerodrome to the same maritime unit in a day. Each aerodrome helicopter can do several round-trip flights per day.

However, unexpected events are common in practice and may cause delays in the departure times of flights, changes in the assignment of helicopters to flights, changes in the departure aerodrome of the flights, and even the rescheduling of flights to the next day (in which case they are called *day-transferred flights*). The recovery operation implies in rescheduling the original timetables of the aerodromes, so that transferred flights of different priorities from previous days can be included into the current day plans, preferably with minimal disturbance to the other flights of the timetables. For this rescheduling, it is possible to reallocate helicopters and/or allocating additional helicopters of the same aerodrome in one day. It is also possible to change the departure aerodrome of a flight on the same day, although this is highly undesirable. In this case, the flight will use a helicopter from the other aerodrome with a capacity to transport the number of passengers and with autonomy (flying range) to fly until its maritime unit. Although the helicopters can be of different models, there are two main fleets of helicopters in each aerodrome: one called *normal* fleet, which include the helicopters originally assigned to the daily timetables, and the *pool* fleet, which are spare helicopters promptly available at the aerodrome that can be used with higher additional costs.

Each aerodrome has a single runway for helicopters taking off and landing, and each maritime unit can bear a single helicopter at a time as it has a single heliport. The ARP consists of determining joint daily flight reschedules for all aerodromes that satisfy operational constraints and recovers all pending flights transferred from the last days, while minimizing flight transfers between aerodromes, usage of helicopters and overall flight delay. The goal is to generate a suitable timetable for the aerodromes and their helicopters, indicating when each flight operated by which helicopter will depart during the day and which flights will be transferred to the following day. The result of this flight rescheduling (including the unrealized flights transferred to the next day) is known as the *recovery plan*, and it is usually required in short computer runtimes in practical settings (in the case of the company studied, in at most a few minutes).

There are two main categories of flights: *table* flights, that is, the ones originally scheduled in the timetables of the aerodromes, and the aforementioned *day-transferred* flights, which are the pending unrealized flights transferred from the last days. There are two other classes of high priority and less frequent flights, called *mandatory* and *entourage* flights. Mandatory flights are those that must be performed according to the original timetable, that is, they must not be changed to another aerodrome or transferred to the next day, whereas entourage flights are the ones used to transport managers and other representatives of the company for visits to the maritime units. These are special flights that, after landing at the heliport of a maritime unit, block it for the whole time spent by the entourage in the unit, commonly for the rest of the day. As the mandatory flights, the entourage flights must not be changed to another aerodrome, but differently from the mandatory flights, they can be transferred to the next day. We note that the rescheduling of table flights, entourage flights and mandatory

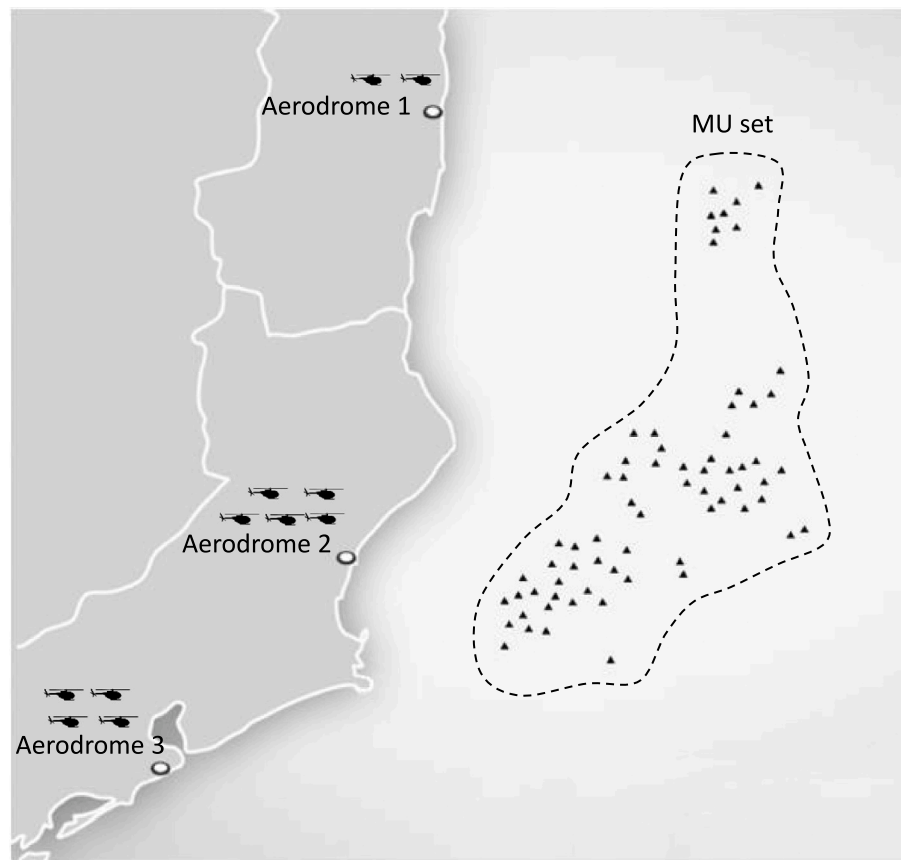


Fig. 1. Spatial representation of the studied problem with three aerodromes and a set of maritime units — MU.

flights can only delay the departures, but never anticipate them. Each flight departs from an aerodrome to a destination unit with a group of previously booked passengers and returns to the aerodrome from the unit with another group of previously booked passengers, or the same group in case of entourage flights. Therefore, any flight comprises a set of passengers, a planned departure time, the time required to reach the destination (or to go from the destination back to the aerodrome) and the service time required by the maritime unit to transfer passengers and prepare the aircraft to return. The recovery plan is not allowed to split or join flights, or change the set of passengers previously assigned to the flights in the original timetable. All aerodrome operations must occur during sunlight, forcing an operational time window for each helicopter. A table or entourage flight is transferred to the next day if it either cannot respect the operational time window, or its rescheduled departure time is more than 4 h from its original departure time.

Thus, at the beginning of each day (or sometimes more than once a day), the company programmers have to reschedule the daily timetable for the aerodromes taking into account the set of flights transferred from the last days. To generate feasible reschedules, the programmers can apply some proceedings, for example, assign pending flights to vacancy spots of the timetables, delay table flights to later departure times in the timetables, reallocate the aerodrome helicopters to the flights, transfer flights to other aerodromes, among others. Additional restraints should be considered for changing the aerodrome of the flight, for example, the land transfer time (by bus) of the passengers of the flight between the two aerodromes should not exceed some time limit (e.g., 4 h), here called *local-transfer*. There are some rules that have to be followed for this recovery plan. For instance, if the original timetable contains two or more flights from any origin (aerodrome) but to the same destination (maritime unit), the sequence of these flights in the timetable must be maintained. For example, if both flight A of an aerodrome and flight B of the same or other aerodrome go to

the same maritime unit, and if flight A is scheduled before flight B in the original timetable, then the recovery plan cannot schedule the departure of flight B before the departure of flight A. This precedence rule may, for example, force flight A to be allocated into the original departure slot of flight B, while transferring flight B to the next day.

There are also minimum time intervals between consecutive flight departures at the runway of each aerodrome (the departures must be spaced by, at least, 5 min) and minimum time intervals for service times of the flights at the maritime units (typically of 15 min), as well as minimum time intervals for ground preparation times between consecutive flights of the same helicopter at the aerodrome, called turnaround time — *tat* (e.g., 45 min). A simple representation of the elements of a single aircraft plan is depicted in Fig. 2. In this example, three flights departing from the same aerodrome with different destinations are assigned to a single aircraft.

As an aircraft is not allowed to spend the night in a maritime unit, all helicopters must return to the aerodromes before the end of the day. Moreover, the single heliport and other operational constraints forbid multiple helicopters to land on the maritime unit at the same time. Note that these constraints may affect the departure of different helicopters in different aerodromes, as presented in Fig. 3. In this example, aircraft A1 is based on aerodrome 1 and has two scheduled flights: one to maritime unit MU1 and the other to MU2. Aircraft A2 is based on aerodrome 2 and has one flight assigned to MU1. The gray area represents the delay inserted for the departure of aircraft A2 due to flight landing conflicts at maritime unit MU1. Without this delay, A2 would arrive at MU1 when the heliport of MU1 were occupied by aircraft A1.

During the requirement development phase of this research, the company programmers indicate some guidelines to identify better recovery plans. For example, flights transferred from the last days have higher priorities than table flights of the daily timetable, and helicopters of the normal fleet should be preferred for use rather than

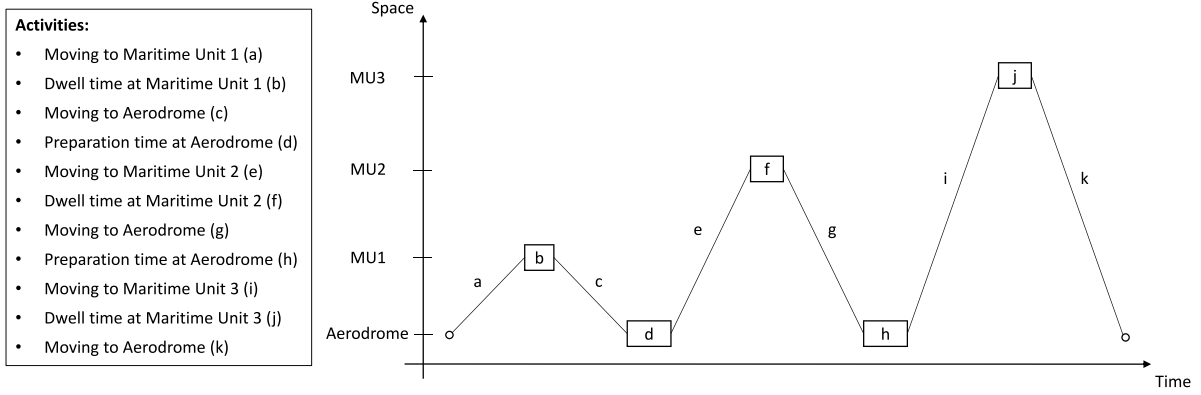


Fig. 2. A representation of a 3-departure flight plan of a single aircraft.

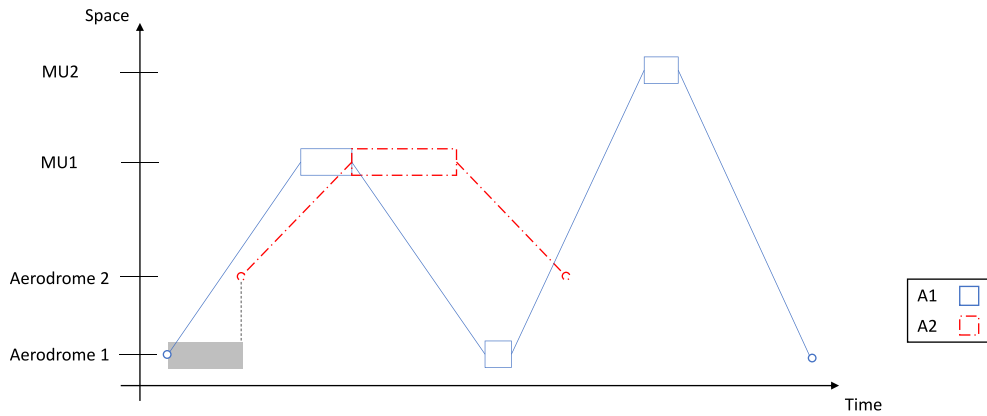


Fig. 3. A representation of the flight plans of two aircrafts A1 and A2 — the gray rectangle represents the delay on the departure of A2 to avoid conflicts at maritime unit MU1.

helicopters of the pool fleet. These and other guidelines were translated into the following objectives, here depicted in lexicographical order, in accordance with the recommendation of the company's programmers: (i) minimum number of unrealized entourage flights during the day, (ii) minimum number of unrealized flights during the day, which were transferred from the last two or more days, (iii) minimum number of unrealized flights during the day, which were transferred from the last day, (iv) minimum number of unrealized table flights during the day, (v) minimum number of table and day-transferred flights changed to a different aerodrome during the same day (mandatory and entourage flights cannot change the aerodrome), (vi) minimum number of pool helicopters used during the day, (vii) minimum number of normal helicopters used during the day, (viii) minimum overall delay of mandatory, entourage and table flights during the day, (ix) minimum overall delay of the other flights during the day. Formally, a set of performance indicators translates the quality of a daily recovery plan, as detailed in the weighted sum objective function of the model of the next section, as well as a formal description of this problem, including the necessary symbols and the mathematical formulation.

3. Mathematical formulation

In this section, we present a flow model to represent the present ARP. Let I , A , H , and U be the total number of flights, aerodromes, helicopters, and maritime units in the problem instance, respectively. The indices used in the model are as follows: i and j for the flights; k, ℓ for aerodromes; g, h for helicopters, and v for maritime units. The MIP model uses the following parameters and decision variables:

Parameters:

r_i Scheduled departure time of flight i .

s_{ih} Duration of flight i using helicopter h , including the dwell time at the destination maritime unit of this flight.

$t_{f_{ih}}$ Travel time of flight i from the aerodrome of helicopter h to the destination maritime unit of this flight.

dt_i Dwell time of flight i at its destination maritime unit.

$\tilde{t}$ Minimum time interval between consecutive take-offs of helicopters at the same aerodrome (the safety briefing).

$\hat{t}$ Minimum time interval between a landing and a take-off of the same helicopter at an aerodrome (the turnaround time).

$d^{\max}$ Maximum allowed delay for table, entourage and mandatory flights.

wd_h Work day of helicopter h .

$[b_k, \bar{b}_k]$ Time window of aerodrome k .

$tt_{k\ell}$ Land transfer time (by bus) of flight passengers between aerodromes k and ℓ .

$t^{\max}$ Maximum travel time (by bus) used to allow local-transfer of flights between pairs of different aerodromes.

u_i Destination maritime unit of flight i .

a_h Aerodrome of helicopter h .

a_i Aerodrome base of flight i (i.e., the aerodrome where flight i was scheduled in the original timetable).

$p_{ij} = 1$, if the departure of flight i from any aerodrome must be scheduled before the departure of flight j from any aerodrome, and 0, otherwise.

$c_{ih} = 1$, if helicopter h can be assigned (i.e., it is compatible in terms of capacity, flying range, etc.) to flight i , and 0, otherwise.

$$n_i = \begin{cases} 0, & \text{if flight is a table flight.} \\ 1, & \text{if flight is a transferred flight delayed by 1 day.} \\ 2, & \text{if flight is a transferred flight delayed by 2 or more days.} \\ 3, & \text{if flight is an entourage flight.} \\ 4, & \text{if flight is a mandatory flight.} \end{cases}$$

BigM Sufficiently large number (in our implementation, we set $BigM = \max_{k=1, \dots, A} \{\bar{b}_k\}$).

Decision Variables:

$X_{ijh} = 1$, if and only if flight j is performed immediately after flight i by helicopter h , and 0, otherwise.

$Y_{ih} = 1$, if and only if helicopter h is assigned to flight i , and 0, otherwise.

$Z_{ij} = 1$, if and only if the departure of flight i from any aerodrome is before the departure of flight j from any aerodrome, and 0, otherwise.

$V_h = 1$, if and only if helicopter h is used.

$DT_i \geq 0$, departure time of flight i .

$AT_i \geq 0$, arrival time of flight i .

$D_i \geq 0$, delay of flight i .

Before presenting the model, we highlight the following remarks:

- Two dummy flights 0 and $I + 1$ are created. We impose that any sequence of flights from a helicopter starts with dummy flight 0 and ends with dummy flight $I + 1$. On these dummy flights, $s_{0h} = s_{(I+1)h} = 0$, $\forall h = 1, \dots, H$ and $r_0 = r_{I+1} = 0$.
- The aerodrome index k is omitted in the decision variables because this information can be obtained from the helicopter index h and parameter a_h . We recall that the problem allows transferring flights between aerodromes, but not helicopters between aerodromes.
- The flow variable X_{ijh} is set to 0 if $c_{ih} + c_{jh} < 2$, which implies having no flow in the immediate sequence (i, j) of helicopter h if any or both of these flights i and j are not compatible with helicopter h .
- Some previously defined parameters are made available by the company in different units of time. However, before running the model, we convert all time parameters to minutes.

The MIP model for the present ARP is formulated as follows:

Objective function. As the present ARP involves different objectives with different priorities to be optimized, the objective function (1) uses a weighted sum method to minimize the sum of the penalties associated to the following terms: (f_1) flight transfers for the next day (except for mandatory flights); (f_2) flight transfers to aerodromes different from the flight base aerodrome; (f_3) use of helicopters; (f_4) flight delays. The vectors w^1 , w^2 , w^3 and w^4 establish the corresponding weights of each these terms and their relative importance in the present case study is such that: $w^1 > w^2 > w^3 > w^4$, as indicated by the company's programmers. The penalty of $w_i^1 \in w^1$ depends on the class of flight i (table, transferred from the last day, transferred from the last two or more days, entourage), the penalty of $w_{ik}^2 \in w^2$ depends on the distance

between the original base aerodrome of flight i and aerodrome k , the penalty of $w_h^3 \in w^3$ depends on the fleet and the model of helicopter h (normal, pool), and the penalty of $w_i^4 \in w^4$ depends on the class of flight i (table, transferred from the last day, transferred from the last two or more days, entourage, mandatory). Each of these penalty values were carefully defined by the company's programmers in order to reflect their relative priorities.

$$\text{Min } f = \sum_{p=1}^4 f_p, \quad \text{where} \quad f_1 = \sum_{i=1}^I w_i^1 \left(1 - \sum_{h=1}^H Y_{ih} \right), \quad (1)$$

$$f_2 = \sum_{i=1}^I \sum_{k=1}^A \sum_{h=1}^H w_{ik}^2 Y_{ih}, \quad f_3 = \sum_{h=1}^H w_h^3 V_h, \quad f_4 = \sum_{i=1}^I w_i^4 D_i.$$

Flow constraints. Constraints (2)–(4) define the flow constraints. If flight i is assigned to helicopter h , i.e., if $Y_{ih} = 1$, in a given solution, then constraints (2) and (3) ensure that there is an immediate predecessor flight (which can be the dummy flight 0) and another immediate successor flight (which can be the dummy flight $I + 1$) in the helicopter h schedule, respectively. Constraints (4) ensure the outflow and inflow of dummy flights 0 and $I + 1$ if helicopter h is used in the solution (i.e., if $V_h = 1$).

$$\sum_{i=0}^I X_{ijh} = Y_{jh}, \quad \forall j = 1, \dots, I; \quad h = 1, \dots, H. \quad (2)$$

$$\sum_{j=1}^{I+1} X_{ijh} = Y_{ih}, \quad \forall i = 1, \dots, I; \quad h = 1, \dots, H. \quad (3)$$

$$\sum_{i=1}^I X_{0ih} = \sum_{j=1}^I X_{j(I+1)h} = V_h, \quad \forall h = 1, \dots, H. \quad (4)$$

Assignment constraints. The assignment constraints are defined by constraints (5)–(8). Constraints (5) are related to the scheduling or not of the day-transferred and table flights, either at their base aerodrome or at a different one. Constraints (6) and (7) ensure that the entourage and mandatory flights, if scheduled in a solution, must depart from their base aerodromes, respectively. Note that the equality of constraints (7) forces the scheduling of mandatory flights in the solution — this is why there is no need for penalties associated with their transfer to the next day in the objective function. Finally, constraints (8) are used to avoid assigning flights to a helicopter h if this helicopter is not used in the solution (i.e., if $V_h = 0$).

$$\sum_{h=1}^H Y_{ih} \leq 1, \quad \forall i = 1, \dots, I; \quad n_i = 0, 1, 2. \quad (5)$$

$$\sum_{h=1}^H Y_{ih} \leq 1, \quad \forall i = 1, \dots, I; \quad n_i = 3. \quad (6)$$

$$\sum_{h=1}^H Y_{ih} = 1, \quad \forall i = 1, \dots, I; \quad n_i = 4. \quad (7)$$

$$Y_{ih} \leq V_h, \quad \forall i = 1, \dots, I; \quad h = 1, \dots, H. \quad (8)$$

Helicopter synchronization constraints. The synchronization of the flight departures made by the same helicopter is guaranteed by the set of constraints (9)–(12), which also act as sub-tour elimination constraints for the flow variables X_{ijh} . Constraints (9) impose a minimum time of $\hat{t}$ minutes between the arrival of flight i and the departure of flight j if both flights are performed consecutively by the same helicopter. This time is mainly associated with the helicopter inspection.

Constraints (10) and (11) avoid adding the $\hat{t}$ minutes of helicopter inspection when flights i and j are the dummy flights 0 and $I + 1$, respectively. Constraints (12) determine the arrival times of the flights in accordance with their departure times and assigned helicopters. Note that these constraints and constraints (13)–(14) (described later) ensure that the continuous variables DT_i , AT_i , and D_i assume the value 0 if flight i is not scheduled in a solution, i.e., if $\sum_{h=1}^H Y_{ih} = 0$. Therefore, the equality sign of constraints (12) is without loss of generality.

$$DT_j \geq AT_i + \hat{t}X_{ijh} - \text{BigM}(1 - X_{ijh}), \forall i, j = 1, \dots, I; i \neq j; h = 1, \dots, H. \quad (9)$$

$$DT_j \geq AT_0 - \text{BigM}(1 - X_{0jh}), \forall j = 1, \dots, I; h = 1, \dots, H. \quad (10)$$

$$DT_{(I+1)} \geq AT_i - \text{BigM}(1 - X_{i(I+1)h}), \forall i = 1, \dots, I; h = 1, \dots, H. \quad (11)$$

$$AT_i = DT_i + \sum_{h=1}^H s_{ih}Y_{ih}, \forall i = 1, \dots, I. \quad (12)$$

Scheduled departure time and maximum delay constraints. Constraints (13) impose the minimum departure times for flights and also determine the flight delays. Note that these constraints indicate that the flight departures can be delayed, but not advanced. Constraints (14) impose a maximum delay of $d^{\max}$ hours on the table, entourage and mandatory flights.

$$\sum_{h=1}^H r_i Y_{ih} \leq DT_i \leq \sum_{h=1}^H r_i Y_{ih} + D_i, \forall i = 1, \dots, I. \quad (13)$$

$$D_i \leq d^{\max} \sum_{h=1}^H Y_{ih}, \forall i = 1, \dots, I; n_i = 0, 3, 4. \quad (14)$$

Time window constraints. The imposition of aerodrome time windows is given by constraints (15) and (16). These constraints require that the take-off and landing of the flights scheduled in the solution, respectively, satisfy the minimum and maximum operating timetables of the aerodrome associated with the helicopter that performs them.

$$DT_i \geq \sum_{h=1}^H b_{-a_h} Y_{ih}, \forall i = 1, \dots, I. \quad (15)$$

$$AT_i \leq \sum_{h=1}^H \bar{b}_{a_h} Y_{ih}, \forall i = 1, \dots, I. \quad (16)$$

Helicopter work day constraints. The work day of the helicopters is imposed by constraints (17). For a given helicopter h , these constraints determine the elapsed time between its first take-off and its last landing and also ensure that this time does not exceed its work day of wd_h hours.

$$AT_j - DT_i \leq wd_h + \text{BigM}(2 - X_{0ih} - X_{j(I+1)h}), \forall h = 1, \dots, H. \quad (17)$$

Precedence constraints. Constraints (18)–(20) activate the variables Z_{ij} used to enforce the synchronization of the departures from the aerodrome of the flights, as well as their arrivals at the maritime units. Constraints (18) ensure at most an order of precedence between flights i and j , regardless of the aerodromes where they are scheduled in a solution. Constraints (19) guarantee that if flights i and j are scheduled in a solution, then there must be at least one order of precedence between them, i.e., the departure of flight i precedes the departure of flight j (in this case $Z_{ij} = 1$) or the departure of flight j precedes the departure of i ($Z_{ji} = 1$). Constraints (20) are complementary to constraints (19) and ensure the scheduling of flights i and j if there is an order of precedence between them, i.e., if $Z_{ij} = 1$ or $Z_{ji} = 1$. It is important to note that constraints (18)–(20) only guarantee the existence of one order of precedence between flights i and j if they are scheduled in the solution, but they do not define the order. Constraints (21) and (22) (described below) are responsible for defining this order,

which is the one whose departure times from the aerodrome of flights i and j incur the least penalty of the total delay.

$$Z_{ij} + Z_{ji} \leq 1, \forall i, j = 1, \dots, I; i \neq j. \quad (18)$$

$$Z_{ij} + Z_{ji} \geq \sum_{h=1}^H Y_{ih} + \sum_{h=1}^H Y_{jh} - 1, \forall i, j = 1, \dots, I; i \neq j. \quad (19)$$

$$2(Z_{ij} + Z_{ji}) \leq \sum_{h=1}^H Y_{ih} + \sum_{h=1}^H Y_{jh}, \forall i, j = 1, \dots, I; i \neq j. \quad (20)$$

Aerodrome synchronization constraints. The synchronization of the departure times of scheduled flights at the same aerodrome is guaranteed by constraints (21). In these constraints, if $Z_{ij} = 1$ and both flights i and j are scheduled at the same aerodrome k , then their departure times must be at least $\tilde{t}$ minutes apart.

$$DT_j - DT_i \geq \tilde{t}Z_{ij} - \text{BigM} \left(3 - Z_{ij} - \sum_{h=1, a_h=k}^H Y_{ih} - \sum_{h=1, a_h=k}^H Y_{jh} \right), \quad (21)$$

$\forall k = 1, \dots, A; i, j = 1, \dots, I; i \neq j.$

Unit maritime synchronization constraints. Constraints (22) ensure the synchronization of the arrival times of the flights at a maritime unit v . In Fig. 4, we illustrate how the imposition of the time interval $(tf_{ih} + dt_i - tf_{jg})$ ensure this synchronization. In this figure, we assume that flight i is assigned to helicopter h at aerodrome k , while flight j is assigned to helicopter g at aerodrome ℓ (we note that helicopters h and g can be associated to the same aerodrome; in this case aerodrome k coincides with aerodrome ℓ). Moreover, we also assume that the departure of flight i from aerodrome k occurs before the departure of flight j from aerodrome ℓ . Observe that the $(tf_{ih} + dt_i - tf_{jg})$ interval in Fig. 4 prevents flight j from landing on the maritime unit while its heliport is occupied by flight i . It is worth mentioning that the synchronization of the arrival times of the flights in the maritime unit can be done only using the departure times of the flights from their aerodromes because the flight travel times are considered deterministic.

$$DT_j - DT_i \geq (tf_{ih} + dt_i - tf_{jg})Z_{ij} - \text{BigM}(3 - Z_{ij} - Y_{ih} - Y_{jg}), \quad (22)$$

$\forall v = 1, \dots, U; i, j = 1, \dots, I; i \neq j; u_i = u_j = v; h, g = 1, \dots, H.$

Other practical constraints: Constraints (23)–(25) are necessary to consider other practical characteristics of the company problem. The set of constraints (23) ensures that flight i (if scheduled in a solution) is not carried out by a helicopter incompatible with it. Constraints (24) impose an order of precedence between flights i and j . Thus, in a given solution and regardless of the aerodromes where these flights are scheduled, the departure of flight j will never occur before the departure of flight i if $p_{ij} = 1$. The set of constraints (25) prevents the local-transfer of flights between different aerodromes if the travel time by bus between these aerodromes exceed $t^{\max}$ units of time.

$$Y_{ih} \leq c_{ih}, \forall i = 1, \dots, I; h = 1, \dots, H. \quad (23)$$

$$Z_{ji} = 0, \forall i, j = 1, \dots, I; i \neq j; p_{ij} = 1. \quad (24)$$

$$Y_{ih} = 0, \forall i = 1, \dots, I; h = 1, \dots, H; a_i \neq a_h; tt_{a_i, a_h} > t^{\max}. \quad (25)$$

Domain of the decision variables. Constraints (26)–(30) indicate the type and domain of the decision variables. From these constraints, we observe that variables X_{ijh} , Y_{ih} , Z_{ij} and V_h are binaries, while variables DT_i , AT_i and D_i are non-negative continuous.

$$X_{ijh} \in \{0, 1\}, \forall i = 0, \dots, I; j = 1, \dots, I + 1; i \neq j; h = 1, \dots, H. \quad (26)$$

$$Y_{ih} \in \{0, 1\}, \forall i = 1, \dots, I; h = 1, \dots, H. \quad (27)$$

$$Z_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, I; i \neq j. \quad (28)$$

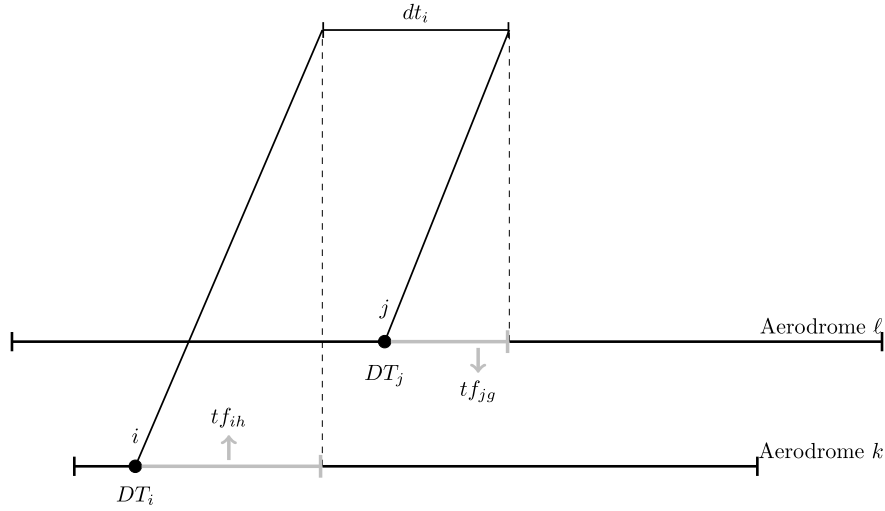


Fig. 4. Illustration of the synchronization of flights at a maritime unit.

$$V_h \in \{0, 1\}, \forall h = 1, \dots, H. \quad (29)$$

$$DT_i \geq 0, AT_i \geq 0, D_i \geq 0, \forall i = 0, \dots, I + 1. \quad (30)$$

4. Heuristic solution approaches

This section first presents the MIP-based local search heuristic and then, the two-phase heuristic.

4.1. MIP-based local search

Given an initial feasible assignment vector of helicopters to flights $\bar{Y}$, our local search algorithm explores its neighborhood $\mathcal{N}(\bar{Y})$ using a MIP solver to solve the flow model (1)–(30) with some fixed assignment variables and within a computer time limit. More precisely, at each iteration of the algorithm, the MIP solver solves the flow model with a limited space of feasible solutions given by:

$$\begin{aligned} \mathcal{N}(\bar{Y}) = \{Y : Y_{ih} = \bar{Y}_{ih}, \forall i \in S; |S| = c; h = 1, \dots, H \\ \text{and } Y \text{ is a feasible assignment of helicopters to flights} \\ \text{according to model (1)–(30)}\} \end{aligned} \quad (31)$$

where S defines a subset of flights and c is its cardinality.

Our MIP-based local search algorithm is resumely described in Algorithm 1. This algorithm requires a priori knowledge of an initial feasible assignment of helicopters to flights, which is determined using the MIP model of the last section or any constructive heuristic procedure (Mitrović-Minić and Punnen, 2010). There are seven parameters involved in Algorithm 1: c is the cardinality of S ; Δc is the downward step size to decrease the value of c ; t is the time limit for the MIP solver; Δt is the upward step size of t ; v is the iteration count; no_best is the count of consecutive iterations without improvements in the incumbent solution, and $iter_Max$ is the minimum number of consecutive iterations without improvements in the incumbent solution to adjust parameters c and t . Parameters Δc and Δt are used to dynamically reduce and increase parameters c and t , respectively. If parameter c is reduced, the size of the neighborhood $\mathcal{N}(\bar{Y})$ that defines the space of feasible solutions increases, so that the difficulty of solving the MIP model by the solver also increases. This is why we also increase the time limit t of the solver in Δt in order to better explore the space of feasible solutions of the MIP model.

Lines 3–4 of the algorithm are the initialization lines. Note that at the beginning of the algorithm, the cardinality of set S is equal to

the total number of flights, i.e., $c = I$. This strategy helps the MIP solver to find feasible solutions quickly in the first iteration, especially in large problem instances. In line 6, the flow model is solved within the time limit t . In line 8, it is verified if the current solution is better than the incumbent solution. In this case, it is updated and no_best count is set to 1. Otherwise, in line 12, no_best count is increased by 1. Parameters c and t are adjusted in line 14 if no improvements in the incumbent solution are found after $iter_Max$ consecutive iterations. In the next line, set S is updated and the new neighborhood $\mathcal{N}(\bar{Y})_v$ is generated. In line 16, no_best count is set to 1. Finally, the iteration count v of the algorithm is incremented in line 17. This procedure is applied repeatedly until the termination criterion is met.

Algorithm 1: MIP-based Local Search

```

1 Input:  $c, \Delta c, t, \Delta t, v, no\_best, iter\_Max$ 
2 Output: Incumbent solution  $(\hat{f}, \hat{X}, \hat{Y}, \hat{Z}, \hat{V}, \hat{DT}, \hat{AT}, \hat{D})$ 
3 Set  $c := I$ ,  $v := 0$  and  $\hat{f} := +\infty$ 
4 Update the set  $S_v$ ,  $|S| = c$  and generate the neighborhood  $\mathcal{N}(\bar{Y})_v$ 
5 do
6   Solve the model (1)–(30) with space of feasible solutions  $\mathcal{N}(\bar{Y})_v$  within the time limit  $t$ .
7   Let  $(f^v, X^v, Y^v, Z^v, V^v, DT^v, AT^v, D^v)$  the obtained solution
8   if  $f^v < \hat{f}$  then
9     Let  $(\hat{f}, \hat{X}, \hat{Y}, \hat{Z}, \hat{V}, \hat{DT}, \hat{AT}, \hat{D}) := (f^v, X^v, Y^v, Z^v, V^v, DT^v, AT^v, D^v)$ 
10    Let  $no\_best := 1$ 
11   else
12     Let  $no\_best := no\_best + 1$ 
13   if  $no\_best > iter\_Max$  then
14     Let  $c = c - \Delta c$  and  $t = t + \Delta t$ 
15     Update the set  $S_v$ ,  $|S| = c$  and generate the neighborhood  $\mathcal{N}(\bar{Y})_v$ 
16     Let  $no\_best := 1$ 
17   Let  $v := v + 1$ 
18 while the stopping criterion is not satisfied;

```

An important issue of Algorithm 1 is the selection of the c flights in S . To do this, we use the savings $saving_i$ of the flights defined as follows:

$$saving_i = \frac{w_i^1}{s_i} \quad (32)$$

where $\bar{s}_i = \sum_{h=1}^H \frac{s_{ih}}{H}$. For each flight i , $saving_i$ defines the savings per time unit obtained if flight i is scheduled in the solution. Note that the flights with the highest weights w_i^1 and lowest average travel times are more attractive. Finally, to determine the flights in S , we simply list the flights in increasing order of savings and select the first c flights from that list. We chose this strategy because it produced better results than the strategy of selecting the flights with the c greatest savings.

4.2. Two-phase heuristic

This proposed heuristic comprises two main phases. The first phase starts by decomposing the problem into A disjoint subproblems, each characterized by aerodrome $k = 1, \dots, A$ and flights $i \in I_k$ in which $I_k = \{i = 1, \dots, I : a_i = k\}$, i.e., all flights that depart from aerodrome k in the original schedule. Each subproblem k is then solved by a construction heuristic followed by a sequence of local searches. We name this phase *Decompose-Construct-and-Improve*. If the first phase solution does not serve all flights, the second phase, called *Integration*, tries to allocate them to helicopters, used or not, based on a different aerodrome.

The method shares the same theoretical heuristic framework presented in Vieira et al. (2021). In that work, a single aerodrome is considered, the crew's working hours are disregarded and flight times are independent of the helicopter type. In addition to extending the problem to the case of multiple aerodromes, our heuristic does not apply the strategy of relaxing the fleet heterogeneity as seen in Vieira et al. (2021) since the current configuration of the operation requires prior knowledge of the helicopters to determine their schedules (there are non-scalar parameters that depend on identifying the helicopter to be used). Next, we describe the two parts that form *Decompose-Construct-and-Improve Phase*, *Construction Part* and *Improvement Part*.

4.2.1. The construction part

For each subproblem (aerodrome) k , sequences of flights $i \in I_k$ are constructed and allocated to helicopters based on k . The construction of flight sequences requires assessing the number of helicopters that serve as many flights as possible and, given the fleet heterogeneity, choosing a particular subset of helicopters in H_k in which $H_k = \{h = 1, \dots, H : a_h = k\}$, i.e., all helicopters based on aerodrome k in the original schedule. Given the difficulty of assessing the required number of helicopters, we start with its lower bound as:

$$H_k^{min} = \left\lceil \frac{\sum_{i \in I_k} [\hat{t} + \min_{h \in H_k} (s_{ih})]}{\min \left\{ \bar{b}_k - \underline{b}_k, \max_{h \in H_k} (wd_h) \right\}} \right\rceil. \quad (33)$$

and iteratively add more helicopters, if necessary. Note that the computation of H_k^{min} assumes that all flights are served and disregards any flight superposition in the aerodromes runways. Thus, the number of unique combinations of H_k^{min} helicopters is given by:

$$Cn = \binom{|H_k|}{H_k^{min}} = \frac{|H_k|!}{H_k^{min}! (|H_k| - H_k^{min})!}. \quad (34)$$

In order to choose an appropriate subset of H_k , each helicopter h is evaluated according to function $gw_h = I - nH_h + (I - nG_h)/10 + 100\hat{s}_h/wd_h + w_h^3$, where nH_h is the number of flights compatible with h , nG_h is the number of flights to which h is assigned in the original schedule and $\hat{s}_h$ is h 's average duration flight time considering its compatible flights. The score of a given combination $Comb_l$ of helicopters is given by $score_l = \sum_{h \in Comb_l} gw_h, \forall l = 1, \dots, Cn$. Through the non-decreasing ordering of $score_l$, the G first subsets of helicopters are selected to be checked in function of the best performance that will be observed by the allocation of flights during the rest of the *Construction Part*. We defined these subsets as $SH_g, \forall g = 1, \dots, G$ (note that $|SH_g| = H_k^{min}$).

A two-flight sequence Pr_h is then constructed for each helicopter $h \in SH_g$. The start of the assembly of each sequence is done in

relation to the due time $pz_i = r_i + \hat{s}_i$, where $\hat{s}_i$ is the average duration time of flight i . Specifically, the first and the last flight of Pr_h is chosen by $i' = \operatorname{argmin}_{\{i \in I_k : i \notin Pr_h, v \in SH_g, v \neq h\}} \{pz_i | c_{ih} = 1\}$ and $j' = \operatorname{argmax}_{\{j \in I_k : j \notin Pr_h, v \in SH_g, v \neq h\}} \{pz_j | c_{jh} = 1\}$, respectively. This corresponds to following an EDT (Earliest Due Time) rule ordering. Regarding the remaining helicopter $(|H_k| - H_k^{min})$, we set $Pr_h = \emptyset, \forall h \in H_k | h \notin SH_g, g = 1, \dots, G$.

As a way of verifying feasibility, carrying out the schedule, as well as quantifying the delay of this first flight association, the heuristic executes backward programming, called *GetSchedule*, responsible for making all scheduling from flight sequences. In general, *GetSchedule* delays the flights of each Pr_h only when necessary, in order to meet the problem's constraints aimed at building schedules. For this, the routine starts the departure and arrival times of each flight as if they were planned (without delays). Based on this, there is a looping designed to "push" flights that violate the conditions related to the overlapping of helicopter in a MU (dt_i), safety briefing ($\bar{t}$), turnaround time of the same consecutive helicopter ($\hat{t}$), and calculating the arrival duration time of a flight to the aerodrome (s_{ih}). The closing of this looping occurs when all these requirements are guaranteed, no matter how big the values of DT_i and AT_i may be in this first moment.

Then, the routine quantifies all delays D_i and verifies the feasibility condition of the rest constraints related to the schedule, i.e., checking if any table flight delays more than what is tolerated ($D_i > d^{max}$), if the landing of any flight of the schedule does not occur with sunlight ($AT_i > \bar{b}_k$), if there is a flight j that has a scheduled departure time before a flight i for the same MU, given by parameter $p_{ij} = 1$, and if there is some Pr_h in which the difference between the arrival time of the last flight and the departure time of the first flight is greater than wd_h . In view of this inspection, *GetSchedule* also returns a binary parameter entitled *feasibleTime*, having a value of 1, if all checks confirm the feasibility of the analyzed schedule, and 0, otherwise.

For every unscheduled flight, its insertion is analyzed before and after the position of each scheduled flight ($j \in Pr_h$) using *GetSchedule*. The criterion for including a flight in the schedule is made when evaluating its type (whether it is a day-transfer flight or not), the level of precedence over other flights, and the degree of compatibility with the available helicopter, determined by:

$$(i^*, j^*, h^*) = \operatorname{argmin}_{(i,j) \in I_k : i \notin Pr_h, j \in Pr_h, h \in SH_g} \left\{ fType_i + fPrec_i + \frac{nH_i}{H} + \min(TD_{ijh}, TD_{jih}) \right\}. \quad (35)$$

where:

- $fType_i = \begin{cases} 1, n_i \in \{1, 2\}; \\ 2, n_i \in \{0, 3, 4\}; \end{cases}$
- $fPrec_i = \begin{cases} 2I, nM_i = 0; \\ I/nM_i, \text{otherwise}; \end{cases}$
- nM_i : number of flights that are precedent (not necessarily immediately) to flight i ;
- nH_i : number of helicopters compatible with flight i ;
- TD_{ijh} : total delay obtained if flight i is inserted immediately before j (TD_{jih} , if the insertion is after).

This procedure is called *InsertFlights* and it is repeated until all unscheduled flights from the aerodrome k are included, or the circumstance arises in which it is no longer possible to insert flights into the schedule (all the desired insertions have the return of *GetSchedule*, *feasibleTime* = 0). For the second situation, the remaining unscheduled flights from aerodrome k are placed in the set defined by R_{gk} .

Therefore, the objective function for aerodrome k in iteration it is calculated by:

$$OF_{it,k} = \min_{g=1, \dots, G} f_{it,k,g}. \quad (36)$$

where:

$$f_{it,k,g} = \sum_{i \in R_{gk}} w_i^1 + \sum_{h \in SH_g} w_h^3 + \sum_{h \in SH_g} \sum_{i \in Pr_h} w_i^4 D_i.$$

If $OF_{(it-1),k} > OF_{it,k}$ and $H_k^{min} < |H_k|$, the iteration is increased and the values of $OF_{it,k}$ and solution $sol_{it,k}(X, DT, AT, D)$ are stored (for iteration $it = 1$, we initialize $OF_{0,k} = +\infty$). The heuristic proceeds by doing $H_k^{min} = H_k^{min} + 1$ and repeating the steps that determine SH_g . It makes the first flight insertions using the EDT ordering and tries to designate the remaining flights using the *InsertFlights* routine by the criterion (35) and the calculation of $OF_{it,k}$. This cycle is performed until the comparison between the objective functions and the availability of the fleet is considered false, which implies a convergence to the best result found ($OF_{(it-1),k^*}$), thus finishing the *Constructive Part*.

4.2.2. The improvement part

The *Decompose-Construct-and-Improve Phase* completion consists of applying five different local searches to the solution found in the previous step, as follows:

1. N1. Helicopter itinerary rescheduling;
2. N2. Exchange of scheduled flights by unscheduled flights;
3. N3. Inter-helicopter flight insertion;
4. N4. Inter-helicopter flight exchange;
5. N5. Intra-helicopter flight exchange.

Neighborhood N1 aims at accommodating unscheduled flights at the cost of some extra delay time. It consists of replacing a scheduled flight i by an unscheduled one in the same order of visitation, followed by i 's reinsertion in a different order of visitation or in the itinerary of a distinct helicopter. Neighborhood N2 aims to reduce the solution cost by replacing a scheduled flight by an unscheduled one with a lower day-transfer penalty as described in the objective function of the optimization model. For instance, serving day-transferred flights is less costly than table flights. Note that differently from N1, N2 does not alter the cardinality of the unscheduled flights set. Neighborhood N3, in turn, consists of transferring a flight i from the itinerary of the helicopter that serves it followed by i 's reinsertion into the itinerary of a different helicopter. Neighborhood N4 tries to swap two flights from two different helicopter schedules. It is basically an inter-change of flights between different helicopters maintaining the precedent and subsequent positions. Finally, Neighborhood N5 consists of adjacent pairwise interchanges of flights in the same helicopter itinerary. In other words it performs rearrangement of flights from the same helicopter. This is performed for all flights of all helicopters.

Local searches N1 to N5 are applied consecutively, following the lexicographic order. For a given local search, the neighborhood is fully investigated, the feasible move that provides the largest improvement in the objective function is selected to produce a new incumbent solution, and the process is repeated until no improving move is found.

4.2.3. Integration phase

As a last resort for incorporating flights in the solution, the second phase aims at transferring flights from their original aerodrome to a different one. For all unscheduled flights i , originally assigned to departure from aerodrome k , we compute the feasible insertion positions of the helicopters' itinerary that are compatible with i and based on aerodrome $k' \neq k$. The insertion that results in the lowest delay is selected to produce the new incumbent solution, and the process is repeated until no improving insertion is found. Note that the required time for the passengers' displacement from the original aerodrome to the new one must be taken into account when assessing the feasibility of the insertion.

In case the procedure fails in incorporating one or more flights, we apply a second local search whose neighborhood is similar to N1 described in the *Improvement Part*. A scheduled flight i is replaced by

an unscheduled i' in the same order of visitation, followed by the reinsertion of i in a different order of visitation or in the itinerary of a distinct helicopter. The move that results in the lowest delay is selected to produce the new incumbent solution, and the process is repeated until no improving insertion is found. If this attempt fails, a third local search is applied, whose move consists of assigning an unused compatible helicopter (if there is) to i' .

The general scheme of the heuristic is presented by Algorithm 2. The heuristic starts with the *Decompose-Construct-and-Improve Phase*, which comprises Steps 4 to 21. For each aerodrome, the *Construction Part* and *Improvement Part* are executed in sequence. Basically, the *Construction Part* is responsible for quantifying and determining the aircraft to be used through the best score $f_{it,k,g}$ found, and also defining the sequences and schedules of flights for each aircraft, thus storing the results $OF_{it,k}$ and $sol_{it,k}(X, DT, AT, D)$. As mentioned earlier, after obtaining a feasible solution for the first phase, the *Improvement Part* is triggered, which can be seen in Steps 19 to 21. As long as there is improvement in the incumbent solution, the five local searches are performed. Next, the *Integration Phase* tries to allocate flights that were not scheduled at their original aerodromes, according to Steps 23 to 27.

Algorithm 2: General algorithm of the heuristic method

```

1 begin
2   read instance;
3   // Decompose-Construct-and-Improve Phase
4   foreach aerodrome  $k = 1, \dots, A$  do
5     Construction Part
6     calculate  $H_k^{min}$ ;
7     determine the subset  $SH_g, \forall g = 1, \dots, G$ ;
8     initialize:  $it \leftarrow 1, OF_{0,k} \leftarrow +\infty, improvement \leftarrow 1$  (an
        auxiliary variable);
9     do
10      for  $g = 1$  to  $G$  do
11        make the preliminary flight sequencing by EDT
            rule;
12        for all the remaining unscheduled flights, use the
            InsertFlights procedure;
13        calculate and store  $f_{it,k,g}$ ;
14        choose the best result of  $f_{it,k,g}$  and compute it as
             $OF_{it,k}$ , saving the corresponding
             $sol_{it,k}(X, DT, AT, D)$ ;
15         $H_k^{min} \leftarrow H_k^{min} + 1$ ;
16         $it \leftarrow it + 1$ ;
17      while  $OF_{(it-1),k} > OF_{it,k} \wedge H_k^{min} < |H_k|$ ;
18    Improvement Part
19    while  $improvement = 1$  do
20      execute the improvement local searches N1-N5;
21      if  $OF_{it,k}$  does not reduce more then  $improvement = 0$ ;
22  // Integration Phase
23  repeat
24    for all unscheduled flights, try to insert them at other
        aerodromes (if there are compatible helicopters and inland
        travel time between aerodromes  $\leq t^{max}$ );
25    for all unscheduled flights, try an insertion based on
        neighborhood N1;
26    if there are still remaining flights to assign and helicopter
        available, try to activate the most compatible helicopter due
        to unscheduled flights, executing together with
        neighborhood N3 and N4;
27  until all flights were scheduled, all compatible helicopter were
        assigned, or the number of unscheduled flights has not decreased;
28  generate output;
```

5. Computational experiments

This section presents the results of the numerical experiments to evaluate the computational performance of four solution approaches:

Table 1
Characteristics of the realistic problem instances.

Instance	I	H	U	A	No_{D0}	No_{D-1}	No_{D-2}	No_E	No_M	$NoVars_{int}$	$NoVars_{cont}$	$NoConst$
123_2	23	7	17	2	15	6	2	0	0	5833	75	5885
135_2	35	12	22	2	23	9	0	3	0	19660	111	17998
148_3	48	18	32	3	35	9	2	1	1	50969	150	33556
156_3	56	18	40	3	27	21	3	2	3	68401	174	44176
167_3	67	21	46	3	35	22	3	2	5	111043	207	75305
171_3	71	21	50	3	40	25	2	0	4	124195	219	102820
182_3	82	24	56	3	46	27	3	0	6	185581	252	153879
190_3	90	24	52	3	49	31	4	1	5	222389	276	186423
182_2	82	21	56	2	37	35	4	0	6	164158	252	129223
190_2	90	18	52	2	39	39	6	1	5	171047	276	143204

(i) solve flow model (1)–(30) using the solver CPLEX 12.8 with default parameters (called M1); (ii) apply the two-phase heuristic of Section 4.2 (called H1); (iii) solve model M1 via CPLEX using the solution of heuristic H1 as a starting feasible solution (called H1M1); (iv) apply the MIP-based local search heuristic of Section 4.1 using the flight-helicopter assignments of the solution of heuristic H1 as starting assignments (called H1LS). All these solution approaches were implemented in C++ language and the experiments were carried out on a PC Intel core i7-8700K CPU with 16 GB of memory and operational system Ubuntu 16.04.

Table 1 resumes the main characteristics of ten realistic problem instances used in the computational experiments, ranging from 23 to 90 flights, from 7 to 24 helicopters, from 17 to 59 maritime units and from 2 to 3 aerodromes. In this table, column “Instance” indicates the instance name, while the next four columns present the numbers of flights (I), helicopters (H), maritime units (U) and aerodromes (A), respectively. The next five columns present the numbers of table flights, transferred flights from the last day, transferred flights from the last two or more days, entourage flights and mandatory flights, respectively. The first eight instances used real data provided by the company and the last two (182_2 and 190_2) were generated based on instances 182_3 and 190_3. These instances were built as a means to simulate the important unexpected event that one of the aerodromes (in the case, the one with the fewest flights) is closed for the whole day and its respective demand is transferred to the other aerodromes. To do this, helicopters from the closed aerodrome are disregarded and their types of booked flights, table and transfer delayed by one day flights, become, respectively, transfer delayed by one day and transfer delayed by two or more days flights. Table 1 highlights the differences between 182_2 and 182_3, and between 190_2 with 190_3. From now on, the first eight instances of Table 1 are called *real* instances, whereas the last two ones are called *simulated* instances. We also use Table 1 to show for each instance the number of binary variables (column $NoVars_{int}$), the number of continuous variables (column $NoVars_{con}$), and the number of constraints (column $NoConst$) generated by the solver CPLEX 12.8 when using approach M1. This information provides an idea of the problem size and the challenge of our solution approaches to solve the problem, especially in large instances with several thousands of binary variables and constraints.

All parameter values correspond to the values used by the company. For example, $\hat{t}$ was set to 45 min, $\tilde{t}$ to 5 min, and $d^{\max}$ and $t^{\max}$ to 4 h (240 min). The remaining data of all instances, such as the scheduled departure times of the flights (r_i), the duration of flight i using helicopter h (s_{ih}), the service times of the flights at the maritime units (dt_i), the work days of the helicopters (wd_h), the time windows of the aerodromes ($[b_k, \bar{b}_k]$), the land transfer times (by bus) of flight passengers between aerodromes k and ℓ ($tt_{k\ell}$), among others, can be obtained upon request to the authors. These parameters are in hours:minutes format but we convert them to minutes before using any solution approach. Each of the weight values of the model objective function were carefully defined by the company’s programmers in order to reflect their relative priorities. These chosen penalties for $w^1 > w^2 > w^3 > w^4$ used in the experiments were: w_i^1 equals 360 if flight i is an

entourage flight, 240 if it is a transferred flight from the last two or more days, 180 if it is a transferred flight from the last day and 80 if it is a table flight; w_{ik}^2 equals 40 if flight i is transferred to aerodrome k when it was originally scheduled in another aerodrome; w_h^3 equals 20 if helicopter h is of the pool fleet and 10 if it is of the normal fleet; w_i^4 equals 0.001 if flight i is a table, entourage or mandatory flight and 0.005 if it is a day-transferred flight.

The MIP-based local search run with the following parameters values: $\Delta c = [0.3I]$, $t = 300$ seconds, $\Delta t = 100$ seconds and $iter_Max = 2$. In iteration 0, parameter p was set to I and thus, all binary variables Y_{ih} are fixed in the first iteration of the local search. The stop criterion of this heuristic was either a proven optimal by CPLEX (in a neighborhood of parameter c fixed to 0), or the reached time limit of 3600 s. Regarding the two-phase heuristic, parameter G (the number of aircraft subsets in *Constructive Part of Decompose-Construct-and-Improve Phase*) was set to 3 for all cases.

5.1. Toy problem

Before presenting the results of the real problem instances, let us consider an illustrative example with a daily timetable of only 9 table flights ($I = \{i_1, i_2, \dots, i_9\}$ with $n_1 = n_2 = \dots = n_9 = 0$) and 2 aerodromes ($A = \{k_1, k_2\}$). Fig. 5 depicts the corresponding Gantt diagram for the scheduled timetable, where the flights are represented by colored bars: the orange ones represent the travel times to and from each maritime unit, tf_{ih} ; the yellow ones represent the time that each helicopter remains on the ground at each maritime unit, dt_i ; the gray ones represent the turnaround time of each helicopter just after each flight at the aerodrome, $\hat{t}$. The departure times of each flight, r_i , are shown over these bars. The flights depart from the two aerodromes, represented on the left-side of the figure, which have different time windows, i.e., $b_1 = 06:55$, $b_2 = 07:00$, and, $\bar{b}_1 = 19:00$, $\bar{b}_2 = 17:30$, respectively. The flight labels are indicated inside the orange bars of the figure. Three helicopters are used ($h = h_1, h_2, h_3$) in this timetable, represented along the ordinate axis of the figure, with a work day of 10 h ($wd_1 = wd_2 = wd_3 = 10$). Note that helicopters h_1 and h_2 belong to aerodrome k_1 while helicopter h_3 to aerodrome k_2 . There are 7 maritime units ($u = PA, PB, \dots, PG$) and note that there are flights with same unit destination (e.g., flights i_4 and i_6 goes to unit PD , i.e., $P_4 = P_6$). The maritime unit labels are indicated inside the yellow bars of the figure.

Now, let us consider that four unrealized flights ($i_{10}, i_{11}, i_{12}, i_{13}$) were transferred from the last days to aerodrome k_1 , two of them were 2-day delay flights (i_{11}, i_{12} with $n_{11} = n_{12} = 2$), and the other two were 1-day delay flights (i_{10}, i_{13} with $n_{10} = n_{13} = 1$). The unit destination of flights i_{10}, i_{11} coincides with the destination of flights i_4, i_6 (i.e., PD); the destination of flight i_{12} coincides with the destination of flight i_1 (PA); the destination of flight i_{13} coincides with the destination of flight i_7 (PF). Therefore, these 4 day-transferred flights should be scheduled before the table flights because of the precedence constraints (i.e., $p_{10,4} = p_{10,6} = p_{11,4} = p_{11,6} = p_{12,1} = p_{13,7} = 1$). Fig. 6 depicts a recovery plan after the inclusion of these day-transferred flights in the previous daily timetable.

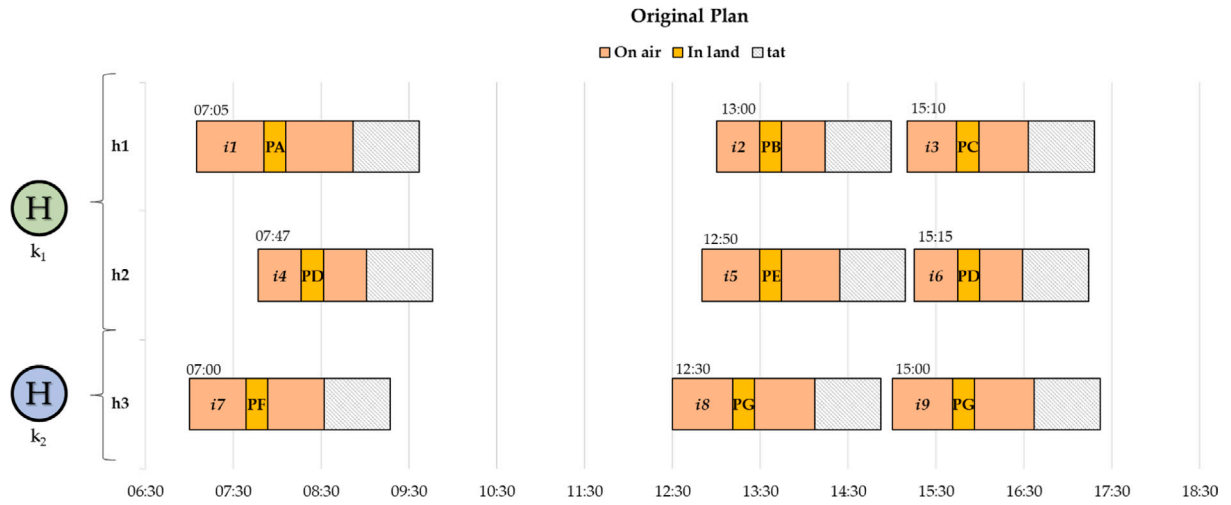


Fig. 5. Original timetable of the toy problem.

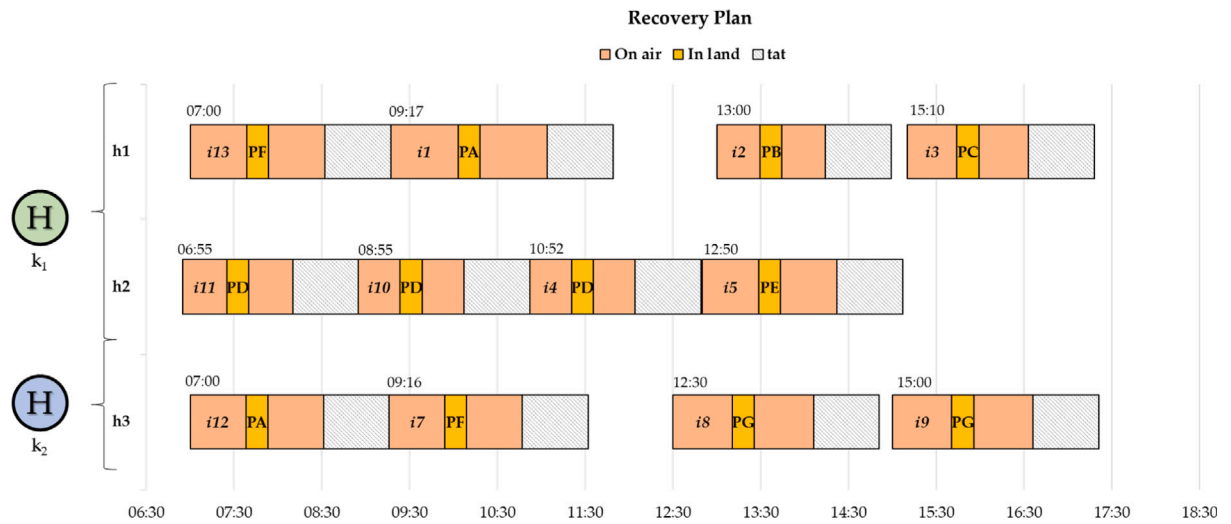


Fig. 6. Recovery plan of the toy problem.

As shown in Fig. 6, the four day-transferred flights were successfully rescheduled in the recovery plan, which required some decisions to be taken. First, the precedence constraints were met for the flights with the same unit destination — note that flights i_{11} and i_{10} were rescheduled before flight i_4 (in the same aerodrome), and flight i_{12} was scheduled before flight i_1 (in this case, in different aerodromes). For this, the departure times of flights i_1, i_4, i_7 had to be postponed in a few hours, implying in some delays ($D_1, D_4, D_7 > 0$). Moreover, flight i_{12} , which was previously scheduled in aerodrome k_1 , had to change to aerodrome k_2 . Given that the departure of flight i_{12} must occur before the departure of flight i_1 , the scheduling of flight i_{12} in helicopter h_1 of aerodrome k_1 would imply in delaying flight i_1 in, at least, $s_{1,1} + \hat{t}$ minutes, which is greater than the time limit for a delay (d^{max}). Thus, flight i_1 would be transferred to the next day, which would be highly penalized. Similarly, given that the departure of flight i_{12} must occur before the departure of flight i_4 , the scheduling of flight i_{12} in helicopter h_2 at aerodrome k_1 would imply in delaying flight i_4 for more than the time limit of a delay and, hence, flight i_4 would be transferred to the next day. A better alternative is to move flight i_{12} from aerodrome k_1 to aerodrome k_2 and use helicopter h_3 . Despite this recovery plan implying a delay for flight i_7 , it is able to include more flights in the current day.

It should be observed that flight i_6 was unrealized in the recovery plan of Fig. 6 and had to be transferred to the next day. Note that

this flight could not be maintained as the last flight of helicopter h_2 , because of the work day constraint of 10 h of this helicopter. Similarly, if flight i_6 were assigned to helicopters h_1 or h_3 , the work day limit of these helicopters would be violated as well and hence, transferring flight i_6 to the next day seems to result in a good solution. In fact, the recovery plan of Fig. 6 is indeed an optimal solution for this problem, which is proven by solving to optimality model M1. Table 2 shows the details of approach M1 solution, together with the solution obtained by the heuristic approach H1. Column “Solution Approach” indicates the solution approach; column f the total solution penalty; columns f_1, f_2, f_3, f_4 the penalties of the four terms of the objective function, respectively; column “Time” the computer runtime (in seconds); column nR the number of flights transferred to the next day; column nT the number of flights that change the aerodrome; column nH the number of helicopters used. While the solution of M1 is optimal, the solution obtained by heuristic H1 is very close to the optimal value. This difference is due to a slightly higher overall flight delay (column f_4). On the other hand, the heuristic solution is obtained within a much shorter computational time, and this advantage of H1 over M1 will become more evident when solving the realist problem instances of the next section.

Table 2
Summary of the results of the toy problem.

Solution approach	f	f_1	f_2	f_3	f_4	Time (s)	nR	nT	nH
M1	151.053	80	40	30	1.053	51.24	1	1	3
H1	151.93	80	40	30	1.93	0.09	1	1	3

5.2. Real problem instances

Table 3 presents the results obtained by the four solution approaches for each of the real problem instances. Some of the table columns are the same as Table 2, except for the column “Instance” that indicates the instance name; column Gap^1 , the optimality gap (in percentage) with respect to the best lower bound LB^{M1} for each instance found by solution approach M1 after one hour of execution time; column Gap^2 , the gap (in percentage) with respect to the best upper bound UB^{IN} for each instance. The values of columns Gap^1 and Gap^2 were determined as:

$$\text{Gap}^1 = (1 - LB^{M1}/f) * 100. \quad (37)$$

$$\text{Gap}^2 = (1 - UB^{IN}/f) * 100. \quad (38)$$

Some of the main results of this table are:

- The solution approach H1 was the fastest to generate feasible solutions, less than 170 s on average. Moreover, these solutions are of good quality as the obtained recovery plans were able to reschedule all pending flights of the previous days, as well as all flights of the day, which is the most important criterion of the company.
- The solution approach H1LS was the one that found the best feasible solutions within the time limit of one hour. This means that the local search is able to improve the solutions of approach H1, 0.5% on average. These improvements are not significant and are basically related to reductions in flight delays, which is the less penalized term of the objective function.
- The solution approach M1 was able to find feasible solutions without transferring flights to the next day until problem instance **I48_3**. However, for the remaining larger instances, the obtained recovery plans are not able to include all flights within the time limit of one hour, which is highly undesirable by the company. The high values of column Gap^1 show that the solver cannot produce tight lower bounds using the flow model under this time limit.
- Although the solution approach H1M1 improved the solutions obtained by approach H1 within the time limit of one hour, it is dominated by approach H1LS, which produces better feasible solutions.

Next, we compare the solutions in more detail obtained by approaches H1 and M1 in each of the real instances. The other two approaches were unconsidered in this comparison as their solutions differ from the ones of approach H1 only in terms of small flight delays, but their runtimes are much longer. In the H1 solution for instance **I23_2**, all available helicopters were used, yielding in a total penalty of 82.520 (only 0.08% larger than the one of approach M1). The running time was only 0.56 s and the difference of its solution value was in the flight delays. In instance **I35_2**, two helicopters available in the first aerodrome are not used in the solutions of both approaches. The total penalty of approach H1 is only 0.42% higher than the one of approach M1 and its runtime was 7.35 s. In instance **I48_3**, where the number of aerodromes increases from two to three, the solution of H1 schedules all flights using 14 out of 18 available helicopters and requiring 16.51 s. The solution of M1, on the other hand, needs one more helicopter to schedule all flights and its value is worse than approach H1 (4.61%), even using one hour of runtime.

The difficulty of solving the problem increases considerably for the next instances with more flights. In instance **I56_3**, although the recovery plan of H1 does not use all available helicopters of the second and third aerodromes, all available helicopters of the first aerodrome are used and one of its scheduled flights had to be transferred to the second aerodrome, resulting in a total penalty of 263.36. However, the H1 solution is still able to reschedule all flights of all aerodromes under a runtime of 52.29 s, while the M1 solution transfers 4 flights to the next days. The difference between the two solution values is over 72.05%. The H1 solutions for instances **I67_3** and **I71_3** also reschedule all flights and the runtimes increase to 148.46 and 80.79 s, respectively. The relative deviations between the solution values of H1 and M1 increase considerably. This behavior is also observed when solving the largest instances, **I82_3** and **I90_3**, with 82 and 90 flights and three aerodromes. All these flights are rescheduled in the recovery plans of H1, although the runtimes jump to around 500 s. Nevertheless, these runtimes are still acceptable for practical purposes as pointed out by the company programmers. These results indicate that H1 can be a suitable approach for use in practical settings of the present ARP. The recovery plans of approach M1, on the other hand, become very ineffective as the problem instances increase, involving transference of several flights to the next day, as in instances **I82_3** and **I90_3**.

5.3. Simulated problem instances

As previously mentioned, we also consider a simulated pessimistic scenario where the aerodrome with the fewest flights closes for the whole day due to an unexpected event, for example, very bad weather. In this case, all scheduled flights in the timetable of the closed aerodrome have to be either changed to the other aerodromes, or transferred to the next day. For this simulation, we consider instances **I82_2** and **I90_2** with only two aerodromes. These are supposed to be difficult problem instances as the recovery plans have to deal with very congested situations. Table 4 reports the results obtained with the four solution approaches for these simulated instances. The columns of this table are the same as Table 3.

Note in Table 4 that, as expected, the recovery plans of H1 did not reschedule all flights, even using all available fleets in the aerodromes. All mandatory, entourage and day-transferred flights from the last days were included in the recovery plans, as they are priority flights, and the flights transferred to the next day are only table flights (i.e., 5 table flights in instance **I82_2** and 12 flights in instance **I90_2**). The required runtimes are around 500 s, which are tolerable for supporting decisions in this ARP. The recovery plans produced by approaches H1M1 and H1LS have slightly lower total penalties than the ones of approach H1 and their gains are only in terms of overall flight delays, at the expense of requiring much higher runtimes. Similarly to the results with the real instances, approach H1LS was the one that provides the smallest total penalties, while approach M1 was, by far, the one that provides the largest values.

5.4. Sensitivity analysis for H1

In addition to the experiments with the real-life instances of Section 5.2 (called here scenario “Real”), we perform a sensitivity analysis on the objective function terms of the problem as a way to evaluate the robustness of the solutions obtained from heuristic method H1 (scenario “SA”). As the CPLEX solutions of model M1 had high optimality gaps, the sensitivity analysis was focused only on heuristic H1. Thereby, in scenario SA, we perform different computational tests making changes on the objective function weights in accordance with the weight-family groupings: $\mathcal{F}_1 = \{0, 1, 2, 3\}$ (types of flights only alluding to $n_i = 0, 1, 2, 3$, i.e., without mandatory flights $n_i = 4$); $\mathcal{F}_2 = \{\text{local-transfer}\}$ (transference of flights between different aerodromes); $\mathcal{F}_3 = \{\text{normal, pool}\}$ (types of helicopters); and $\mathcal{F}_4 = \{\text{pre-scheduled, unscheduled}\}$ (referring to the types of flights, where

Table 3
Summary of the results of the real problem instances.

Instance	Solution approach	f	f_1	f_2	f_3	f_4	Time (s)	Gap ¹ (%)	Gap ² (%)	nR	nT	nH
123_2	M1	82.442	0	0	80	2.442	3600.00	37.10	0.00	0	0	7
	H1	82.520	0	0	80	2.520	0.56	37.16	0.09	0	0	7
	H1M1	82.451	0	0	80	2.451	3600.00	37.11	0.01	0	0	7
	H1LS	82.442	0	0	80	2.442	3600.00	37.10	0.00	0	0	7
135_2	M1	141.321	0	0	140	1.321	3600.00	49.89	0.03	0	0	10
	H1	141.920	0	0	140	1.920	7.35	50.11	0.45	0	0	10
	H1M1	141.285	0	0	140	1.285	3600.00	49.88	0.00	0	0	10
	H1LS	141.279	0	0	140	1.279	3600.00	49.88	0.00	0	0	10
148_3	M1	173.577	0	0	170	3.577	3600.00	42.11	5.08	0	0	15
	H1	165.580	0	0	160	5.580	16.51	39.32	0.49	0	0	14
	H1M1	164.904	0	0	160	4.904	3600.00	39.07	0.08	0	0	14
	H1LS	164.765	0	0	160	4.765	3600.00	39.02	0.00	0	0	14
156_3	M1	942.394	640	80	210	12.394	3600.00	90.40	72.15	4	2	17
	H1	263.360	0	40	200	23.360	52.29	65.66	0.34	0	1	16
	H1M1	262.595	0	40	200	22.595	3600.00	65.56	0.05	0	1	16
	H1LS	262.454	0	40	200	22.454	3600.00	65.54	0.00	0	1	16
167_3	M1	1238.710	880	120	230	8.710	3600.00	93.50	79.94	0	0	19
	H1	250.390	0	0	230	20.390	148.46	67.87	0.75	0	0	19
	H1M1	249.699	0	0	230	16.699	3600.00	67.78	0.48	0	0	19
	H1LS	248.509	0	0	230	18.509	3600.00	67.62	0.00	0	0	19
171_3	M1	1504.296	1120	120	250	14.296	3600.00	93.33	82.45	9	3	20
	H1	267.200	0	0	240	27.200	80.79	62.43	1.17	0	0	20
	H1M1	266.599	0	0	240	26.599	3600.00	62.35	0.95	0	0	20
	H1LS	264.073	0	0	240	24.073	3600.00	61.99	0.00	0	0	20
182_3	M1	2031.295	1680	80	260	11.295	3600.00	95.06	84.74	17	2	21
	H1	311.000	0	0	290	21.000	425.83	67.71	0.34	0	0	23
	H1M1	309.946	0	0	290	19.946	3600.00	67.60	0.00	0	0	23
	H1LS	309.941	0	0	290	19.941	3600.00	67.60	0.00	0	0	23
190_3	M1	3588.337	3360	0	210	18.337	3600.00	96.92	90.88	29	0	18
	H1	329.070	0	0	300	29.070	589.22	66.45	0.55	0	0	23
	H1M1	327.960	0	0	300	27.960	3600.00	66.33	0.21	0	0	23
	H1LS	327.255	0	0	300	27.255	3600.00	66.26	0.00	0	0	23

Table 4
Summary of the results of the simulated problem instances.

Instance	Solution approach	f	f_1	f_2	f_3	f_4	Time (s)	Gap ¹ (%)	Gap ² (%)	nR	nT	nH
182_2	M1	5368.933	5200	0	150	18.933	3600.00	98.50	87.17	45	0	14
	H1	690.200	400	0	260	30.200	348.45	88.35	0.17	5	0	21
	H1M1	689.096	400	0	260	29.096	3600.00	88.33	0.01	5	0	21
	H1LS	689.051	400	0	260	29.051	3600.00	88.33	0.00	5	0	21
190_2	M1	6888.938	6720	0	160	8.938	3600.00	97.82	82.13	51	0	14
	H1	1232.860	960	0	230	42.860	499.60	87.81	0.13	12	0	18
	H1M1	1231.280	960	0	230	41.280	3600.00	87.79	0.03	12	0	18
	H1LS	1230.933	960	0	230	40.933	3600.00	87.79	0.00	12	0	18

the pre-scheduled flights are in set $\{0, 3, 4\}$ and the unscheduled flights are in $\{1, 2\}$. These groupings are based on the penalties f_1 to f_4 and they are used to distinguish the values that are different among the weights w_i^1 , w_{ik}^2 , w_h^3 and w_i^4 defined at the beginning of Section 5.

Given these groupings, the first set of tests consists of setting each penalty value to 0. Therefore, we define $\text{Test}^1(f_p)$: $w^p = 0$ for each $p = 1, \dots, 4$. The second set of tests is in charge of leveling the values for each weight-family grouping. We choose by the average values acquired from each F_p with $p \neq 2$ (since all $w_{ik}^2 = 40$ for $k \neq a_i$, which results in scenario Real). Let $\bar{w}^1 = 200$, $\bar{w}^3 = 15$ and $\bar{w}^4 = 0.003$ be the average values of F_1 , F_3 and F_4 , respectively; hence, we have $\text{Test}^2(F_p)$: $w^p = \bar{w}^p$ for each $p \in \{1, 3, 4\}$. The third set of tests cancels (i.e., makes null) the weights in an accumulated way, starting with the elements in F_4 and going up to those in F_1 , in order to maintain the stability of the objective function. The details of the third set of tests are given below (we discard $\text{Test}_2^3(F_4)$ since $\text{Test}_2^3(F_4) = \text{Test}^1(f_4)$):

- $\text{Test}_1^3(\{\text{pre-scheduled}\})$: make all weights $w_i^4 = 0$ if $n_i \in \{0, 3, 4\}$;
- $\text{Test}_1^3(\text{Test}_2^3, \{\text{normal}\})$: together with the determined values from Test_2^3 , make all weights $w_h^3 = 0$ if the type of helicopter h is normal;
- $\text{Test}_4^3(\text{Test}_3^3, F_3)$: together with the determined values from Test_3^3 , make penalty $f_3 = 0$;
- $\text{Test}_5^3(\text{Test}_4^3, F_2)$: together with the determined values from Test_4^3 , make penalty $f_2 = 0$;
- $\text{Test}_6^3(\text{Test}_5^3, \{0\})$: together with the determined values from Test_5^3 , make all weights $w_i^1 = 0$ if $n_i = 0$;
- $\text{Test}_7^3(\text{Test}_6^3, \{0, 1\})$: together with the determined values from Test_6^3 , make all weights $w_i^1 = 0$ if $n_i \in \{0, 1\}$;
- $\text{Test}_8^3(\text{Test}_7^3, \{0, 1, 2\})$: together with the determined values from Test_7^3 , make all weights $w_i^1 = 0$ if $n_i \in \{0, 1, 2\}$;
- $\text{Test}_9^3(\text{Test}_8^3, F_1)$: together with the determined values from Test_8^3 , make all weights $w_i^1 = 0$ if $n_i \in F_1$.

Table 5Result variations between scenarios SA and Real for instance **I90_3**.

Test description	ΔTime (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-140.35	0	0	2	6	1	0	0	0	-3	0	-1130	-744	9	0	-3	-1874
Test ¹ (f_2)	142.02	0	0	0	0	0	0	0	0	0	0	-409	291	0	0	0	-118
Test ¹ (f_3)	134.8	0	0	0	0	0	0	0	0	1	0	-401	-145	0	0	1	-546
Test ¹ (f_4)	-119.79	0	0	0	0	0	0	0	0	0	0	1561	-761	0	0	0	800
Test ² (F_1)	126.17	0	0	0	0	0	0	0	0	0	0	-409	291	0	0	0	-118
Test ² (F_3)	138.43	0	0	0	0	0	0	0	0	1	-1	-406	308	0	0	0	-98
Test ² (F_4)	-46.39	0	0	0	0	0	0	0	0	0	0	1518	-1564	0	0	0	-46
Test ³ ((pre-scheduled))	-24.65	0	0	0	0	0	0	0	0	0	0	-324	645	0	0	0	321
Test ³ (Test ³ , {normal})	-117.5	0	0	0	0	0	0	0	0	0	0	1561	-761	0	0	0	800
Test ³ (Test ³ , F_3)	-112.6	0	0	0	0	0	0	0	0	1	-1	1562	-823	0	0	0	739
Test ³ (Test ³ , F_2)	-126.51	0	0	0	0	0	0	0	0	1	-1	1562	-823	0	0	0	739
Test ³ (Test ³ , {0})	-124.74	0	0	0	0	1	0	0	0	1	-2	1429	-516	1	0	-1	913
Test ³ (Test ³ , {0, 1})	-93.91	0	0	0	2	1	0	0	0	1	-3	1088	-321	3	0	-2	767
Test ³ (Test ³ , {0, 1, 2})	-93.61	0	0	0	2	1	0	0	0	1	-3	1088	-321	3	0	-2	767
Test ³ (Test ³ , F_1)	-104.13	0	0	0	2	1	0	0	0	1	-3	1088	-321	3	0	-2	767

Table A.6Result variations between scenarios SA and Real for instance **I23_2**.

Test description	ΔTime (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-0.29	0	0	0	1	1	0	0	-1	0	-124	124	2	0	-1	0	
Test ¹ (f_2)	-0.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Test ¹ (f_3)	-0.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Test ¹ (f_4)	-0.37	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8	
Test ² (F_1)	-0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Test ² (F_3)	-0.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Test ² (F_4)	-0.37	0	0	0	0	0	0	0	0	0	138	-155	0	0	0	-17	
Test ³ ((pre-scheduled))	0.38	0	0	0	0	0	0	0	0	0	0	11	0	0	0	11	
Test ³ (Test ³ , {normal})	0.05	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8	
Test ³ (Test ³ , F_3)	0.1	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8	
Test ³ (Test ³ , F_2)	0.03	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8	
Test ³ (Test ³ , {0})	-0.2	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280	
Test ³ (Test ³ , {0, 1})	-0.16	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280	
Test ³ (Test ³ , {0, 1, 2})	-0.2	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280	
Test ³ (Test ³ , F_1)	-0.22	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280	

For each instance, we perform the sensitivity analysis using all these tests and heuristic H1. The results from these experiments are presented in the tables of Appendix A. They are shown in the form of deviations (variations), i.e., differences between the values of scenarios SA and Real. Among the results, we select the ones of instance **I90_3** to detail here (Table 5) because this is the largest instance representing the most complex operating situation. The columns of Table 5 are: **Test description** (name of the proposed test); ΔTime (s) (variation of the computational time in seconds); ΔR_M (variation of the total number of mandatory flights moved or transferred to the next day); ΔR_E (variation of the total number of entourage flights moved to the next day); ΔR_{D-2} (variation of the total number of 2 day-transfer or more flights moved to the next day); ΔR_{D-1} (variation of the total number of 1 day-transfer flights moved to the next day); ΔR_{D0} (variation of the total number of table flights moved to the next day); ΔT_{A1} (variation of the total number of flights that left aerodrome A1 and went to another using a local-transfer); ΔT_{A2} (variation of the total number of flights that left aerodrome A2 and went to another using a local-transfer); ΔT_{A3} (variation of the total number of flights that left aerodrome A3 and went to another using a local-transfer); ΔH_P (variation of the total number of type-pool helicopters used); ΔH_N (variation of the total number of type-normal helicopters used); ΔDL_2 (variation of the total delay of unscheduled flights, in minutes); ΔDL_1 (variation of the total delay of pre-scheduled flights, in minutes); ΔnR (variation of the total number of all flights moved to the next day); ΔnT (variation of the total number of all flights that changed the aerodrome); ΔnH (variation of the total number of all used helicopters); and, ΔnDL (variation of the total delay of all flights, in minutes).

In the first set of tests, starting with Test¹(f_1), there was an increase in the number of flight transfers to the next day and a decrease in the use of aircraft and delays in scenario SA with respect to scenario Real

(note that a positive value in the table means an increase in scenario SA over scenario Real, while a negative value means a decrease). Once the weights w^1 were annulled, the optimization bias becomes the local-transfers, fleet usage and flight delay times, which causes the cancellation of flights to achieve these goals. In Test¹(f_2), the penalties f_1 , f_3 and f_4 are prioritized to the detrimental of f_2 . In this case, the heuristic solution did not result in an increase in the local-transfers, however, $w^2 = 0$ allowed a new flight assignment to aircraft, reducing delays. Test¹(f_3) tends to lead to a greater use of aircraft and a smaller measure of local-transfers and delays. The solution obtained revealed a greater use of type-pool aircraft to improve the total delay. In Test¹(f_4), the heuristic only made the flight delay worse, which was expected since making $w^4 = 0$ does not generate major perturbations in the method due to the presence of the maximum tolerated delay as a hard constraint.

Going to the second set of tests, leveling the weights of family F_1 in Test²(F_1) implies changing the solutions that have rejected or canceled flights, i.e., variations related to the types of flights. Since the heuristic solutions do not reject flights, one would expect no changes in day-transfers and fleet usage. Test²(F_2) allowed the heuristic to change the used helicopters, achieving a slight decrease in the total delay. Defining the same value for the weights w^3 varies the delays between pre-scheduled flights and those to be scheduled. The heuristic increased the delay for unscheduled flights and decreased it for pre-scheduled flights, being able to improve the total delay.

In the third set of tests, which consists of gradually turning off the weights, starting with the lowest value and going up to the highest value, one would expect that the solution method to be used would have a standard behavior of worsening the metric corresponding to the weight-off to try to improve others that are on. As the best solution for scenario Real does not contain rejected flights and no local-transfer, we

Table A.7Result variations between scenarios SA and Real for instance **I35.2**.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-3.56	0	0	0	2	3	0	0	-3	0	-141	-218	5	0	-3	-359	
Test ¹ (f_2)	-1.18	0	0	0	0	0	0	0	-1	0	-79	-254	0	0	-1	-333	
Test ¹ (f_3)	-1.27	0	0	0	0	0	0	0	0	-1	-79	-287	0	0	-1	-366	
Test ¹ (f_4)	-3.17	0	0	0	0	0	0	0	-1	0	25	155	0	0	-1	180	
Test ² (F_1)	-1.15	0	0	0	0	0	0	0	-1	0	-79	-254	0	0	-1	-333	
Test ² (F_3)	-1.03	0	0	0	0	0	0	0	0	-1	-79	-287	0	0	-1	-366	
Test ² (F_4)	-2.52	0	0	0	0	0	0	0	-1	0	2	-371	0	0	-1	-369	
Test ³ ((pre-scheduled))	-1.69	0	0	0	0	0	0	0	-1	0	-79	254	0	0	-1	175	
Test ³ (Test ³ , {normal})	-2.64	0	0	0	0	0	0	0	-1	0	25	155	0	0	-1	180	
Test ⁴ (Test ³ , F_3)	-2.53	0	0	0	0	0	0	0	0	-1	25	-197	0	0	-1	-172	
Test ⁴ (Test ³ , F_2)	-2.52	0	0	0	0	0	0	0	0	-1	25	-197	0	0	-1	-172	
Test ⁵ (Test ³ , {0})	-1.86	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435	
Test ⁵ (Test ³ , {0, 1})	-2.46	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435	
Test ⁵ (Test ³ , {0, 1, 2})	-2.31	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435	
Test ⁹ (Test ³ , F_1)	-2.23	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435	

Table A.8Result variations between scenarios SA and Real for instance **I48.3**.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-4.42	0	0	1	3	0	0	0	0	-2	0	-205	50	4	0	-2	-155
Test ¹ (f_2)	-5	0	0	0	0	0	0	0	0	0	1	-259	-230	0	0	1	-489
Test ¹ (f_3)	-4.73	0	0	0	0	0	0	0	0	1	3	-349	-1101	0	0	4	-1450
Test ¹ (f_4)	-5.83	0	0	0	0	0	0	0	0	0	1	155	-439	0	0	1	-284
Test ² (F_1)	-4.82	0	0	0	0	0	0	0	0	0	1	-259	-230	0	0	1	-489
Test ² (F_3)	-4.46	0	0	0	0	0	0	0	0	1	0	-270	-246	0	0	1	-516
Test ² (F_4)	-6.16	0	0	0	0	0	0	0	0	0	1	47	-466	0	0	1	-419
Test ³ ((pre-scheduled))	-4.25	0	0	0	0	0	0	0	0	0	1	-337	768	0	0	1	431
Test ³ (Test ³ , {normal})	-6.42	0	0	0	0	0	0	0	0	0	1	155	-439	0	0	1	-284
Test ⁴ (Test ³ , F_3)	-6.33	0	0	0	0	0	0	0	0	1	0	174	-503	0	0	1	-329
Test ⁴ (Test ³ , F_2)	-5.78	0	0	0	0	0	0	0	0	1	0	174	-503	0	0	1	-329
Test ⁵ (Test ³ , {0})	-6.32	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
Test ⁵ (Test ³ , {0, 1})	-5.9	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
Test ⁵ (Test ³ , {0, 1, 2})	-6.34	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
Test ⁹ (Test ³ , F_1)	-7.02	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797

Table A.9Result variations between scenarios SA and Real for instance **I56.3**.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-23.27	0	0	1	1	2	-1	0	0	-2	1	-127	-335	4	-1	-1	-462
Test ¹ (f_2)	-23.36	0	0	0	0	1	-1	0	0	-1	1	13	-237	1	-1	0	-224
Test ¹ (f_3)	-21.92	0	0	0	0	1	-1	0	0	0	2	13	-502	1	-1	2	-489
Test ¹ (f_4)	-30.66	0	0	0	0	0	0	0	0	-1	1	1377	-106	0	0	0	1271
Test ² (F_1)	-22.55	0	0	0	0	0	0	0	0	-1	1	39	224	0	0	0	263
Test ² (F_3)	-6.55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Test ² (F_4)	9.9	0	0	0	0	0	0	0	0	-1	1	1097	-1040	0	0	0	57
Test ³ ((pre-scheduled))	-3.06	0	0	0	0	0	0	0	0	0	0	-86	296	0	0	0	210
Test ³ (Test ³ , {normal})	-30.48	0	0	0	0	0	0	0	0	-1	1	1377	-106	0	0	0	1271
Test ⁴ (Test ³ , F_3)	-29.81	0	0	0	0	0	0	0	0	-1	1	1377	-106	0	0	0	1271
Test ⁴ (Test ³ , F_2)	-30	0	0	0	0	0	0	0	0	-1	1	1377	-106	0	0	0	1271
Test ⁵ (Test ³ , {0})	-30.23	0	0	0	0	2	-1	0	0	-1	0	1377	-131	2	-1	-1	1246
Test ⁵ (Test ³ , {0, 1})	-30.67	0	0	0	1	1	-1	0	0	-1	0	1044	44	2	-1	-1	1088
Test ⁵ (Test ³ , {0, 1, 2})	-30.13	0	0	0	1	1	-1	0	0	-1	0	1044	44	2	-1	-1	1088
Test ⁹ (Test ³ , F_1)	-30.07	0	0	0	1	1	-1	0	0	-1	0	1044	44	2	-1	-1	1088

conclude that the heuristic generated coherent solutions, only worsening the metrics with weight-off, as shown in the corresponding rows from Test³((pre-scheduled)) to Test⁹(Test³, F_1).

6. Concluding remarks

This study deals with a short-term helicopter recovery problem in the oil and gas industry. It was motivated by the case of a Brazilian company that employs a heterogeneous fleet of helicopters and round-trip flights for daily passenger transportation between onshore aerodromes and offshore units. We proposed a detailed MIP model that appropriately represents this ARP with all the specific characteristics of its real settings, but which is unable to provide good recovery plans

for realistic problem instances within acceptable runtimes using the CPLEX solver. Therefore, we also proposed MIP-based local search and two-phase heuristics able to produce effective recovery plans for the previously scheduled daily timetables of these instances requiring relatively short runtimes, and which can be suitable for successful application in practice.

The results obtained from the proposed solution approaches indicate some interesting perspectives for future research, such as a more complete and detailed analysis of the benefits and impacts of the use of these heuristic approaches after implementing and operating them in real situations. Regarding possible improvements on the approaches, as the optimality gaps of the MIP model solutions are high, even after one hour of computer runtimes using CPLEX, a motivating line of research would be to explore model decomposition techniques to improve

Table A.10

Result variations between scenarios SA and Real for instance I67_3.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	55.57	0	0	0	3	0	0	0	0	-2	0	-492	534	3	0	-2	42
Test ¹ (f_2)	43.64	0	0	0	0	0	0	0	0	0	0	-24	-147	0	0	0	-171
Test ¹ (f_3)	42.12	0	0	0	0	0	0	0	0	1	0	-14	-354	0	0	1	-368
Test ¹ (f_4)	0.25	0	0	0	0	0	0	0	0	0	0	1087	-177	0	0	0	910
Test ² (F_1)	42.8	0	0	0	0	0	0	0	0	0	0	-24	-147	0	0	0	-171
Test ² (F_3)	41.94	0	0	0	0	0	0	0	0	0	0	-24	-149	0	0	0	-173
Test ² (F_4)	40.66	0	0	0	0	0	0	0	0	0	0	539	-757	0	0	0	-218
Test ³ ((pre-scheduled))	83.04	0	0	0	0	0	0	0	0	0	0	-220	640	0	0	0	420
Test ³ (Test ³ , {normal})	2.98	0	0	0	0	0	0	0	0	0	0	1087	-177	0	0	0	910
Test ³ (Test ³ , F_3)	-1.71	0	0	0	0	0	0	0	0	1	-1	1105	-216	0	0	0	889
Test ³ (Test ³ , F_2)	-1.6	0	0	0	0	0	0	0	0	1	-1	1105	-216	0	0	0	889
Test ³ (Test ³ , {0})	3.45	0	0	0	0	2	0	0	0	0	-1	1092	-201	2	0	-1	891
Test ³ (Test ³ , {0, 1})	0.79	0	0	0	0	2	0	0	0	0	-1	1092	-201	2	0	-1	891
Test ³ (Test ³ , {0, 1, 2})	1.62	0	0	1	2	3	0	0	0	0	-2	748	17	6	0	-2	765
Test ³ (Test ³ , F_1)	-5.46	0	0	1	2	3	0	0	0	0	-2	748	17	6	0	-2	765

Table A.11

Result variations between scenarios SA and Real for instance I71_3.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-29.31	0	0	0	2	0	0	0	0	0	-1	-44	-152	2	0	-1	-196
Test ¹ (f_2)	-20.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Test ¹ (f_3)	-22.97	0	0	0	0	0	0	0	0	1	0	2	-447	0	0	1	-445
Test ¹ (f_4)	-39.34	0	0	0	0	0	0	0	0	0	0	1062	-245	0	0	0	817
Test ² (F_1)	-20.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Test ² (F_3)	-20.57	0	0	0	0	0	0	0	0	1	-1	-12	-14	0	0	0	-26
Test ² (F_4)	-8.45	0	0	0	0	0	0	0	0	0	0	503	-392	0	0	0	111
Test ³ ((pre-scheduled))	-0.17	0	0	0	0	0	0	0	0	0	0	-237	854	0	0	0	617
Test ³ (Test ³ , {normal})	-39.39	0	0	0	0	0	0	0	0	0	0	1062	-245	0	0	0	817
Test ³ (Test ³ , F_3)	-38.16	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594
Test ³ (Test ³ , F_2)	-36.41	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594
Test ³ (Test ³ , {0})	-37.58	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594
Test ³ (Test ³ , {0, 1})	-37.51	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594
Test ³ (Test ³ , {0, 1, 2})	-38.02	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594
Test ³ (Test ³ , F_1)	-37.14	0	0	0	0	0	0	0	0	1	-1	1189	-595	0	0	0	594

Table A.12

Result variations between scenarios SA and Real for instance I82_3.

Test description	ΔT_{Time} (s)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
Test ¹ (f_1)	-54.57	0	0	2	6	1	0	0	0	-4	1	-1265	-390	9	0	-3	-1655
Test ¹ (f_2)	133.08	0	0	0	0	0	0	0	0	-1	1	257	303	0	0	0	560
Test ¹ (f_3)	108.03	0	0	0	0	0	0	0	0	0	0	55	152	0	0	0	207
Test ¹ (f_4)	-60.83	0	0	0	0	0	0	0	0	-1	1	2052	-812	0	0	0	1240
Test ² (F_1)	127.44	0	0	0	0	0	0	0	0	-1	1	257	303	0	0	0	560
Test ² (F_3)	97.85	0	0	0	0	0	0	0	0	0	0	55	152	0	0	0	207
Test ² (F_4)	25.79	0	0	0	0	0	0	0	0	-1	1	1093	-1110	0	0	0	-17
Test ³ ((pre-scheduled))	-2.34	0	0	0	0	0	0	0	0	-1	1	223	285	0	0	0	508
Test ³ (Test ³ , {normal})	-82.89	0	0	0	0	0	0	0	0	-1	1	2052	-812	0	0	0	1240
Test ³ (Test ³ , F_3)	-88.05	0	0	0	0	0	0	0	0	0	0	1710	-526	0	0	0	1184
Test ³ (Test ³ , F_2)	-84.98	0	0	0	0	0	0	0	0	0	0	1710	-526	0	0	0	1184
Test ³ (Test ³ , {0})	-81.3	0	0	0	0	0	0	0	0	0	0	1710	-526	0	0	0	1184
Test ³ (Test ³ , {0, 1})	-89.79	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474
Test ³ (Test ³ , {0, 1, 2})	-89.4	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474
Test ³ (Test ³ , F_1)	-87.54	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474

the model lower bounds. This could be useful for the development of more effective exact solution approaches for this problem. These decomposition-based exact approaches could also be used within the MIP-based local search heuristic in order to better explore the local search neighborhoods and thus, improve the heuristic performance. Other possible improvements on the solution approaches would be to combine the present two-phase heuristic with metaheuristics and mathheuristics in order to improve its computational performance.

As mentioned before, each of the weight/penalty values of the weighted additive objective function of the model were carefully defined by the company's programmers in order to reflect their relative priorities. These weights with different scales are depicted in lexicographical order and are actually used by the programmers to indicate the guidelines to identify better recovery plans for the company. The

use of normalization techniques for the term weights would be helpful and advisable for the standardization of the objective values and would not undermine the interpretation of the weights assigned in practical applications. However, as the used weight/penalty values defined by the programmers do not affect the correctness of the results, we prefer maintaining the analysis of the present application based on them, but we point out that the use of normalization techniques (e.g., nadir and utopia points) would be an interesting topic for future research in this and other practical ARP applications.

Another interesting topic for future research is to extend the present study to include decisions on assigning passengers to flights, merging flights, splitting flights, and flights that visit more than one maritime unit on the same route, among others. It is also in our research agenda to revisit this ARP considering uncertainties in some problem

parameters, for example, in the travel times of the flights, the service times in the maritime units and the times on ground at the aerodromes after landing and before taking-off, among others, by means of robust optimization and stochastic programming techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank an anonymous reviewer for the useful comments and suggestions of revision. The authors also thank the financial support of the Brazilian National Petroleum Agency (ANP), São Paulo Research Foundation (FAPESP), Brazil (grant numbers 16/01860-1 and 20/11602-5), the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES) and the Brazilian National Council for Scientific and Technological Development (CNPq) to carry out this research.

Appendix. Results obtained by scenario SA

As a complement to Section 5.4, Tables A.6–A.12 summarize the results for the remaining problem instances of the numerical experiments. These experiments were conducted to study the impact on computational performance and solution quality of heuristic H1 with regard to the changes in the weights of the objective function (Scenario SA). The analysis of the information of these tables is similar to the analysis presented for Table 5.

References

- Arguello, M.F., 1997. Framework for Exact Solutions and Heuristics for Approximate Solutions to Airlines' Irregular Operations Control Aircraft Routing Problem. (Ph.D. thesis). Department of Mechanical Engineering, University of Texas, Austin.
- Arkan, U., Gürel, S., Aktürk, M.S., 2016. Integrated aircraft and passenger recovery with cruise time controllability. *Ann. Oper. Res.* 236 (2), 295–317. <http://dx.doi.org/10.1007/s10479-013-1424-2>.
- Bard, J.F., Yu, G., Arguello, M.F., 2001. Optimizing aircraft routings in response to groundings and delays. *IEE Trans.* 33 (10), 931–947. <http://dx.doi.org/10.1080/07408170108936885>.
- Brunner, J.O., 2014. Rescheduling of flights during ground delay programs with consideration of passenger and crew connections. *Transp. Res. Part E: Logist. Transp. Rev.* 72, 236–252. <http://dx.doi.org/10.1016/j.tre.2014.10.004>.
- Cao, J., Kanafani, A., 1997. Real-time decision support for integration of airline flight cancellations and delays Part I: mathematical formulation. *Transp. Plan. Technol.* 20 (3), 183–199. <http://dx.doi.org/10.1080/03081069708717588>.
- De La Vega, J., Santana, M., Pureza, V., Morabito, R., Bastos, Y., Ribas, P.C., 2022. Model-based solution approach for a short-term flight rescheduling problem in aerial passenger transportation to maritime units. *International Transactions in Operational Research* 29 (6), 3400–3434. <http://dx.doi.org/10.1111/itor.13079>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.13079>.
- Eggenberg, N., Salani, M., Bierlaire, M., 2010. Constraint-specific recovery network for solving airline recovery problems. *Comput. Oper. Res.* 37 (6), 1014–1026. <http://dx.doi.org/10.1016/j.cor.2009.08.006>.
- Hu, Y., Song, Y., Zhao, K., Xu, B., 2016. Integrated recovery of aircraft and passengers after airline operation disruption based on a GRASP algorithm. *Transp. Res. Part E: Logist. Transp. Rev.* 87, 97–112. <http://dx.doi.org/10.1016/j.tre.2016.01.002>.
- Jarrah, A.I.Z., Yu, G., Krishnamurthy, N., Rakshit, A., 1993. A decision support framework for airline flight cancellations and delays. *Transp. Sci.* 27 (3), 266–280. <http://dx.doi.org/10.1287/trsc.27.3.266>.
- Lin, H., Wang, Z., 2018. Fast variable neighborhood search for flight rescheduling after airport closure. *IEEE Access* 6, 50901–50909. <http://dx.doi.org/10.1109/ACCESS.2018.2869842>.
- Mitrović-Minić, S., Punnen, A.P., 2010. Variable intensity local search. In: Maniezzo, V., Stützle, T., Voß, S. (Eds.), *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer US, Boston, MA, pp. 245–252.
- Sinclair, K., Cordeau, J.F., Laporte, G., 2014. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European J. Oper. Res.* 233 (1), 234–245. <http://dx.doi.org/10.1016/j.ejor.2013.08.034>.
- Teodorović, D., Guberinić, S., 1984. Optimal dispatching strategy on an airline network after a schedule perturbation. *European J. Oper. Res.* 15 (2), 178–182. [http://dx.doi.org/10.1016/0377-2217\(84\)90207-8](http://dx.doi.org/10.1016/0377-2217(84)90207-8).
- Teodorović, D., Stojković, G., 1990. Model for operational daily airline scheduling. *Transp. Plan. Technol.* 14 (4), 273–285. <http://dx.doi.org/10.1080/03081069008717431>.
- Teodorović, D., Stojković, G., 1995. Model to reduce airline schedule disturbances. *J. Transp. Eng.* 121 (4), 324–331. [http://dx.doi.org/10.1061/\(ASCE\)0733-947X\(1995\)121:4\(324\)](http://dx.doi.org/10.1061/(ASCE)0733-947X(1995)121:4(324)).
- Vieira, T., De La Vega, J., Tavares, R., Munari, P., Morabito, R., Bastos, Y., Ribas, P.C., 2021. Exact and heuristic approaches to reschedule helicopter flights for personnel transportation in the oil industry. *Transp. Res. Part E: Logist. Transp. Rev.* 151, 102322. <http://dx.doi.org/10.1016/j.tre.2021.102322>, URL <https://www.sciencedirect.com/science/article/pii/S136655452100096X>.
- Yan, S., Tu, Y., 1997. Multifleet routing and multistop flight scheduling for schedule perturbation. *European J. Oper. Res.* 103 (1), 155–169. [http://dx.doi.org/10.1016/S0377-2217\(96\)00260-3](http://dx.doi.org/10.1016/S0377-2217(96)00260-3).
- Yan, S., Yang, D.-H., 1996. A decision support framework for handling schedule perturbation. *Transp. Res. B* 30 (6), 405–419. [http://dx.doi.org/10.1016/0191-2615\(96\)00013-6](http://dx.doi.org/10.1016/0191-2615(96)00013-6).
- Zhang, D., Yu, C., Desai, J., Lau, H.H., 2016. A math-heuristic algorithm for the integrated air service recovery. *Transp. Res. B* 84, 211–236. <http://dx.doi.org/10.1016/j.trb.2015.11.016>.