Exact and heuristic approaches to reschedule helicopter flights for personnel transportation in the oil industry

Thiago Vieira[a], Jonathan De La Vega[a], Roberto Tavares[a], Pedro Munari[a], Reinaldo Morabito[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*

Abstract

This paper addresses a real-life short-term rescheduling problem of helicopter flights from one onshore airport to several maritime units in the context of the oil industry. This is a complex and challenging problem to solve because of the particular characteristics observed in practice, such as pending flights transferred from previous days with different recovering priorities, changes in flight timetables and helicopter assignments previously planned for the current day, time windows and minimum time intervals between take-offs from the airport and the maritime units, mandatory flight precedence, maximum flight delays, among many others. The problem consists of determining a daily flight reschedule that satisfies operational constraints and recovers all pending flights, while minimizing flight delays and costs related to helicopter usage and reassignments. We propose two mixed integer programming models to formulate the problem with all relevant characteristics, one based on the extension of traditional network flow models and other that relies on a novel event-based representation of the problem. Additionally, we develop an effective heuristic approach based on constructive and improvement heuristics, able to produce high-quality solutions within acceptable computational times. The results of computational experiments with real-life data provided by an oil company highlight the potential of the proposed approaches to support decision making in this context.

Keywords: Aircraft recovery problem, mixed integer programming, flight rescheduling problem, constructive and improvement heuristics, helicopters and flights re-assignments, oil industry.
2010 MSC: 00-01, 99-00

1. Introduction

We address a rescheduling problem of helicopter flights that transport personnel from a coastal airport to different maritime units spread over the sea (e.g., offshore oil rigs, gas-producing platforms, etc.), motivated by the real case of a Brazilian oil company. These flights are mainly related to transporting crews that are starting or finishing their duty at each offshore platform. At the beginning of the week, the company determines a flight schedule for each day in that week, defining departure times for the flights required on that day. Some maritime units have more than one flight per day and each flight is often a round trip between the company airport and a destination maritime unit (i.e., airport - maritime unit - airport). Each

*Corresponding author
Email addresses: `thiagozevieira@gmail.com` (Thiago Vieira), `jdlvmartinez@gmail.com` (Jonathan De La Vega), `tavares@dep.ufscar.br` (Roberto Tavares), `munari@dep.ufscar.br` (Pedro Munari), `morabito@ufscar.br` (Reinaldo Morabito), `yan.bastos@petrobras.com.br` (Yan Bastos), `paulo.ribas@petrobras.com.br` (Paulo César Ribas)

flight departs from the airport with a group of previously booked personnel with duties at the maritime unit of destination. Once these personnel arrive at their destination, the helicopter comes back from this unit with another group of also previously booked personnel that finished their duty.

Each helicopter hired by the company may carry out several round-trip flights per day. In the absence of unexpected events, these daily flights should follow the schedule as planned by the company at the beginning of the week, each one with its given departure time, maritime unit destination, booked personnel and a given helicopter. However, the daily flight schedule often cannot be carried out as previously planned because of unexpected events, such as bad weather conditions at the airport and/or the maritime units, passenger delays due to boarding/unboarding, variations in the travel times and ground times, helicopter failures, unavailable fuel supply, problems with the air traffic control, among others. As a consequence, the departure times of some flights may be delayed to other times of the day, if possible, or even postponed to the next day if a reschedule for the same day is not possible, in which case it is called a *transferred* flight. Such flights, together with other relevant random events on the day, require an effective rescheduling of the flights originally planned, which should be done before the beginning of the next day. The company operators define different penalties that incur when e.g. there is a delay in the original departure times of the flights scheduled for the current day; flights are rescheduled for the next day; the original assignment of helicopters to flight changes; additional helicopters are needed to cover the flights; and transferred flights from previous days are not assigned on the current day.

This rescheduling problem is complex and challenging to solve because of the many different characteristics that should be taken into account in the decision making process, the size of the problem in practical settings and the various specific constraints regarding the airport, maritime units, flights and helicopters, as discussed below. For instance, the aerial passenger transportation of the oil company considered here is the fourth largest in number of flights in Brazil (Hermeto et al., 2019) and a typical problem size can have dozens of maritime units, dozens of different daily flights to them and several available helicopters. This transport operation is essential for oil companies with offshore maritime units; similar situations arise in other oil companies operating in the North sea, Gulf of Mexico, West of Africa and Australia, for example. For simplicity, in this study, all problem parameters are assumed to be deterministic (i.e., known in advance). Such rescheduling problem can be seen as the problem of assigning round-trip flights to helicopters, sequencing these flights in a daily journey of the helicopters and, at the same time, sequencing all flights in the busy runway of the airport and in the single heliport of each maritime unit.

In the literature, the problem of determining new departure times for flights after disruptions in the schedule is called the aircraft recovery problem (ARP) (Belobaba et al., 2015). Teodorovic and Guberinic (1984) were the first to address the ARP, considering a real case with 8 flights and 3 aircraft. Several works on the ARP have emerged since then, encompassing different particularities. In terms of optimization objectives, the problem generally consists of minimizing the delays of rescheduled flights and number of aircraft swaps and flight cancellations (Belobaba et al., 2015), but there are also studies considering other objectives, such as minimizing fuel costs by adjusting the cruising speed, minimizing the maximum delay of the rescheduled flights and minimizing the number of routes that cover the rescheduled flights (Jozefowiez et al., 2013; Akturk et al., 2014; Arıkan et al., 2016; Hu et al., 2017; Zhang, 2017).

Given the practical appeal of the ARP, different studies have considered a set of specific features when addressing the problem. Nevertheless, a few characteristics are common in most studies, such as fleet heterogeneity and the presence of time windows. Regarding the fleet type, there are studies considering homogeneous fleet (Bard et al., 2001; Rosenberger et al., 2003; Eggenberg et al., 2010; Gao et al., 2012; Brunner, 2014; Kammoun et al., 2014; Zhu et al., 2015; Kammoun et al., 2016; Zhang et al., 2016; Kammoun and Rezg, 2018; Liang et al., 2018; Ali and Nidhal, 2019; Erkan et al., 2019), as well as heterogeneous fleet (Thengvall et al., 2000, 2001; Filar et al., 2007; Jafari and Zegordi, 2011; D'Ariano et al., 2012; Jozefowiez et al., 2013; Sinclair et al., 2014; Zhang et al., 2015; Arıkan et al., 2016; Hu et al., 2015, 2016; Sinclair et al., 2016; Hu et al., 2017). For time windows, a few studies impose them only to airports (Bard et al., 2001; Rosenberger et al., 2003; Eggenberg et al., 2010; Jafari and Zegordi, 2011; Gao et al., 2012; Kammoun et al., 2014; Zhang et al., 2015; Hu et al., 2015, 2016; Kammoun et al., 2016; Hu et al., 2017; Kammoun and Rezg, 2018; Erkan et al., 2019), others only to aircraft (Zhang et al., 2016; Liang et al., 2018), or even to both airports and aircraft (Jozefowiez et al., 2013; Sinclair et al., 2014, 2016).

The literature on the ARP also includes studies addressing uncertain aircraft recovery times related to repair times in case of a breakdown (Zhu et al., 2015); maintenance periods that are previously determined and must be included in the new schedule (Bard et al., 2001; Zhang et al., 2016); aircraft time windows that model the maximum flight time allowed until required preventive maintenance (Jozefowiez et al., 2013; Sinclair et al., 2014, 2016; Liang et al., 2018); and aircraft balance, i.e., keeping a given number of aircraft at each airport by the end of the new schedule (Thengvall et al., 2001; Bard et al., 2001; Hu et al., 2016, 2017). There are also papers on the helicopter transportation of oil rig crew members in the context of oil companies (Galvão and Guimarães, 1990; Moreno et al., 2006; Menezes et al., 2010; Qian et al., 2011, 2012, 2015; Gribkovskaia et al., 2015), but most of them focus on helicopter routing and passenger allocation decisions or minimizing operation safety risks, as pointed out in Bastos et al. (2020), which is not the case of the present study. Additionally, we assume that each flight is planned for only one maritime unit and there are no splitting or merging of flights.

Some features of the ARP considered in the present study are uncommon (but not new) in the literature, such as flights with different rescheduling priorities (Gao et al., 2012) and the possibility of using spare aircraft to recover flights (Jafari and Zegordi, 2011). However, we did not find models incorporating both simultaneously. Moreover, other constraints of the addressed ARP were not identified in other studies, such as mandatory flight precedence (i.e., certain flights must be rescheduled before others, even if this worsens the objective function value of the reschedule); there is a single runway for taking-off and landing at the airport; there is a single heliport at each maritime unit that allows only one helicopter on the ground at a time; the so-called *entourage flight*, which are flights that after landing at the maritime unit, occupy the heliport of the maritime unit until the end of the day; strict operational rules related to the daily flight timetables; and limitations on the number and type of the available helicopters for some offshore flights. Furthermore, this ARP does not operate with connection flights, as passengers cannot wait at the maritime units due to safety reasons (each take-off and landing operation is considered as a high-risk activity); passengers are not customers, but actually company's staff working on stressful activities and hence even short flight delays can have a negative impact to the psychological and physical health (which affects their productivity), and also can increase the company's expenses with extra daily allowances and overtime. When combined, such characteristics make the problem unique in the literature. Therefore, we define this problem as an aircraft recovery problem with priority distinction between flights, fleet and delays (ARP-PD).

To the best of our knowledge, this is the first study focusing on a real-life short-term rescheduling problem of helicopter flights transporting personnel to and from maritime units in the context of an oil company and under the particular constraints mentioned above. The main contributions of this paper are threefold: (*i*) we describe in details the characteristics of the addressed real-life problem, so that researchers may further use this detailed description in their studies; (*ii*) we propose two mixed integer programming (MIP) formulations, based on different representations of the ARP-PD, that fully represents its relevant characteristics; (*iii*) we develop customized heuristic approaches to find relatively good feasible solutions within acceptable computational times. In addition to the theoretical contributions related to the proposed models and algorithms, this study contributes to the practice of operations research, enabling the improvement of the company's decision making regarding flight rescheduling, by highlighting the potential of the proposed approach when comparing its solutions with the company's solutions.

The remainder of this paper is organized as follows. In Section 2, we describe in details the addressed flight rescheduling problem. Then, we propose two alternative MIP formulations to model this problem in Section 3, and develop the tailor-made heuristic approaches in Section 4. Computational results using real-life data of the Brazilian company are presented and analyzed in Section 5. These results show that the heuristics are effective for solving realistic problem instances in practical settings. Finally, the concluding remarks of this study and some perspectives for future research are discussed in Section 6.

2. Problem description

As mentioned before, this study is inspired by a real-life problem of rescheduling daily helicopter flights that transport personnel from the company's airport to its maritime units (e.g., rigs, floating platforms, fixed platforms, floating hotels, maintenance units, support vessels, etc.). The company programmers previously schedule several daily round-trip flights at the beginning of the week and, in the absence of unexpected events (see the previous section for examples of such events), the personnel transportation should follow these previously scheduled flights.

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 and even the rescheduling of a flight to the next day (transferred flights). These changes require a revised schedule for the previously planned flights, which are treated as an ARP-PD by the company programmers. In the case of transferred flights, the flight schedule of the day $d + 1$ should be revised to cope with transferred flights from day d, plus the flights previously scheduled for day $d + 1$. When rescheduling a flight for the next day, we should take into account the following characteristics and requirements regarding the airport, maritime units, flights and helicopters:

i) Airport: The airport has a single and busy runway, from which only one helicopter can take off at a time. Thus, the interdeparture time between two consecutive flights in this runway should not be less than a given time interval, typically of 5 minutes. There is a daily time window for the operation of the airport that depends on the sunrise and sunset times of the day, and all flights should take off and land at the airport in sunlight (i.e., within this time window). After landing at the airport, the helicopter must not take off before undergoing an inspection procedure, which corresponds to the total preparation time of the helicopter to perform the next flight, referred to as the time on the ground. The minimum time interval between the landing and take-off for consecutive flights using the same helicopter should not be less than a given time interval, typically of 45 minutes, including the times for passenger unboarding and boarding the helicopter.

ii) Maritime units: The location of a maritime unit of a flight is fixed and known in advance. In practice, they may be mobile units and their locations can change over time, but the programmers can determine exactly their location during the planning process. Each maritime unit has a single heliport for landing and taking off and hence only one helicopter should be on the ground of this unit per time. Because of the boarding and unboarding operations in the unit, the time interval between landing and take-off of a helicopter should not be less than a given time interval, typically of 15 minutes, during which other helicopters must not land at the unit. This also influences the departure times at the airport and hence the flight rescheduling operation should ensure that the time interval between the take-offs of two consecutive flights from the airport runway going to the same maritime unit should avoid more than one helicopter on the ground of this unit at the same time.

iii) Flights: Each flight is defined by its maritime unit destination and its departure time from the airport on the current day. We represent it as a simple sequence (route): airport - maritime unit - airport. Moreover, each flight has a previously assigned group of passengers going to the maritime unit and another previously assigned group of passengers coming back from this unit using the same helicopter, which are typically related to the work shifts at the unit. The flight can be originally scheduled in the timetable of the current day (table flight), or a flight transferred from previous days that needs to be rescheduled on the current day. There may be more than one flight to each maritime unit on a given day. The travel times of a flight are assumed deterministic and depend only on the destination (maritime unit) and the helicopter (assigned to the flight). A table flight should not departure from the airport before its original departure time. Hence, the rescheduling of table flights can only postpone their departure times, but never anticipate them. Moreover, a table flight may be delayed for up to four hours; otherwise, it has to be transferred to the next day and be rescheduled together with the table flights of the next day. If there is a transferred flight from the last day and a table flight of the current day going to the same maritime unit, the transferred flight has to land in that unit before the table flight, even if this implies in a delayed departure time for the table flight. In this case, the table flight has to be rescheduled because of this precedence constraint between flights going to the same maritime unit. There is also a subset of priority flights called *entourage flights* which after landing at a maritime unit heliport, block it for the whole time spent by the entourage in the unit, commonly for the rest of the day. These special flights are used to transport managers and other representatives of the company for special visits at the maritime unit.

iv) Helicopters: The helicopter fleet is hired by the company for a relatively long term, e.g. through affreightment contracts of one or more years. The fleet is heterogeneous in terms of travel cruise speed, the capacity of passengers, flying range, etc., because the helicopters are of different models. Moreover, some helicopters are not able to fly to some maritime units because of their distances to the airport, or because of the sizes of their heliports. The same helicopter can perform several flights on a day as long as these flights have a number of booked passengers that does not exceed the capacity of the helicopter. The number of flights (turns) a helicopter can do in a day is implicitly limited by the time window of the airport, duration of flights, constraints regarding the interdeparture and interarrival times, among others. There are three types of helicopter in the fleet: *normal* aircraft, which are helicopters originally assigned to the daily table flights; *pool* aircraft, which are spare helicopters promptly available at the airport but not previously assigned to any flight – they can be used for flight rescheduling at additional costs; and *spot* aircraft, which are not promptly available at the airport but can be used for flight rescheduling with much higher additional costs than *pool* aircraft.

To facilitate the problem description, consider the time-space diagram presented in Fig. 1 for a simple schedule of five table flights (i_1 to i_5) of a given time interval of a day. For each flight, this diagram indicates the corresponding scheduled departure times (r_{i_1} to r_{i_5}), the travel times from the airport (represented along the horizontal axis of the figure) to the maritime units (MU_1 to MU_5), visiting times in the units and travel times back to the airport. Three helicopters are used to perform these flights: helicopter 1 for flights i_1, i_3, i_5, helicopter 2 for flight i_2 and helicopter 3 for flight i_4. In the absence of unexpected events, table flights i_1 to i_5 would strictly follow the schedule illustrated in Fig. 1. On the other hand, if there were events causing the partial interruption of the operations of the airport and/or maritime units, some table flights of previous days may be transferred to the current day considered in the figure. Fig. 2 illustrates this situation, in which a transferred flight (i_0) from the previous day is rescheduled (or "recovered", as named by the company operators) to the current day (originally planned as in Fig. 1). After the rescheduling, helicopter 1 is assigned to flights i_0, i_3, i_5, helicopter 2 to flight i_2 and helicopter 3 to flights i_1, i_4. Note in Fig. 2 that the revised schedule implies in a short delay ($D_{i_1} > 0$) in the departure time of flight i_1 and a change of its assigned helicopter. Helicopter 3 becomes assigned to flight i_1, instead of helicopter 1 as before, because the former is no longer able to perform flight i_1 followed by flight i_3 without delaying the departing time of flight i_3 due to the requirement of a minimum time on the ground between two flights of the same helicopter. Depending on the number of transferred flights from previous days, the rescheduling could imply in further delays of other flights and reassignments of other helicopters, or even having to transfer some of these flights to the next day.

Therefore, in addition to involving the rescheduling of the table flights of the day, the flight rescheduling problem addressed in this paper also involves the rescheduling (recovering) of transferred flights from previous days. It should satisfy different practical requirements, while minimizing a weighted sum of different penalties defined by the company operators and associated to: *i*) transferred and table flights that cannot be rescheduled on the current day, *ii*) the use of additional and more expensive spare helicopters (e.g., from the pool and spot fleets) to cover transferred and table flights, *iii*) delays in the departure times of the table flights, and *iv*) changes in the previous assignment of (normal) helicopters to cover the table flights. In case *i*), the company operators define different penalties for not scheduling table flights and entourage flights of the day, transferred flights from the previous day (i.e., one day before the current day), and transferred flights from two or more days before. In case *ii*), they define different penalties for using helicopters from the pool and spot sets. In case *iii*), they define different penalties for short flight delays of less than 15 minutes (called type I delays), and for longer delays from 15 minutes to 4 hours (called type II delays).

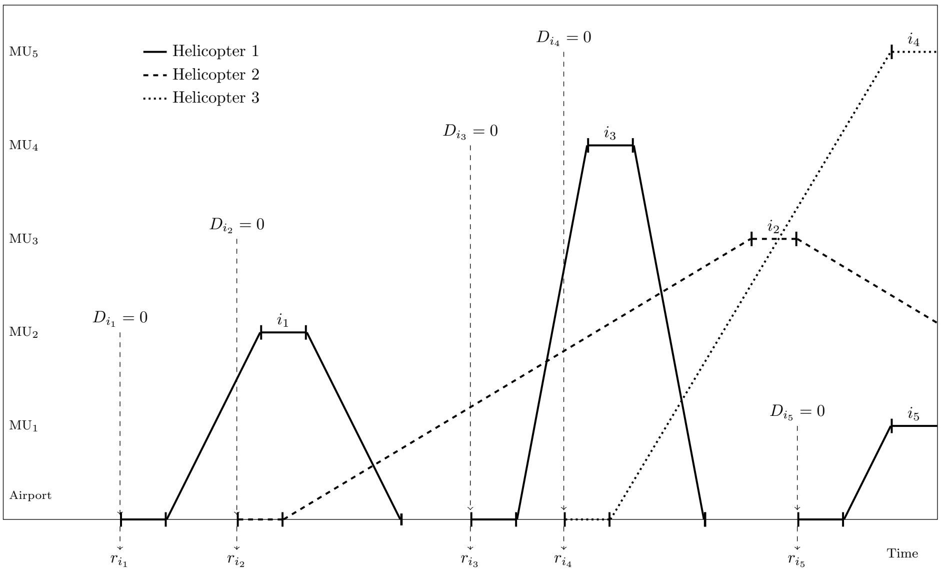


Figure 1: Time-space diagram illustrating a schedule with five flights, five maritime units and three aircraft

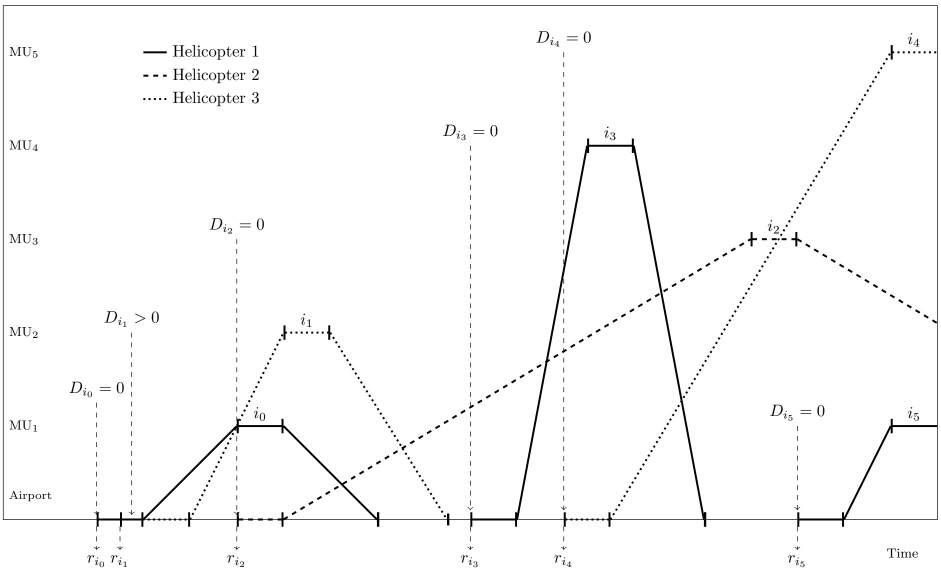


Figure 2: Time-space diagram illustrating the reschedule with five flights, five maritime units and three aircraft

In the next section, we present two MIP models that fully formulates the described problem, while in Section 4 we develop a tailor-made heuristic able to find relatively good feasible solutions within acceptable computational times.

3. MIP formulations

This section introduces the notation used throughout the paper and presents two MIP formulations of the ARP-PD, one based on the extension of a traditional network flow model, and the other based on a novel event-based representation. Based on the ARP-PD description presented in Section 2, we define the following notation to denote the sets and parameters that are common in both models:

Sets and indices:

$\mathcal{I}$ Set of flights, indexed by i and j.

$\mathcal{H}$ Set of helicopters, indexed by h.

$\mathcal{P}$ Set of maritime units, indexed by p.

$\mathcal{I}_0$ Subset of table flights.

$\mathcal{I}_1$ Subset of transferred flights from the previous day.

$\mathcal{I}_2$ Subset of transferred flights from two or more days before.

$\mathcal{I}_C$ Subset of entourage flights.

$\mathcal{I}_h$ Subset of flights that can be assigned to helicopter h.

$\mathcal{I}_p$ Subset of flights to maritime unit $p \in \mathcal{P}$.

$\mathcal{H}_n$ Subset of helicopters in the normal set.

$\mathcal{H}_p$ Subset of helicopters in the pool set.

$\mathcal{H}_s$ Subset of helicopters in the spot set.

$\mathcal{H}_i$ Subset of helicopters able to perform flight $i \in \mathcal{I}$.

Parameters:

r_i Original scheduled departure time of flight i.

tf_i Duration of flight i, including the time spent in the maritime unit of the flight.

s_i Helicopter originally assigned to flight $i \in \mathcal{I}_0 \cup \mathcal{I}_C$.

tat Minimum time on the ground between two consecutive flights of the same helicopter.

t^{a} Minimum time between two consecutive take-offs in the runway of the airport (*briefing* of safety).

t_i^{u} Time spent in the maritime unit for flight i.

$d_{\max}^{I}$ Maximum delay (of type I) allowed in the departure time of a table flight (15 minutes).

$d_{\max}^{II}$ Maximum delay (of type II) allowed in the departure time of a table flight (240 minutes).

$[tw^a, tw^b]$ Daily time window of the airport (typically between 7:00am and 6:00pm).

3.1. Network-based formulation

The network-based formulation proposed in this section is based on the extension of the traditional three-index vehicle-flow formulation that has been widely used in the literature to formulate vehicle routing problems with heterogeneous fleet. The extension consists in creating new variables and constraints to guarantee that all relevant characteristics of the real-life problem are properly incorporated in the model. In this way, consider the following decision variables:

$X_{ijh} \in \{0,1\}$ Assumes the value of 1 if and only if helicopter h performs flight j immediately after flight i.

$Y_{ih} \in \{0,1\}$ Assumes the value 1 if and only if flight i is performed by helicopter h.

$Z_{ij} \in \{0,1\}$ Assumes the value 1 if and only if the departure of flight i from the airport is before the departure of flight j from the airport.

$V_h \in \{0,1\}$ Assumes the value 1 if and only if helicopter h is used.

$B_i^I \in \{0,1\}$ Assumes the value 1 if and only if the delay of flight i is less than or equal to $d_{\max}^I$, for $i \in \mathcal{I}_0 \cup \mathcal{I}_C$.

$B_i^{II} \in \{0,1\}$ Assumes the value 1 if and only if the delay of flight i is greater than $d_{\max}^I$ but less than or equal to $d_{\max}^{II}$, for $i \in \mathcal{I}_0 \cup \mathcal{I}_C$.

$D_i \geq 0$ Delay of flight i with respect to its original scheduled departure time r_i.

$DT_i \geq 0$ Departure instant of flight i.

$AT_i \geq 0$ Arrival instant of flight i.

The objective function (3.1) consists of minimizing the total weighted sum of the following terms: f_1, number of entourage flights not scheduled on the current day; f_2, number of transferred flights from two or more days before that are not scheduled on the current day; f_3, number of transferred flights from the previous day that are not schedule on the current day; f_4, number of table flights not scheduled on the current day; f_5, number of additional helicopters used from the spot set; f_6, number of additional helicopters used from the pool set; f_7, number of normal helicopters; f_8, number of delayed table and entourage flights for which the delay is greater than 15 minutes but less than or equal to 4 hours (type II delay); f_9, number of delayed table or entourage flights for which the delay is less than or equal to 15 minutes (type I delay); f_{10}, number of helicopters assigned to a table or an entourage flight that is different from the one originally assigned; and f_{11}, total delay considering all flights. Penalties w_1 to w_{11} are the weights corresponding to terms f_1 to f_{11}, respectively. This weighted objective function is based on the company's policy and experience to deal with flight recovery, helicopter allocation and flight delays in practice. The weights are carefully defined by the company's manager in order to impose the lexicographic order: $w_1 > w_2 > \ldots > w_{11}$, taking into account the relative importance of each type of flight and each type of helicopter used and the consequences of the respective flight delays.

$$\min f = \sum_{i=1}^{11} w_i f_i, \tag{3.1}$$

where each component f_i is defined as

$$f_1 := \left(|\mathcal{I}_C| - \sum_{i \in \mathcal{I}_C} \sum_{h \in \mathcal{H}_i} Y_{ih} \right); \quad f_2 := \left(|\mathcal{I}_2| - \sum_{i \in \mathcal{I}_2} \sum_{h \in \mathcal{H}_i} Y_{ih} \right); \quad f_3 := \left(|\mathcal{I}_1| - \sum_{i \in \mathcal{I}_1} \sum_{h \in \mathcal{H}_i} Y_{ih} \right);$$
$$f_4 := \left(|\mathcal{I}_0| - \sum_{i \in \mathcal{I}_0} \sum_{h \in \mathcal{H}_i} Y_{ih} \right); \quad f_5 := \sum_{h \in \mathcal{H}_s} V_h; \quad f_6 := \sum_{h \in \mathcal{H}_p} V_h; \quad f_7 := \sum_{h \in \mathcal{H}_n} V_h;$$
$$f_8 := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} B_i^{II}; \quad f_9 := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} B_i^I; \quad f_{10} := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} \sum_{\substack{h \in \mathcal{H}_i: \\ h \neq s_i}} Y_i^h; \quad f_{11} := \sum_{i \in \mathcal{I}} D_i.$$

Having stated the objective function, we define the flow constraints (3.2)-(3.4) that guarantee the correct flow of aircraft through the network and related the flow variables X_{ijh} with the aircraft assignment and usage variables Y_{ih} and V_h.

$$\sum_{j \in \mathcal{I}_h \cup \{n+1\}} X_{ijh} = Y_{ih}, \; \forall i \in \mathcal{I}, \; \forall h \in \mathcal{H}_i. \tag{3.2}$$

$$\sum_{j \in \{0\} \cup \mathcal{I}_h} X_{jih} = \sum_{j \in \mathcal{I}_h \cup \{n+1\}} X_{ijh}, \; \forall i \in \mathcal{I}, \; \forall h \in \mathcal{H}_i. \tag{3.3}$$

$$\sum_{j\in\mathcal{I}}\sum_{h\in\mathcal{H}_j} X_{0jh} = \sum_{i\in\mathcal{I}}\sum_{h\in\mathcal{H}_i} X_{i(n+1)h} = \sum_{h\in\mathcal{H}} V_h. \tag{3.4}$$

Constraints (3.5)-(3.6) state that each flight can be assigned to at most one aircraft and that such assignment can only happen if this aircraft is used.

$$\sum_{h\in\mathcal{H}_i} Y_{ih} \leq 1,\ \forall i \in \mathcal{I}. \tag{3.5}$$

$$Y_{ih} \leq V_h,\ \forall h \in \mathcal{H}, \forall i \in \mathcal{I}_h. \tag{3.6}$$

Constraints (3.7)-(3.12) are time constraints, which impose the synchronization of the departures of the flights made by the same helicopter, the satisfaction of the planned departure time of the flights, and the closing time of the airport time window. They also determine the total delay for each flight.

$$DT_j \geq AT_i - M\left(1 - \sum_{h\in\mathcal{H}_i\cap\mathcal{H}_j} X_{ijh}\right),\ \forall i \in \{0\},\ j \in \mathcal{I}. \tag{3.7}$$

$$DT_j \geq AT_i - M\left(1 - \sum_{h\in\mathcal{H}_i\cap\mathcal{H}_j} X_{ijh}\right),\ \forall i \in \mathcal{I},\ j \in \{n+1\}. \tag{3.8}$$

$$DT_j \geq AT_i + tat\sum_{h\in\mathcal{H}_i\cap\mathcal{H}_j} X_{ijh} - M\left(1 - \sum_{h\in\mathcal{H}_i\cap\mathcal{H}_j} X_{ijh}\right),\ \forall i,j \in \mathcal{I},\ i \neq j. \tag{3.9}$$

$$AT_i \geq DT_i + tf_i\sum_{h\in\mathcal{H}_i} Y_{ih},\ \forall i \in \mathcal{I}. \tag{3.10}$$

$$r_i\sum_{h\in\mathcal{H}_i} Y_{ih} \leq DT_i \leq r_i\sum_{h\in\mathcal{H}_i} Y_{ih} + D_i,\ \forall i \in \mathcal{I}. \tag{3.11}$$

$$AT_i \leq tw^b\sum_{h\in\mathcal{H}_i} Y_{ih},\ \forall i \in \mathcal{I}. \tag{3.12}$$

Constraints (3.13) and (3.14) activate the binary variables related to the type of delays (I or II) incurring to table and entourage flights. Specifically, constraints (3.13) ensure that a flight can have at most one type of delay, while constraints (3.14) limit the delay duration according to the delay type. Note that, together with the weights assigned to B_i^I and B_i^{II} in the objective function (3.1), these constraints model the piecewise linear behavior of the penalties applied in terms of the delay duration.

$$B_i^I + B_i^{II} \leq 1,\ \forall i \in \mathcal{I}_C \cup \mathcal{I}_0. \tag{3.13}$$

$$D_i \leq d_{\max}^I B_i^I + d_{\max}^{II} B_i^{II},\ \forall i \in \mathcal{I}_C \cup \mathcal{I}_0. \tag{3.14}$$

If flights i and j are scheduled in a solution, constraints (3.15)-(3.17) impose a single order of precedence between them.

$$Z_{ij} + Z_{ji} \leq 1,\ \forall i,j \in \mathcal{I},\ i \neq j. \tag{3.15}$$

$$Z_{ij} + Z_{ji} \geq \sum_{h\in\mathcal{H}_j} Y_{jh} + \sum_{h\in\mathcal{H}_i} Y_{ih} - 1,\ \forall i,j \in \mathcal{I},\ i \neq j. \tag{3.16}$$

$$2\left(Z_{ij} + Z_{ji}\right) \leq \sum_{h\in\mathcal{H}_i} Y_{ih} + \sum_{h\in\mathcal{H}_j} Y_{jh}, \forall i,j \in \mathcal{I}, i \neq j. \tag{3.17}$$

Constraints (3.18) and (3.19) enforce the synchronization of the departure of flights from the airport and their arrival at the maritime units, respectively.

$$DT_j - DT_i \geq t^{\mathrm{a}} Z_{ij} - M(1 - Z_{ij}),\ \forall i,j \in \mathcal{I},\ i \neq j. \tag{3.18}$$

$$DT_j - DT_i \geq t_i^{\mathrm{u}} Z_{ij} - M(1 - Z_{ij}),\ \forall p \in \mathcal{P},\ \forall i,j \in \mathcal{I}_p,\ i \neq j. \tag{3.19}$$

Constraints (3.20) guarantee the precedence between transferred and table flights going to the same maritime unit. Note that the departure of a table flight from the airport can never be before the departure of a transferred flight if both flights have the same maritime unit of destination. Constraints (3.21) block the maritime unit of destination of the entourage flights, while constraints (3.22) block the helicopter that performs them.

$$Z_{ij} = 0,\ \forall i \in \mathcal{I}_0 \cap \mathcal{I}_p,\ \forall j \in (\mathcal{I}_1 \cup \mathcal{I}_2) \cap \mathcal{I}_p,\ \forall p \in \mathcal{P}. \tag{3.20}$$
$$Z_{ij} = 0,\ \forall i \in \mathcal{I}_C \cup \mathcal{I}_p,\ \forall j \in \mathcal{I} \setminus \mathcal{I}_C \cup \mathcal{I}_p,\ \forall p \in \mathcal{P}. \tag{3.21}$$
$$X_{i(n+1)h} = Y_{ih},\ \forall i \in \mathcal{I}_C,\ \forall h \in \mathcal{H}_i. \tag{3.22}$$

Finally, constraints (3.23)-(3.29) define the type and domain of the decision variables.

$$X_{ijh} \in \{0,1\},\ \forall i \in \{0\} \cup \mathcal{I};\ j \in \mathcal{I} \cup \{n+1\};\ i \neq j; h \in \mathcal{H}. \tag{3.23}$$
$$Y_{ih} \in \{0,1\},\ \forall i \in \mathcal{I};\ h \in \mathcal{H}. \tag{3.24}$$
$$Z_{ij} \in \{0,1\},\ \forall i,j \in \mathcal{I}. \tag{3.25}$$
$$V_h \in \{0,1\},\ \forall h \in \mathcal{H}. \tag{3.26}$$
$$B_i^I \in \{0,1\},\ \forall i \in \mathcal{I}. \tag{3.27}$$
$$B_i^{II} \in \{0,1\},\ \forall i \in \mathcal{I}. \tag{3.28}$$
$$DT_i \geq 0, AT_i \geq 0, D_i \geq 0,\ \forall i \in \mathcal{I}. \tag{3.29}$$

3.2. Event-based formulation

We developed an alternative formulation based on the assignment of flights to *take-off events* that can take place in the airport runway. To explain the reasoning of this formulation, we start with the example presented in Fig. 3, which illustrates a feasible solution of an instance with three flights and three maritime units (MU_1 to MU_3). The vertical axis of the figure illustrates the maritime units, while the horizontal axis shows the planning time horizon of the airport. As indicated in the figure, this planning horizon is represented through a finite number of airport events. These events define take-off operations from the airport runway throughout the day and hence they represent the departure times of the flights assigned to them. Thus, the number of take-off events coincides with the total number of flights in the instance (note that there are three take-off events in Fig. 3) and each flight must be assigned to an event.

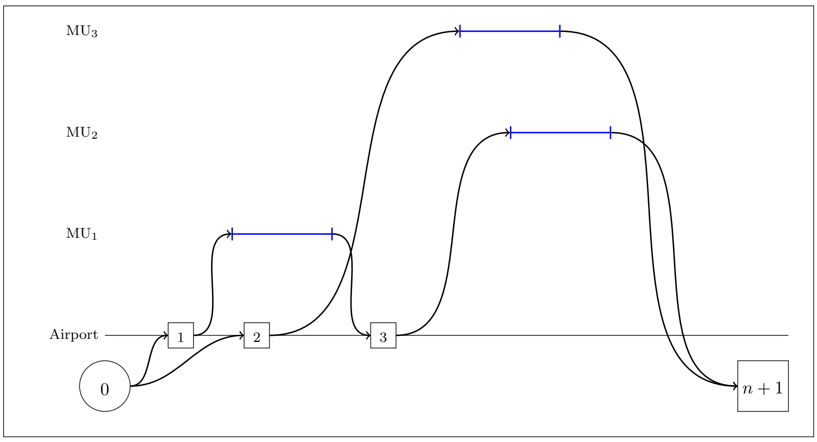


Figure 3: Schematic representation of the event-based formulation for a problem instance with three flights

There are two routes in the feasible solution presented in Fig. 3, each one associated with a helicopter. Both start at the (dummy) event 0 and finish at the (dummy) event $n + 1$, and consist of an alternating sequence of take-off events from the airport and round-trip flights to maritime units. Thus, the problem is represented by a set of take-off events $\mathcal{E} = \{1, \ldots, n\}$, such that each of these events is assigned to at most one helicopter and round-trip flight, and the additional dummy events 0 and $n+1$ represent the first and last event assigned to any helicopter that performs at least one flight. We assume a lexicographic order of events and thus event $e \in \mathcal{E}$ must start before another event $g \in \mathcal{E}$ if $e < g$. Then, in addition to variables V_h, B_i^I, B_i^{II} and D_i already defined for the network-based model, we further define the following decision variables for the event-based formulation:

$X_{eih} \in \{0,1\}$ Assumes the value of 1 if and only if helicopter h performs flight i using the take-off event e.

$W_e \geq 0$ Starting time of event e.

The objective function is defined similarly as in the network-based model using (3.1), but since there are different types of variables in the event-based model, we need to redefine the following terms:

$$f_1 := \left(|\mathcal{I}_C| - \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{I}_C} \sum_{h \in \mathcal{H}_i} X_{eih} \right); \quad f_2 := \left(|\mathcal{I}_2| - \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{I}_2} \sum_{h \in \mathcal{H}_i} X_{eih} \right); \quad f_3 := \left(|\mathcal{I}_1| - \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{I}_1} \sum_{h \in \mathcal{H}_i} X_{eih} \right);$$
$$f_4 := \left(|\mathcal{I}_0| - \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{I}_0} \sum_{h \in \mathcal{H}_i} X_{eih} \right); \quad f_{10} := \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C} \sum_{\substack{h \in \mathcal{H}_i: \\ h \neq s_i}} X_{eih}.$$

These terms have the same meaning as before, but they were rewritten using the event-based decision variables. The remaining terms, namely f_5 to f_9 and f_{11}, are defined exactly as previously stated in (3.1).

The first block of constraints guarantee that aircraft routes correspond to a sequence of airport take-off events and round-trip flights. Constraints (3.30) ensure that if flight i is not transferred to the next day, then it should be performed by a single helicopter using a single take-off event. Constraints (3.31) ensure that if the take-off event e is used, then a single helicopter uses it to perform a single flight. Constraints (3.32) relate the usage of a helicopter with the assignment of this helicopter to a flight.

$$\sum_{e \in \mathcal{E}} \sum_{h \in \mathcal{H}_i} X_{eih} \leq 1, \forall i \in \mathcal{I}; \tag{3.30}$$
$$\sum_{i \in \mathcal{I}} \sum_{h \in \mathcal{H}_i} X_{eih} \leq 1, \forall e \in \mathcal{E}; \tag{3.31}$$
$$\sum_{e \in \mathcal{E}} X_{eih} \leq V_h, \forall i \in \mathcal{I}, \forall h \in \mathcal{H}_i. \tag{3.32}$$

Constraints (3.33) ensure that take-off events do not overlap in the airport runway, as they impose a minimum time interval t^{a} between any event e and its lexicographic predecessor in set $\mathcal{E}$. Note that these constraints imply an ordering to events in $\mathcal{E}$ and impose a sequential assignment of events to time slots of duration at least t^{a}. Constraints (3.34) synchronize the starting times of two take-off events of the same helicopter. If $X_{ei}^h = 1$ then helicopter h uses the take-off event e to perform flight i and, hence, any other later event g used by the same helicopter (to perform any flight j) can only start after the starting time of e, plus the duration of flight i (tf_i), plus the inspection time (tat) of the helicopter. Constraints (3.35) set the starting time of the dummy event $n+1$ based on the last flight performed by any helicopter h (the inspection time tat should not be applied after the last flight). The value of W_{n+1} is used later to guarantee the satisfaction of the airport time window.

$$W_e \geq W_{(e-1)} + t^{\mathrm{a}} \sum_{i \in \mathcal{I}} \sum_{h \in \mathcal{H}_i} X_{eih}, \; \forall e \in \mathcal{E} \cup \{n+1\}; \tag{3.33}$$

$$W_g \geq W_e + (tf_i + tat) \left(X_{eih} + \sum_{\substack{j \in \mathcal{I}: \\ h \in H_j}} X_{gjh} - 1 \right), \; \forall e, g \in \mathcal{E}, \; g > e, \; \forall i \in \mathcal{I}, \forall h \in \mathcal{H}_i. \tag{3.34}$$

$$W_{n+1} \geq W_e + tf_i X_{eih}, \; \forall e \in \mathcal{E}, \; \forall i \in \mathcal{I}, \forall h \in \mathcal{H}_i. \tag{3.35}$$

Constraints (3.36) act as time windows for the starting time of a take-off event e assigned to a flight i. If there is an aircraft $h \in \mathcal{H}_i$ with this assignment (i.e., $X^h_{ei} = 1$), these constraints guarantee that event e starts after the originally scheduled departure time r_i of flight i, and allow a delay D_i, if necessary. Otherwise, they are redundant. Constraints (3.37) ensure that all events satisfy the airport time window.

$$r_i \sum_{h \in \mathcal{H}_i} X_{eih} \leq W_e \leq r_i \sum_{h \in \mathcal{H}_i} X_{eih} + D_i + tw^b \left(1 - \sum_{h \in \mathcal{H}_i} X_{eih} \right), \; \forall e \in \mathcal{E}, \; \forall i \in \mathcal{I}; \tag{3.36}$$

$$tw^a \leq W_e \leq tw^b, \; \forall e \in \mathcal{E} \cup \{0, n+1\}; \tag{3.37}$$

Constraints (3.38) and (3.39) are similar to constraints (3.13) and (3.14) defined in the network-based model and thus, they guarantee that there is at most one type of delay (I or II). Note that because of the summation on the right-hand of (3.38), these constraints also ensure that a flight cannot be delayed if it is not assigned to any aircraft and event.

$$B^I_i + B^{II}_i \leq \sum_{e \in \mathcal{E}} \sum_{h \in \mathcal{H}_i} X_{eih}, \forall i \in \mathcal{I}_C \cup \mathcal{I}_0; \tag{3.38}$$

$$D_i \leq d^I_{\max} B^I_i + d^{II}_{\max} B^{II}_i, \; \forall i \in \mathcal{I}_C \cup \mathcal{I}_0; \tag{3.39}$$

Constraints (3.40) impose the minimum time interval between consecutive take-offs of different helicopters performing different flights (i and j) to the same maritime unit destination (p) (i.e., if $\sum_{h \in \mathcal{H}_i} X^h_{ei} = \sum_{h \in \mathcal{H}_j} X^h_{gj} = 1$). Note that these constraints ensure a synchronization between the arrival and departure times of any two different helicopters going to the same maritime unit, avoiding these two aircraft of being on the ground of this maritime unit simultaneously.

$$W_g \geq W_e + t^{\mathrm{u}}_i \left(\sum_{h \in \mathcal{H}_i} X_{eih} + \sum_{h \in \mathcal{H}_j} X_{gjh} - 1 \right), \forall p \in \mathcal{P}, \; \forall i, j \in \mathcal{I}_p, \; i \neq j, \; \forall e, g \in \mathcal{E}, \; g > e. \tag{3.40}$$

Constraints (3.41) impose the precedence order between a transferred flight and a table flight going to the same maritime unit. Constraints (3.42) and (3.43) guarantee that the helicopter performing an entourage flight blocks the corresponding maritime unit after landing, until the end of the day.

$$\sum_{h \in \mathcal{H}_j} \sum_{g=1}^{e-1} X_{gjh} \leq 1 - \sum_{h \in \mathcal{H}_i} X_{eih}, \; \forall p \in \mathcal{P}, \; \forall i \in \mathcal{I}_2 \cup \mathcal{I}_1 \cap \mathcal{I}_p, \; \forall j \in \mathcal{I}_0 \cap \mathcal{I}_p, \; \forall e \in \mathcal{E}; \tag{3.41}$$

$$\sum_{h \in \mathcal{H}_j} \sum_{g=e+1}^{|\mathcal{E}|} X_{gjh} \leq 1 - \sum_{h \in \mathcal{H}_i} X_{eih}, \; \forall p \in \mathcal{P}, \; \forall i \in \mathcal{I}_C \cap \mathcal{I}_p, \; \forall j \in \mathcal{I} \setminus \mathcal{I}_C \cap \mathcal{I}_p, \; \forall e \in \mathcal{E}; \tag{3.42}$$

$$\sum_{g=e+1}^{|\mathcal{E}|} X^h_{gj} \leq 1 - X_{eih}, \; \forall i \in \mathcal{I}_C, \; \forall j \in \mathcal{I}, \; j \neq i, \; \forall e \in \mathcal{E}, \; \forall h \in \mathcal{H}_i \cap \mathcal{H}_j. \tag{3.43}$$

Finally, constraints (3.44)-(3.48) impose the domain of the decision variables.

$$X_{eih} \in \{0,1\},\ \forall e \in \mathcal{E},\ \forall i \in \mathcal{I},\ \forall h \in \mathcal{H}; \tag{3.44}$$
$$V_h \in \{0,1\},\ \forall h \in \mathcal{H}; \tag{3.45}$$
$$B_i^I, B_i^{II} \in \{0,1\},\ \forall i \in \mathcal{I}; \tag{3.46}$$
$$W_e \geq 0,\ \forall e \in \mathcal{E} \cup \{0, n+1\}; \tag{3.47}$$
$$D_i \geq 0,\ \forall i \in \mathcal{I}. \tag{3.48}$$

It is worth mentioning that the proposed event-based formulation can be extended to the more general case that includes not only take-off but also landing events, in order to avoid these two types of events to overlap in the runway of the airport. This can be done by duplicating the number of events in the airport and then, associating two events to each flight: one for take-off and another for landing. We have not included landing events in our approach because the company disregards overlaps between two landing events or between take-off and landing events. The time intervals between these events are not bottlenecks and the air traffic control handles such situations by maintaining the helicopter in the air and delaying its landing for a couple of minutes.

4. Heuristic approach

We propose a heuristic approach that consists of a construction phase followed by an improvement phase. The construction phase is a relax-and-repair procedure that alternates between a relaxed stage that searches for a (relaxed) solution disregarding the heterogeneity of the fleet and a repairing stage applied to this relaxed solution, until it finds a feasible solution to the problem. After that, the improvement phase tries to improve this feasible solution based on six different local search movements, as described below and in more detail in the Appendix.

4.1. Construction phase

The main steps of the construction phase are summarized in Fig. 4, in which Y^k denotes the relaxed solution vector of iteration k, OF_k denotes the objective value of the aircraft schedule obtained from Y^k, and H_{total}^{min} is a lower bound for the number of aircraft required for performing all flights – all these components are detailed in Section 4.1.1. This phase is an iterative process that starts calling the relaxed construction stage to obtain two relaxed solutions that are then modified in the repairing stage, if necessary. Depending on the objective values of the repaired solutions, it may repeat the call to these stages using different parameter choices to obtain solutions with an increased number of aircraft. The relaxed construction and repairing stages are detailed in Subsections 4.1.1 and 4.1.2, respectively.

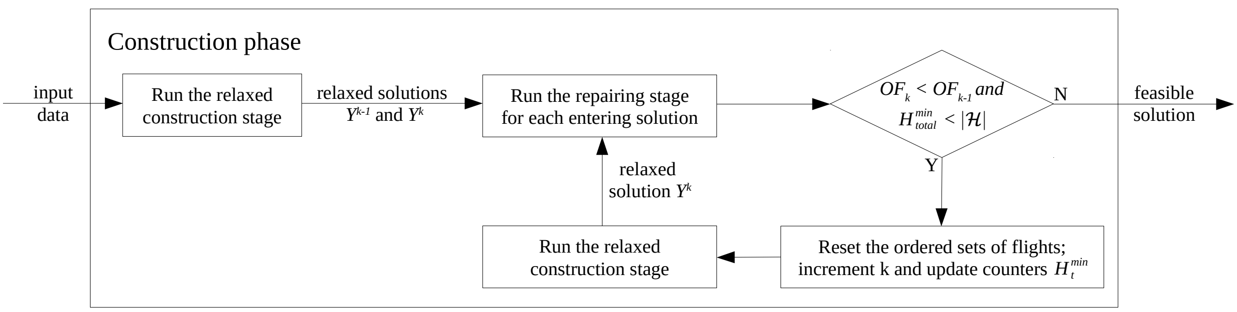


Figure 4: Main steps of the proposed construction phase

4.1.1. Relaxed construction stage

This stage seeks to allocate all flights to a given number of aircraft, taking into account all the problem characteristics defined in Section 2, except for the heterogeneity of the fleet. Before detailing the steps of this stage, we introduce the required notation. Let $\mathcal{HT} = \{r, n, p, s\}$ represent the set of aircraft type, where r is a (fictitious) generic aircraft type (used for the transferred flights), and n, p and s indicate an aircraft in the normal, pool and spot set, respectively. Let $\mathcal{Q}_t$, $\forall t \in \mathcal{HT}\backslash\{r\}$, be the subset of previously scheduled (table and entourage) flights that have an assigned aircraft of type $t \in \{n, p, s\}$ (hence, $\mathcal{Q}_t \subseteq \mathcal{I}_0 \cup \mathcal{I}_C$, $\forall t \in \mathcal{HT}\backslash\{r\}$), and $\mathcal{Q}_r = \mathcal{I}_1 \cup \mathcal{I}_2$ be the set of transferred (recovery) flights without an assigned aircraft. Note that $\mathcal{Q}_r \cap (\bigcup_{t\in\mathcal{HT}\backslash\{r\}} \mathcal{Q}_t) = \varnothing$. Assuming that all scheduled flights will depart on the current day and ignoring possible overlaps in the runway of the airport, the minimum number of aircraft of each type $t \in \mathcal{HT}$ required for performing these flights can be computed as:

$$H_t^{min} = \begin{cases} \left\lceil \frac{\sum_{i\in\mathcal{Q}_t}(tat+tf_i)}{tw^b - tw^a} \right\rceil, & t \in \mathcal{HT}\backslash\{r\}; \\ 0, & t = r. \end{cases} \tag{4.1}$$

Therefore, H_t^{min} is a lower bound for the required number of each aircraft type t. Additionally, a lower bound for the number of aircraft required for performing all flights, denoted as H_{total}^{min}, can be easily estimated by:

$$H_{total}^{min} = \sum_{t\in\mathcal{HT}} H_t^{min}. \tag{4.2}$$

At first, the relaxed construction stage builds H_{total}^{min} flight sequences that are converted into schedules and then assigned to aircraft, as detailed in the remainder of this section. This value is then increased and new flight sequences are built, in an attempt to obtain schedules with improved objective values.

Let pz_i denote the due time of each flight i, computed as the originally scheduled departure time plus the duration of the flight (i.e., $pz_i = r_i + tf_i$). For each aircraft type $t \in \mathcal{HT}$, we define an ordered list $\mathcal{L}_t = \{\ell_1, \ell_2, \ldots, \ell_{|\mathcal{Q}_t|}\}$ of flights sorted in non-descending order of pz_i. Accordingly, the order of flights in this list follows an Earliest Due Time (EDT) rule. The first step of each iteration in this stage is to define initial sequences comprising two flights: one from the beginning and another from the end of the horizon. Thereupon, sequences of flights are obtained by removing the first H_t^{min} elements from list $\mathcal{L}_t$ (i.e., ℓ_1 to $\ell_{H_t^{min}}$) and setting them (in this order) as the initial flights of the H_t^{min} flight sequences of type $t \in \mathcal{HT}\backslash\{r\}$. Next, the last H_t^{min} flights are removed from the list $\mathcal{L}_t$ (i.e., $\ell_{|\mathcal{Q}_t|-H_t^{min}+1}$ to $\ell_{|\mathcal{Q}_t|}$) and set (in this order) as the final flights of the same H_t^{min} flight sequences of type $t \in \mathcal{HT}\backslash\{r\}$. Using this rule, we create set Pr_v, for each $v = 1, \ldots, H_{total}^{min}$, composed by sequenced flights (route) to be later converted to a schedule and then assigned to a specific aircraft. They are created sequentially for each aircraft type, resulting in:

$$\begin{bmatrix} Pr_1 \\ Pr_2 \\ \vdots \\ Pr_{H_n^{min}} \end{bmatrix} = \begin{bmatrix} \{\ell_1, \ell_{|\mathcal{Q}_n|-H_n^{min}+1}\} \\ \{\ell_2, \ell_{|\mathcal{Q}_n|-H_n^{min}+2}\} \\ \vdots \\ \{\ell_{H_n^{min}}, \ell_{|\mathcal{Q}_n|}\} \end{bmatrix}; \quad \begin{bmatrix} Pr_{H_n^{min}+1} \\ Pr_{H_n^{min}+2} \\ \vdots \\ Pr_{H_n^{min}+H_p^{min}} \end{bmatrix} = \begin{bmatrix} \{\ell_1, \ell_{|\mathcal{Q}_p|-H_p^{min}+1}\} \\ \{\ell_2, \ell_{|\mathcal{Q}_p|-H_p^{min}+2}\} \\ \vdots \\ \{\ell_{H_p^{min}}, \ell_{|\mathcal{Q}_p|}\} \end{bmatrix};$$

$$\begin{bmatrix} Pr_{H_n^{min}+H_p^{min}+1} \\ Pr_{H_n^{min}+H_p^{min}+2} \\ \vdots \\ Pr_{H_n^{min}+H_p^{min}+H_s^{min}} \end{bmatrix} = \begin{bmatrix} \{\ell_1, \ell_{|\mathcal{Q}_s|-H_s^{min}+1}\} \\ \{\ell_2, \ell_{|\mathcal{Q}_s|-H_s^{min}+2}\} \\ \vdots \\ \{\ell_{H_s^{min}}, \ell_{|\mathcal{Q}_s|}\} \end{bmatrix}.$$

So, each of the first H_t^{min} elements in $\mathcal{L}_t$ becomes the first flight of each sequence for aircraft type t, and each of the last H_t^{min} elements in $\mathcal{L}_t$ becomes the last flight of each sequence for aircraft of type t. Note that as $H_r^{min} = 0$, there is no removal of transferred flights from list $\mathcal{L}_r$. Moreover, any flight inserted into Pr_v is removed from $\mathcal{L}_t$, and hence:

$$\Big(\bigcup_{t \in \mathcal{HT}} \mathcal{L}_t\Big) \cap \Big(\bigcup_{v=1}^{H_{total}^{min}} Pr_v\Big) = \varnothing.$$

In any iteration k of this heuristic, we denote as PR^k the set containing the subsets of sequenced flights $Pr_1, Pr_2, \ldots, Pr_{H_{total}^{min}}$, and represent the corresponding solution using the binary vector of components Y_{ij}^{vk}, in which Y_{ij}^{vk} assumes the value of 1, if and only if, flight i precedes flight j in the schedule v. To illustrate the sets and parameters defined so far, Fig. 5 depicts an example with 10 flights. The figure presents a table with the details of each flight in which the first column shows the original index of the flight, the second column identifies the flight type (0 for table flight and 1 for transferred) and the third column shows the aircraft type as defined above (r, n, p, s). The next three columns show parameters r_i, tat and tf_i, respectively, and the remaining three columns present the value of the numerator of the ratio in the first case of (4.1), the due time pz_i and the position of the flight in the corresponding ordered list $\mathcal{L}_t$ (EDT), respectively. The airport availability is $tw^b - tw^a = 11{:}05$. Each H_t^{min} value is shown in the upper right corner of the figure. For the flights in Fig. 5, we obtain $H_{total}^{min} = 3$ and the following initial sequence and corresponding solution, according to the definitions and the steps already described: $Pr_1 = \{4, 8\}, Y_{0,4}^{1,1} = 1, Y_{4,8}^{1,1} = 1, Y_{8,11}^{1,1} = 1$; $Pr_2 = \{7, 10\}, Y_{0,7}^{2,1} = 1, Y_{7,10}^{2,1} = 1, Y_{10,11}^{2,1} = 1$; $Pr_3 = \{9, 5\}, Y_{0,9}^{3,1} = 1, Y_{9,5}^{3,1} = 1, Y_{5,11}^{3,1} = 1$, where indices $i = 0$ and $i = 11$ represent the airport vertex.

Airport availability: 11:05

i	Flight Type	ACFT Type	r_i	tat	tf_i	$tat + tf_i$	pz_i	EDT
3	1	r	06:55	00:45	01:27	02:12	08:22	ℓ_1
1	1	r	06:55	00:45	01:42	02:27	08:37	ℓ_2
2	1	r	06:55	00:45	01:42	02:27	08:37	ℓ_3
4	0	n	07:00	00:45	01:19	02:04	08:19	ℓ_1
6	0	n	07:10	00:45	01:36	02:21	08:46	ℓ_2
8	0	n	07:05	00:45	02:00	02:45	09:05	ℓ_3
7	0	p	07:15	00:45	01:25	02:10	08:40	ℓ_1
10	0	p	07:20	00:45	01:26	02:11	08:46	ℓ_2
9	0	s	07:25	00:45	01:19	02:04	08:44	ℓ_1
5	0	s	08:00	00:45	04:00	04:45	12:00	ℓ_2

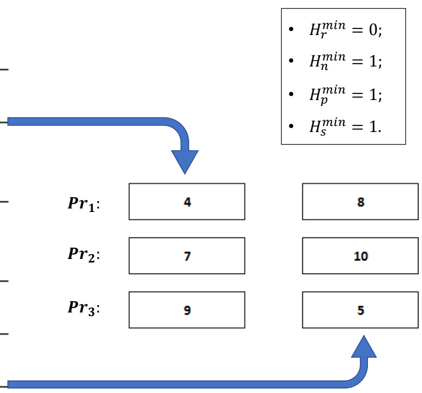


Figure 5: Application example of the initial flight sequencing

Having defined initial flight sequences as described above, the next step is to determine flight schedules from them, by setting the actual departure and arrival times of each flight i, denoted as DT_i and AT_i, respectively, which allows us to compute the resulting delays D_i and penalties in terms of variables B_i^I and B_i^{II}, as defined in Section 3. The scheduling of these sequences without considering the heterogeneity of the feet (in this stage) is done by a procedure named *GetSchedule*. This procedure is detailed in Algorithm 1 for a given iteration k of the heuristic. It uses the flag variables *factTime* and *factD* that are set to true if the tentative schedule is true, and to false, otherwise. Flag *factTime* is related to the feasibility of departure and arrival times, while *factD* regards the feasibility of delays, airport time windows and precedence requirements.

Steps 4-8 of Algorithm 1 initialize the departure and arrival times of each flight, assuming that all flights will depart at the originally planned departure time (r_i) and hence no delay incurs. Then, step 9 creates an ordered list containing all flights $i \in \mathcal{I}$ sorted by the departure times DT_i computed in the previous steps. Steps 10-29 check if these tentative schedules violate the constraints related to the overlapping of aircraft going to the same maritime unit (t_i^{u}), safety briefing (t^{a}), flight time duration (tf_i) and minimum time on the ground between two consecutive flights of same aircraft (tat). If any of these constraints are violated, the algorithm sets $factTime$ to false and modifies the departure and arrival times of flights that cause the violations. These steps are repeated until there are no more violations, and thus the schedules are feasible. Notice that in the loop defined in line 12, index j starts with the flight having the largest departure time and then loop over the next flights following the order defined by list $\mathcal{O}$.

The delays of the scheduled flights and the related penalties depending on the type of these delays are computed in steps 31-41. Flag variable $factD$ is set to false if there is at least one flight that violates (i) the maximum delay allowed in the departure time of a table or entourage flight ($DT_i - r_i > d_{max}^{II}$); ($ii$) airport time window ($AT_i > tw^b$); and/or ($iii$) the precedence requirements between transferred and previously scheduled flights going to the same maritime unit. Therefore, if *GetSchedule* detects an infeasible schedule, it returns with $factD = false$. In this case, the heuristic removes the last scheduled flights from each $Pr_1, \ldots, Pr_{H_{total}^{min}}$ and adds them back to their corresponding list $\mathcal{L}_t$, until feasible schedules are obtained.

Once feasible schedules are obtained from the flight sequences in PR^k, the heuristic tries to insert new flights to these sequences. This whole flight insertion procedure, hereafter called *InsertFlights*, is detailed in Algorithm 2. At first (see Steps 10-17), for each aircraft type $t \in \mathcal{HT}$ and flight sequence $v = 1, \ldots, H_{total}^{min}$, *InsertFlights* checks the impact of inserting each (unscheduled) flight $i \in \mathcal{L}_t$ immediately before and after the position of each scheduled flight $j \in Pr_v$. More specifically, it calculates $\overrightarrow{TD}_{ij}^{tv}$ and $\overleftarrow{TD}_{ij}^{tv}$, which are the total score of delay types that would be obtained if i was inserted immediately before and after j, respectively. This is done using also *GetSchedule* and the insertion is flagged as feasible only if this procedure returns $factD = true$. Observe that *GetSchedule* is called whenever it is necessary to compute a schedule from a sequence. Then, after checking all possible insertions, *InsertFlights* selects the tuple of indices (i^*, j^*, t^*, v^*) such that $\min\{\overrightarrow{TD}_{i^*j^*}^{t^*v^*}, \overleftarrow{TD}_{i^*j^*}^{t^*v^*}\}$ is the smallest value over all feasible insertions calculated, according to Step 18 (in the occurrence of ties, the heuristic selects the flight with the smallest index and stores the others as an alternative for the improvement phase). Lastly, on Steps 20-23, *InsertFlights* moves i^* from $\mathcal{L}_{t^*}$ to Pr_{v^*} and set $Y_{i^*j^*}^{v^*k} = 1$, if $\overrightarrow{TD}_{i^*j^*}^{t^*v^*} \leq \overleftarrow{TD}_{i^*j^*}^{t^*v^*}$, or $Y_{j^*i^*}^{v^*k} = 1$, otherwise.

The insertion of flights is repeated until it is no longer possible to assign unscheduled flights. It may happen because all flights have already been scheduled (and hence $\mathcal{L}_t$ become empty for all $t \in \mathcal{HT}$) or there is no tuple (i^*, j^*, t^*, v^*) that leads to a feasible insertion. Thus, on the Steps 25 and 26, the remaining

Algorithm 1: *GetSchedule*

Input: instance parameters, PR^k and current solution Y^k.
Output: $DT, AT, D, B^I, B^{II}, factD$.

Let $factTime$ and $factD$ be flag variables related to the feasibility of the schedule;
Let $p_i \in \mathcal{P}$ be the destination of flight i, $\forall\ i \in \mathcal{I}$;
Set $factD \leftarrow true$;
foreach $i \in \mathcal{I}$ **do**
  $DT_i \leftarrow 0; AT_i \leftarrow 0; D_i \leftarrow 0; B_i^I \leftarrow 0; B_i^{II} \leftarrow 0$;
  if $i \in PR^k$ **then**
    $DT_i \leftarrow r_i$;
    $AT_i \leftarrow DT_i + tf_i$;
Let $\mathcal{O}$ be an ordered list of all flights $i \in \mathcal{I}$ sorted in non-ascending order of departure times DT_i;
do
  Set $factTime \leftarrow true$;
  foreach $j \in \mathcal{O}$ **do**
    for $i \leftarrow 1$ **to** $|\mathcal{I}|$, $step + 1$ **do**
      if $j \in PR^k \wedge i \in PR^k \wedge DT_j \geq DT_i \wedge j \neq i$ **then**
        // Check feasibility with respect to parameters t_i^{u} and t^{a}
        if $p_j = p_i \wedge DT_j - DT_i < t_i^u$ **then**
          $factTime \leftarrow false$;
          $DT_j \leftarrow DT_j + t_i^{\mathrm{u}} - (DT_j - DT_i)$;
        if $DT_j - DT_i < t^a$ **then**
          $factTime \leftarrow false$;
          $DT_j \leftarrow DT_j + t^{\mathrm{a}} - (DT_j - DT_i)$;
      // Check feasibility with respect to tf_j and tat
      if $AT_j \neq DT_j + tf_j$ **then**
        $factTime \leftarrow false$;
        $AT_j \leftarrow DT_j + tf_j$;
      for $v \leftarrow 1$ **to** $(H_n^{min} + H_p^{min} + H_s^{min})$, $step + 1$ **do**
        if $Y_{i,j}^{v,k} = 1 \wedge DT_j - AT_i < tat$ **then**
          $factTime \leftarrow false$;
          $DT_j \leftarrow DT_j + tat - (DT_j - AT_i)$;
while $factTime \neq true$;
foreach $i \in PR^k$ **do**
  $D_i \leftarrow DT_i - r_i$;
  if $0 < D_i \leq d_{max}^I \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)$ **then**
    $B_i^I \leftarrow 1$;
  if $d_{max}^I < D_i \leq d_{max}^{II} \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)$ **then**
    $B_i^{II} \leftarrow 1$;
  if $\left[D_i > d_{max}^{II} \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)\right] \vee AT_i > tw^b$ **then**
    $factD \leftarrow false$;
  foreach $j \in PR^k$ **do**
    if $DT_j > DT_i \wedge p_i = p_j \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C \wedge j \in \mathcal{I}_1 \cup \mathcal{I}_2)$ **then**
      $factD \leftarrow false$;

unscheduled flights are moved from $\mathcal{L}_t$, $\forall\ t \in \mathcal{HT}$, to one of the following recovery sets: $\mathcal{R}_C$, if it is an entourage flight; $\mathcal{R}_2$, if it is a two-day transferred flight; $\mathcal{R}_1$ if it is a one-day transferred flight; or $\mathcal{R}_0$ if it is a table flight. In addition to instance parameters, PR^k and Y^k, this procedure receives ST as an input, which should specify the state of insertion regarding the constraints that impose the heterogeneity of the fleet. If $ST = R$, then the insertion relaxes these constraints (as in the relaxed construction stage); otherwise, $ST = NR$ imposes that the insertion must consider the heterogeneity of the fleet (as in the repairing stage).

In the next step of the heuristic, each flight sequence Pr_v is assigned to a specific aircraft h, ignoring

Algorithm 2: *InsertFlights*

Input: instance parameters, PR^k, current solution Y^k and state ST.
Output: PR^k, Y^k.

Let $\mathcal{HP}$ be a set of aircraft, initializing $\mathcal{HP} \leftarrow \{\}$;
Let $\mathcal{HA}_j$ be a set of aircraft for flight j;
// Verify if the heterogeneous fleet constraints should be ignored
if $ST = R$ **then**
  $\mathcal{HA}_j \leftarrow \{1, \ldots, H_{total}^{min}\}, \ \forall \, j \in \mathcal{I}$;
else // $ST = NR$
  $\mathcal{HP} \leftarrow \mathcal{HP} \cup \{S_j^k\}, \ \forall \, j \in \mathcal{I}$;
  $\mathcal{HA}_j \leftarrow \mathcal{HP} \cap \mathcal{H}_j, \ \forall \, j \in \mathcal{I}$;
repeat
  foreach $t \in \mathcal{HT}, \, i \in \mathcal{L}_t, \, v \in \mathcal{HA}_i, \, j \in Pr_v$ **do**
    Set i as the immediate predecessor of j in Pr_v, updating PR^k and Y^k accordingly;
    $GetSchedule(instance\ parameters, PR^k, Y^k)$;
    If the resulting schedule is feasible, set:
    $\overrightarrow{TD}_{i,j}^{t,v} \leftarrow w_8 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B_l^{II} + w_9 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B_l^{I} + w_{11} \sum_{l \in \mathcal{I}} D_l$;
    Reset i as the immediate successor of j in Pr_v, updating PR^k and Y^k accordingly;
    $GetSchedule(instance\ parameters, PR^k, Y^k)$;
    If the resulting schedule is feasible, set:
    $\overleftarrow{TD}_{i,j}^{t,v} \leftarrow w_8 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B_l^{II} + w_9 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B_l^{I} + w_{11} \sum_{l \in \mathcal{I}} D_l$;
    Remove i from Pr_v;
  $(i^*, j^*, t^*, v^*) \leftarrow \underset{i \in \mathcal{L}_t, \ t \in \mathcal{HT}, \ v \in \mathcal{HA}_i, \ j \in Pr_v}{argmin} \{\min\{\overrightarrow{TD}_{i,j}^{t,v}, \overleftarrow{TD}_{i,j}^{t,v}\}\}$;
  if $(i^*, j^*, t^*, v^*) \neq \varnothing$ **then**
    Move i^* from $\mathcal{L}_{t^*}$ to Pr_{v^*};
    if $\overrightarrow{TD}_{i^*,j^*}^{t^*,v^*} \leq \overleftarrow{TD}_{i^*,j^*}^{t^*,v^*}$ **then** $Y_{i^*,j^*}^{v^*,k} \leftarrow 1$;
    else $Y_{j^*,i^*}^{v^*,k} \leftarrow 1$;
    Update remaining solution Y^k;
  else
    foreach $t \in \mathcal{HT}, \, i \in \mathcal{L}_t$ **do**
      Move i from $\mathcal{L}_t$ to the appropriate recovery set;
until $\mathcal{L}_t = \varnothing$ *for all* $t \in \mathcal{HT}$;

the heterogeneity of the fleet. This assignment seeks to select the aircraft that, according to the original schedule, is pre-assigned to the largest number of flights in Pr_v. Let $count_{vh}$ be the number of times that aircraft $h \in \mathcal{H}$ is pre-allocated to a flight in Pr_v (i.e., the number of times $h = s_i$ for each flight i in Pr_v), for $v = 1, \ldots, H_{total}^{min}$. Then, for each v and starting with $v = 1$, the aircraft assigned to sequence Pr_v is the one with the largest $count_{vh}$, such that h has not been assigned to any other sequence in iteration k. After aircraft h is assigned to sequence Pr_v, we set $S_i^k = h$ for each flight i in Pr_v, and redefine $Pr_h := Pr_v$ and $Y_{ij}^{hk} := Y_{ij}^{vk}$. The objective function value OF_k of iteration k is then calculated by:

$$
\begin{aligned}
OF_k = {} & w_1|\mathcal{R}_c| + w_2|\mathcal{R}_2| + w_3|\mathcal{R}_1| + w_4|\mathcal{R}_0| + w_5 H_s^{\min} + w_6 H_p^{\min} + w_7 H_n^{\min} \\
& + w_8 \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C} B_i^{II} + w_9 \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C} B_i^{I} + \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C : s_i \neq S_i^k} w_{10} + w_{11} \sum_{i \in \mathcal{I}} D_i.
\end{aligned} \tag{4.3}
$$

If $OF_k < OF_{k-1}$, then the schedule defined in the current iteration is the best found so far, and therefore the heuristic sets it as the incumbent solution of the relaxed stage (we initialize $OF_0 = +\infty$). If $H_{total}^{min} < |\mathcal{H}|$, a new iteration of the current stage starts by setting $k = k+1$ and $H_t^{min} = H_t^{min}+1$, if $H_t^{min} \leq H_t$ (where H_t is the number of aircraft of type t). The choice of which H_t^{min} to increment is based on the least expensive aircraft type available. The lists $\mathcal{L}_t$, $\forall t \in \mathcal{HT}$, are reset and the steps described above are repeated. This iterative process stops if $OF_k \geq OF_{k-1}$ or $H_{total}^{min} = |\mathcal{H}|$, terminating the relaxed construction stage of the heuristic. The main steps of this stage are depicted in the flowchart presented in Fig. 6.

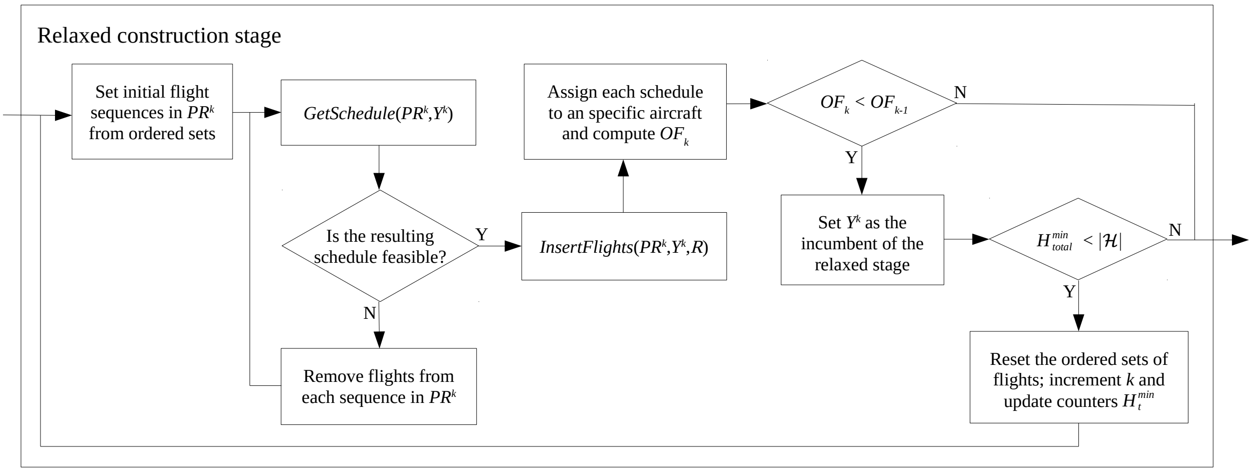


Figure 6: Flowchart of the relaxed construction stage of the proposed heuristic

4.1.2. Repairing stage

Given a solution obtained in the relaxed construction stage, the repairing stage consists of adjusting this solution, if necessary, to take into account the fleet heterogeneity constraints and hence achieve a feasible solution of the problem. It verifies whether S_i^k is compatible with $\mathcal{H}_i$ for each flight $i \in PR^k$, starting with the sets and arrays defined in the last iteration of the relaxed construction phase. If $S_i^k \notin \mathcal{H}_i$, flight i is *incompatible* with the assigned aircraft. So, this stage moves each incompatible flight i from $Pr_{S_i^k}$ to the corresponding $\mathcal{L}_t$. Next, it calls the procedure *InsertFlights* with parameter $ST = NR$ to consider the compatibility between flight i and a given aircraft h when trying to insert i into Pr_h (recall that at the end of the relaxed construction stage, each flight sequence Pr_v was assigned to an specific aircraft h), which uses function *GetSchedule* to obtain the corresponding schedule and check its feasibility in terms of time constraints. The main steps of this stage are depicted in the flowchart shown in Fig. 7.

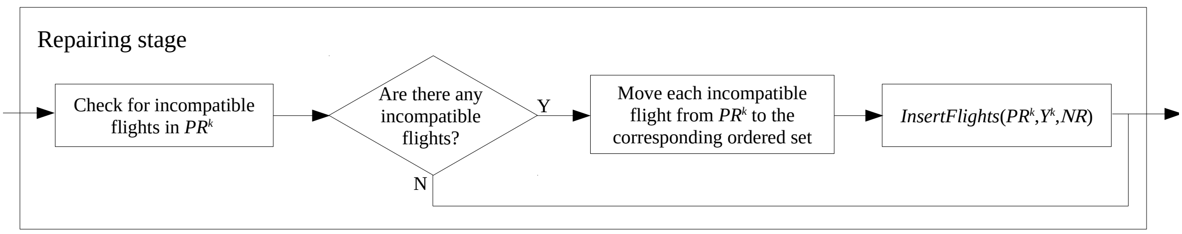


Figure 7: Flowchart of the repairing stage of the proposed heuristic

In summary, considering the overall scope of the construction phase (see Fig. 4), the heuristic applies the repairing stage on both Y^k and Y^{k-1}, the last two (relaxed) solutions obtained in the relaxed construction stage. If there is an improvement in the incumbent solution in this first comparison between OF_k and OF_{k-1}

and also if there are aircraft available, in the following iterations the heuristic executes the entire cycle contained in the relaxed construction stage with an increased H_t^{min} (adding a new aircraft to the schedule, as already discussed), in order to obtain a different relaxed base solution. Thus, the method proceeds by alternating the stages to define which aircraft will be assigned in the schedules, checks the heterogeneity of the fleet and obtains a feasible solution. The condition $OF_k < OF_{k-1}$ is always checked after the repairing stage, considering only feasible solutions (i.e., solutions that take into account the heterogeneity of the fleet). As soon as the heuristic finds no further improvement or the total number of aircraft in the fleet is reached, the construction phase ends with a feasible solution of the problem.

4.2. Improvement phase

After obtaining a feasible solution, the following procedures are executed in the presented sequence, one at a time, until no improving move is found for the aircraft schedules:

1. **Reschedule previously scheduled flights to accommodate transferred flights:** This procedure tries to insert the flights that could not be allocated in the construction phase. It starts removing an allocated flight and then places a flight that is about to be transferred (a rejected flight) in its position. For the newly removed flight, this procedure verifies the impact of reinserting this flight between any pair of flights (from the first to the last flight) in the schedule of each used aircraft that can perform it (ignoring the one in which it was previously assigned). If the proposed allocations are viable, the best recovery is then carried out and the incumbent solution is updated;

2. **Swap unscheduled flights for scheduled flights:** If the previous procedure is not able to allocate all flights, this improvement routine is then activated. It consists of swapping a flight that is allocated in the schedule for another one that was transferred when analyzing the gain condition for the objective function (example: removing a table flight from the schedule to put a recovery flight in its place). Thus, if the replacement is viable by validating the *GetSchedule* function and the weight of the flight to be changed is greater for the objective function than the one to be inserted, the incumbent solution is changed;

3. **Transfer flights to other aircraft:** This procedure is intended to improve the objective function by transferring flights to different aircraft. For each flight in the schedule, the method removes the respective flight from its original aircraft to replace it on another aircraft, analyzing whether this transfer is viable due to the problem's constraints. If a better feasible solution is found, this new schedule is stored;

4. **Inter-aircraft flight swapping:** Basically for this procedure, the heuristic performs the inter-change of flights between different aircraft, preserving the precedent and subsequent positions. For each flight of an aircraft, its change for one of the designated flights of another aircraft is simulated. This routine ends when all flights are considered by the permutation. If any change makes the objective function better than before and the schedule remains viable in terms of times $t_i^{\text{u}}, t^{\text{a}}, tat$ and AT_i, the incumbent solution starts to consider this inter-change;

5. **Intra-aircraft flight swapping:** This procedure performs the precedence rearrangement of flights from the same aircraft. In the first iteration, the routine takes the first flight in relation to the departure time and then places it between the second and third flights of that association. Using the *GetSchedule* function, it is verified if the schedule is viable and when calculating the resulting objective function, it checks if there has been an improvement. Afterwards, the first flight is placed between the third and fourth flights, and so on. When the change of all flights is considered for the aircraft in question, the feasible rearrangement that provides the greatest reduction in the objective function is then chosen. The method is finalized by applying this routine to all aircraft used;

6. **Reduce delay types:** After performing the previous five procedures, this routine that seeks to decrease the count of flights with type I and II delay is activated. Through the current schedule, the heuristic analyzes which flights could return to their departure and arrival times changed to the initial values of planned time (without delay), by increasing the delay of flights located close to these. Thus, for each flight i contained in the schedule with $D_i > 0$ is placed $DT_i = r_i$ and $AT_i = r_i + tf_i$, and then a routine is performed that "pushes" the other flights to the detriment of the time constraints provided by the problem, similar to what we have in *GetSchedule*. If the new schedule is feasible and if there was a decrease in the objective function due to the reduction of the delays types, then the incumbent solution is modified.

The general scheme of the heuristic and these procedures of the improvement phase are presented in detail in the Appendix.

5. Results

In this section, we report the results of computational experiments with the network flow and the event-based formulations proposed in Section 3 and the heuristic approaches developed in Section 4. All approaches were coded in C++ and, in particular, the two models rely on the Concert library and the general-purpose mixed-integer programming solver of the IBM CPLEX Optimization Studio 12.9. All experiments were run on a Linux PC with an Intel Core i7 4790 3.6 GHz CPU and 16 GB of RAM.

The instances of the present ARP-PD are based on real-life data provided by the Brazilian oil company, found on daily flights operated in three airports used by the companionship, named hereafter as airports A, B and C. Table 1 presents the main information regarding these instances: name (Instance), number of flights ($|\mathcal{I}|$), number of aircraft in the normal, pool and spot sets of helicopters ($|\mathcal{H}_n|$, $|\mathcal{H}_p|$ and $|\mathcal{H}_s|$, respectively), number of maritime units ($|\mathcal{P}|$), number of table flights ($|\mathcal{I}_0|$), number of transferred flights from the previous day ($|\mathcal{I}_1|$), number of transferred flights from two or more days before ($|\mathcal{I}_2|$) and number of entourage flights ($|\mathcal{I}_C|$). A total of twenty instances were considered, where seven are from airport A, namely I8A, I9A, ..., I14A, which are small-sized instances with 8, 9, ..., 14 flights, respectively; eight instances are based on airport B, namely I15B, I18B, ..., I30B, which are medium-sized instances with 15, 18, ..., 30 flights, respectively; and five large-sized instances of airport C, namely I33C, I35C, ..., I45C, with 33, 35, ..., 45 flights, respectively.

The penalty values indicated by the company operators for the different terms of the objective function were: $w_1 = 320$, $w_2 = 240$, $w_3 = 160$, $w_4 = 80$, $w_5 = 30$, $w_6 = 25$, $w_7 = 20$, $w_8 = 10$, $w_9 = 1$, $w_{10} = 0.5$ and

Instance	$\|\mathcal{I}\|$	$\|\mathcal{H}_n\|$	$\|\mathcal{H}_p\|$	$\|\mathcal{H}_s\|$	$\|\mathcal{P}\|$	$\|\mathcal{I}_0\|$	$\|\mathcal{I}_1\|$	$\|\mathcal{I}_2\|$	$\|\mathcal{I}_C\|$
I8A	8	2	1	0	6	6	1	1	0
I9A	9	2	1	0	7	7	1	1	0
I10A	10	2	1	0	7	8	1	1	0
I11A	11	2	3	1	10	7	3	0	1
I12A	12	3	0	0	9	11	1	0	0
I13A	13	2	1	1	10	7	5	1	0
I14A	14	4	2	1	11	7	4	2	1
I15B	15	3	2	2	11	11	2	1	1
I18B	18	6	2	0	15	11	5	0	2
I20B	20	4	3	1	12	14	6	0	0
I22B	22	4	1	1	13	17	5	0	0
I25B	25	12	0	0	20	20	5	0	0
I27B	27	10	3	0	22	21	6	0	0
I28B	28	6	2	0	19	13	13	2	0
I30B	30	7	2	2	20	12	16	1	1
I33C	33	5	2	1	21	12	18	3	0
I35C	35	12	0	0	26	22	10	2	1
I37C	37	11	0	0	17	30	7	0	0
I38C	38	11	0	0	24	15	20	3	0
I45C	45	11	0	0	27	22	20	3	0

Table 1: Main information of the 20 real-life-based instances provided by the company

$w_{11} = 0.001$. They also provided the following parameter values: $t^{\mathrm{a}} = 5$ min, $tat = 45$ min, $t^{\mathrm{u}} = 15$ min, $d^{I}_{\max} = 15$ min, $d^{II}_{\max} = 240$ min, $[tw^a, tw^b] =$ [7:00am, 6:00pm].

In addition to the twenty real-life-based instances described in Table 1 (here called Scenario 0), we also defined other instances grouped into eight other different scenarios and based on the larger instances I37C, I38C and I45C of Scenario 0. All instances of these 8 scenarios were generated by randomly changing some parameters of instances I37C, I38C and I45C, as follows:

- Scenario 1: the (previous) assignment of helicopters to the table flights is randomly modified. The motivation of this scenario is to verify the ability of the approach to find effective reschedules with economical helicopter reassignments.

- Scenario 2: helicopter types (normal, pool and spot) are randomly modified. The motivation is to verify the ability of the approach to find economical helicopter assignments, including cases that require spot helicopters.

- Scenario 3: flight types (table, one-day transferred, two-or-more-day transferred and entourage) are randomly modified. The scheduled departure times and the assigned helicopters to these flights were adjusted because a transferred flight can depart at the beginning of the airport time windows and it does not have a previously assigned helicopter. The motivation of this scenario is to verify how the method recovers from transferred flights, including cases that have entourage flights.

- Scenario 4: random changes to the scheduled departure time (r_i) and the duration (tf_i) of the table flights.

- Scenario 5: random changes to the minimum time between consecutive flights of the airport runway (t^{a}), the time on the ground of a helicopter at the airport (tat), and the limit times for delays of types I and II ($d^{I}_{\max}$ and $d^{II}_{\max}$, respectively).

- Scenario 6: the helicopter-flight 0-1 matrix that indicates which helicopter is able to perform each flight is randomly changed.
- Scenario 7: random changes to the maritime unit destination of the table flights.
- Scenario 8: we included more transferred flights than in Scenario 1, generated by randomly increasing the number of transferred flights by a percentage in the interval [1%, 30%], while randomly decreasing the number of table flights by the same percentage. The number of one-day transferred and two-or-more-day transferred flights were equally sorted in the total number of transferred flights increased.
- Scenario 9: it alludes to a sensitivity analysis, in which three types of tests are performed changing the penalties of the objective function. Let us define the following sets: (i) weight families: $\mathcal{F}_1 = \{1, 2, 3, 4\}$ (types of transfers), $\mathcal{F}_2 = \{5, 6, 7\}$ (types of aircraft) and $\mathcal{F}_3 = \{8, 9, 11\}$ (types of delays); (ii) weight range: $\mathcal{RG} = \mathcal{F}_2 \cup \{10\} \cup \mathcal{F}_3$, and rg_j denote the j-th element of $\mathcal{RG}$. The first test consists of canceling (i.e., making null) the values of w_i for families $\mathcal{F}_2$ and $\mathcal{F}_3$; thus, $Test^1_t : w_i = 0, \forall\ i \in \mathcal{F}_t, t = 2, 3$. We note that the omission of family $\mathcal{F}_1$ in $Test^1_t$ is justified by the fact that it characterizes the fundamental objective of the present ARP-PD. The second test levels the values for each weight family. For this, we use a vector AW_t with the following values: $AW_1 = 200$, $AW_2 = 25$ and $AW_3 = 0.1$; so, $Test^2_t : w_i = AW_t, \forall\ i \in \mathcal{F}_t, t = 1, 2, 3$. The third test cancels the weights of the objective function in a descending and accumulated way of $\mathcal{RG}$, starting with w_{11} and going up to w_5, that is: $Test^3_t : w_i = 0, \forall\ i \in \bigcup_{j=(7-t+1)}^{7} rg_j, t = 1, \ldots, 7$ (we also note the omission of family $\mathcal{F}_1$ in $Test^3_t$).

5.1. Results of the MIP formulations

Tables 2 and 3 present the results obtained using CPLEX, within the time limit of 1 hour, for the network- and event-based models, respectively. For each instance and each model, we ran the general-purpose branch-and-cut (B&C) method of CPLEX using the following four configurations: (i) default settings; (ii) with the local branching heuristic turned on; (iii) with the relaxation induced neighborhood search (RINS) heuristic turned on; and (iv) with both heuristics turned on. Local branching and RINS are heuristics that explore neighborhoods of the current incumbent solution to try to find a new, improved incumbent. They are embedded in the solver and were turned on by changing one parameter before calling the B&C method. The first column of Tables 2 and 3 show the instance name (Inst). Then, for each B&C configuration, the tables present the lower bound (f_{LB}), upper bound (f), relative optimality gap (Gap) as a percentage, and computational time (Time) in seconds for the solution obtained with the corresponding configuration. The optimality gap is presented as provided by CPLEX and is computed as $100 \times (f - f_{LB})/(f + 10^{-10})$. In each table, we highlight in bold the best upper bounds obtained among the four configurations using the same model, except when all approaches resulted in the same value. Additionally, the tables show the letters 'tl' in column Time if the corresponding method stopped after reaching the time limit of 1 hour, and the letter 'm' if the method stopped due to memory overflow.

The results in Table 2 show that CPLEX could solve to optimality all instances related to airport A, with up to 14 flights, using the network-based model and the B&C method in its default settings as well as with the local branching heuristic turned on. The B&C configurations (iii) and (iv), both with the RINS heuristic, required larger computational times for these instances than the other configurations and could not

	B&C				B&C + local branching				B&C + RINS				B&C + both			
Inst	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time
I8A	42.13	42.13	0.00	0.07	42.13	42.13	0.00	0.06	42.13	42.13	0.00	0.08	42.13	42.13	0.00	0.08
I9A	67.03	67.03	0.00	0.13	67.03	67.03	0.00	0.10	67.03	67.03	0.00	0.15	67.03	67.03	0.00	0.14
I10A	66.64	66.64	0.00	0.60	66.64	66.64	0.00	0.87	66.64	66.64	0.00	2.08	66.64	66.64	0.00	1.21
I11A	91.03	91.03	0.00	1.70	91.03	91.03	0.00	1.97	91.03	91.03	0.00	87.72	91.03	91.03	0.00	83.68
I12A	71.19	71.19	0.00	1.23	71.19	71.19	0.00	1.41	71.19	71.19	0.00	93.45	71.19	71.19	0.00	107.43
I13A	68.13	68.13	0.00	196.80	68.13	68.13	0.00	80.35	54.12	68.13	20.56	tl	47.85	68.13	29.76	tl
I14A	83.68	83.68	0.00	404.36	83.68	83.68	0.00	487.58	67.85	83.68	18.92	tl	65.71	83.68	21.48	tl
I15B	90.90	152.32	40.32	tl	89.22	152.32	41.43	tl	68.32	**144.36**	52.68	tl	68.18	144.36	52.77	tl
I18B	126.67	147.92	14.37	tl	126.54	**147.92**	14.45	tl	122.21	**147.92**	17.38	tl	122.02	148.18	17.66	tl
I20B	79.09	108.24	26.93	tl	90.81	108.24	16.10	tl	68.01	108.24	37.16	tl	68.71	108.24	36.52	tl
I22B	105.32	131.27	19.77	tl	111.19	131.27	15.29	tl	91.39	**123.91**	26.24	tl	86.09	**123.91**	30.52	tl
I25B	102.39	**227.17**	54.93	tl	109.88	238.05	53.84	tl	99.55	227.58	56.26	tl	99.58	**227.17**	56.17	tl
I27B	132.25	189.01	30.03	tl	128.64	198.98	35.35	tl	127.53	**186.48**	31.61	tl	127.85	**186.48**	31.44	tl
I28B	183.30	274.60	33.25	tl	102.24	**260.29**	60.72	tl	95.56	270.17	64.63	tl	96.08	280.21	65.71	tl
I30B	48.11	729.12	93.40	tl	47.16	430.92	89.06	tl	47.05	295.72	84.09	tl	47.08	**282.43**	83.33	tl
I33C	88.63	518.15	82.89	tl	88.68	1,153.06	92.31	tl	88.60	495.29	82.11	tl	88.59	**466.43**	81.01	tl
I35C	188.07	387.29	51.44	tl	188.12	377.26	50.14	tl	187.77	310.46	39.52	tl	186.71	**309.02**	39.58	tl
I37C	118.04	326.63	63.86	tl	109.31	327.62	66.64	tl	107.01	308.42	65.30	tl	107.36	**259.38**	58.61	tl
I38C				m	46.57	1,095.21	95.75	tl	46.41	**527.71**	91.21	tl	46.41	527.72	91.21	tl
I45C				m	50.76	1,837.12	97.24	tl	51.59	**939.59**	94.51	tl	51.58	960.63	94.63	tl

Table 2: Results obtained for the network-based model using the general-purpose B&C solver of CPLEX with default settings as well as with two embedded heuristics

	B&C				B&C + local branching				B&C + RINS				B&C + both			
Inst	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time	f_{LB}	f	Gap	Time
I8A	42.13	42.13	0.00	0.15	42.13	42.13	0.00	0.20	42.13	42.13	0.00	0.13	42.13	42.13	0.00	0.18
I9A	67.03	67.03	0.00	0.59	67.03	67.03	0.00	0.93	67.03	67.03	0.00	113.40	67.03	67.03	0.00	70.73
I10A	66.64	66.64	0.00	3.68	66.64	66.64	0.00	3.84	66.64	66.64	0.00	1,443.55	66.64	66.64	0.00	1,889.66
I11A	91.03	91.03	0.00	18.66	91.03	91.03	0.00	78.65	76.07	91.03	16.40	tl	76.82	91.03	15.60	tl
I12A	71.19	71.19	0.00	21.99	71.19	71.19	0.00	14.92	71.19	71.19	0.00	1,980.35	71.19	71.19	0.00	999.81
I13A	68.13	**68.13**	0.00	220.44	68.13	**68.13**	0.00	276.75	68.13	66.83	1.90	tl	66.73	**68.13**	2.04	tl
I14A	65.10	**83.68**	22.20	tl	68.46	84.27	18.80	tl	62.42	**83.68**	25.40	tl	62.34	83.68	25.51	tl
I15B	73.85	**144.36**	48.80	tl	72.79	152.34	52.20	tl	68.48	152.34	55.00	tl	68.59	152.34	54.97	tl
I18B	121.79	**148.18**	17.80	tl	121.70	157.36	22.70	tl	92.08	**148.18**	37.90	tl	92.13	**148.18**	37.83	tl
I20B	82.24	**109.80**	25.10	tl	82.15	132.36	37.90	tl	81.50	131.99	38.30	tl	81.50	131.99	38.25	tl
I22B	82.08	160.49	48.90	tl	82.07	161.63	49.20	tl	75.61	150.91	49.90	tl	75.61	**141.41**	46.53	tl
I25B	135.15	337.57	60.00	tl	134.97	322.23	58.10	tl	129.88	257.35	49.50	tl	129.88	**256.92**	49.45	tl
I27B	145.93	**191.80**	23.90	tl	140.81	218.97	35.70	tl	137.63	232.02	40.70	tl	137.63	261.45	47.36	tl
I28B	210.10	252.29	16.70	tl	209.10	**252.22**	17.10	tl	159.09	259.33	38.70	tl	192.18	**252.22**	23.80	tl
I30B	122.50	336.58	63.60	tl	122.50	**288.31**	57.50	tl	122.50	526.86	76.70	tl	122.50	368.58	66.76	tl
I33C	151.10	437.63	65.47	tl	151.69	**332.88**	54.43	tl	150.50	676.13	77.74	tl	150.50	427.98	64.83	tl
I35C	143.50	**768.53**	81.33	tl	143.50	**768.53**	81.33	tl	143.50	1,125.26	87.25	tl	143.50	936.03	84.67	tl
I37C	151.10	**865.46**	82.50	tl	151.10	**865.46**	82.50	tl	151.10	**865.46**	82.50	tl	151.10	941.61	83.95	tl
I38C	129.80	2,925.24	95.56	tl	129.80	2,925.24	95.56	tl	129.80	2,925.24	95.56	tl	129.80	2,925.24	95.56	tl
I45C	156.50	5,400.00	97.10	tl	156.50	5,400.00	97.10	tl	156.50	5,400.00	97.10	tl	156.50	5,400.00	97.10	tl

Table 3: Results obtained for the event-based model using the general-purpose B&C solver of CPLEX with default settings as well as with two embedded heuristics

prove optimality for two of them within the time limit, namely I13A and I14A, although they also obtained optimal solutions for them. In spite of these relatively poor results on instances related to airport A, the RINS heuristic promoted the best overall performance of the solver for the remaining instances, particularly for the largest ones in the set. Indeed, the last two B&C configurations obtained the best upper bound values for more instances than the other configurations. For instances related to airport C, which are the largest ones, the upper bound values were significantly better than the ones obtained by configurations (i) and (ii). For the two largest instances (I38C and I45C), the standalone B&C failed to provide a solution due to memory overflow, and the method using the local branching heuristic finished with solutions that were considerably worse than the solutions provided by the B&C with RINS only. The B&C with both heuristics obtained the best upper bounds for the other three instances related to airport C. Finally, in any of the configurations, we observe that the optimality gaps were significantly large for many instances, particularly for the largest ones.

The heuristics embedded in CPLEX were helpful also when using the event-based model, as indicate the results presented in Table 3. Yet, the standalone B&C found feasible solutions to all instances and obtained the best upper bounds among the four configurations for several of them. Recall that with the network-based model and using this same configuration, CPLEX failed to obtain feasible solutions to the two largest instances, and obtained the best upper bound for only one instance (I25B) in comparison to the other configurations. Additionally, it is worth mentioning that for the largest instances, the lower bounds obtained with the event-based model are significantly better than those obtained with the other model. None of the approaches using the event-based model could solve instance I14A to proven optimality though, which was previously solved by using the network-based model. Regarding the instances related to airports B and C, the B&C with local branching only (second configuration) resulted in the best upper bounds for five of them, namely I28B, I30B, I33C, I35C and I37C, while the B&C with RINS only (third configuration) obtained the best results for two of them, namely I18B, I25B. The B&C with both heuristics presented an inferior performance on instances related to airport C, although it obtained the best upper bounds in four instances related to airport B (I18B, I22B, I25B and I28B). Similar to the results obtained with the network-based model, the optimality gaps were significantly large for the large instances, particularly for those related to airport C.

Table 4 presents the best bounds obtained in the experiments with the two models and four B&C configurations, and details the values of each term in the objective function for the solution corresponding to the presented upper bound. The first column (Inst) presents the instance name. The second and third columns give the best lower bound (f^*_{LB}) and the best upper bound (f^*) obtained in the experiments, considering the two models and the four configurations. Columns 4 to 14 present the values for each of the 11 weight terms $w_1 f_1$ to $w_{11} f_{11}$ of the objective function considering the best solution (i.e., the solution corresponding to f^*). The next two columns, 15 and 16, show the computational time (in seconds) to obtain this solution and the total number of flights not scheduled and thus transferred to the next day (nR). Finally, the last column indicates both the formulation (1: network-based; 2: event-based) and the B&C configuration (D: default; L: with local branching; R: with RINS; B: with both heuristics) that resulted in the best lower (f^*_{LB}) and upper (f^*) bounds, respectively. For example, 2D-1R means that f^*_{LB} was obtained using the event-based model and the B&C with default settings, while f^* was obtained using the network-based model and the B&C with RINS only. In case of ties, we consider the configuration with the shortest computational time and then the simplest configuration (i.e. first the B&C with default settings, then the B&C with one heuristic only and finally the configuration with both).

As the results in Table 4 indicate, the exact approaches obtained solutions that schedule all flights for instances I8A to I30B. However, we observe that on instances related to airport C, the largest ones, the poor performance of the B&C methods is related to the difficulty of finding a schedule that include all flights. In all best solutions obtained for these largest instances, there is at least one table flight that is no scheduled. For instance I45, in addition to three tables flights, there is also one transferred flight from the previous day that was not scheduled. The table also highlights that the B&C approaches using the network-based model obtained most of the best lower and upper bounds and were the fastest when the configurations using the event-based model obtained the same values. The RINS heuristic promoted the best performance for

Inst	f_{LB}^*	f^*	w_1f_1	w_2f_2	w_3f_3	w_4f_4	w_5f_5	w_6f_6	w_7f_7	w_8f_8	w_9f_9	$w_{10}f_{10}$	$w_{11}f_{11}$	Time	nR	Conf.
I8A	42.13	42.13	0	0	0	0	0	0	40	0	2	0.0	0.13	0.06	0	1L-1L
I9A	67.03	67.03	0	0	0	0	0	25	40	0	2	0.0	0.03	0.10	0	1L-1L
I10A	66.64	66.64	0	0	0	0	0	25	40	0	1	0.5	0.14	0.60	0	1D-1D
I11A	91.03	91.03	0	0	0	0	0	50	40	0	0	0.5	0.53	1.70	0	1D-1D
I12A	71.19	71.19	0	0	0	0	0	0	60	10	1	0.0	0.19	1.23	0	1D-1D
I13A	68.13	68.13	0	0	0	0	0	25	40	0	0	1.5	1.63	80.35	0	1L-1L
I14A	83.68	83.68	0	0	0	0	0	0	80	0	1	0.5	2.18	404.36	0	1D-1D
I15B	90.90	144.36	0	0	0	0	0	50	60	30	2	1.0	1.36	tl	0	1D-1R
I18B	126.67	147.92	0	0	0	0	0	25	120	0	0	2.0	0.92	tl	0	1D-1D
I20B	90.81	108.24	0	0	0	0	0	25	80	0	0	2.0	1.24	tl	0	1L-1D
I22B	111.19	123.91	0	0	0	0	0	25	80	10	5	3.0	0.91	tl	0	1L-1R
I25B	135.15	227.17	0	0	0	0	0	0	180	40	1	4.5	1.67	tl	0	2D-1D
I27B	145.93	186.48	0	0	0	0	0	0	180	0	3	2.5	0.98	tl	0	2D-1R
I28B	210.10	252.22	0	0	0	0	0	25	100	120	6	0.0	1.22	tl	0	2D-2L
I30B	122.50	282.43	0	0	0	0	0	50	140	90	1	2.5	4.81	tl	0	2D-1B
I33C	151.69	332.88	0	0	0	80	30	50	100	60	4	2.0	6.88	tl	1	2L-2L
I35C	188.12	309.02	0	0	0	80	0	0	220	0	5	2.5	2.96	tl	1	1L-1B
I37C	151.10	259.38	0	0	0	80	0	0	200	20	4	3.5	0.92	tl	1	2D-1B
I38C	129.80	527.71	0	0	0	240	0	0	200	80	2	0.5	5.21	tl	3	2D-1R
I45C	156.50	939.59	0	0	160	480	0	0	200	90	2	3.0	4.59	tl	8	2D-1R

Table 4: Best results obtained in the experiments with the two models and the four B&C configurations

the approaches using the network-based model, either alone or in combination with the local branching, in particular for the largest instances. Also for these instances, the event-based model was important to obtain the best lower bounds for most of them, using the standalone B&C.

5.2. Results of the heuristic approach

The fact that the solver failed to obtain feasible solutions with reasonable optimality gaps within the time limit using the proposed models for the larger problem instances highlights the importance of developing effective tailor-made heuristics, such as the ones proposed in Section 4 and analyzed in what follows. We first present the results for instances of Scenario 0 and then for instances of Scenarios 1 to 8.

5.2.1. Results for the instances of Scenario 0

Table 5 presents the heuristic results for the twenty real-life-based instances (Scenario 0). The meaning of each column is the same as in Table 4. These results show that the heuristic was able to obtain feasible solutions including all flights for all instances. Hence, the obtained flight reschedules recover all transferred flights of the last days in addition to schedule all table flights of the day.

When comparing the results of the heuristic with the best bounds f obtained by the models (instances of airport A), as presented in Table 4, we note that the corresponding optimality gaps are small (less than 6.3%), indicating the quality of the heuristic solutions. It should be observed that these solutions were found in 0.077 seconds from the heuristic, while the two models spend 552.46 and 86.41 seconds, respectively on average. For the medium instances (instances of airport B), the heuristic is still able to find good reschedules as all transferred flights are recovered in affordable computational times (4.2 s for central tendency), while the best feasible solutions obtained by the B&C approaches reach the run time limit (1 hour). Finally, for the larger instances of Scenario 0 (airport C), the heuristic surpasses the optimization models (including the applications of local branching and RINS methods), where the results of models were considerably worse and failed to schedule all flights. Analyzing f, the heuristic gaps are 87.1% and 51.7% better than solving the modeling, in this order and on average. Particularly, in instance I33C, the heuristic solution uses the entire

Inst	f	w_1f_1	w_2f_2	w_3f_3	w_4f_4	w_5f_5	w_6f_6	w_7f_7	w_8f_8	w_9f_9	$w_{10}f_{10}$	$w_{11}f_{11}$	Time	nR
I8A	42.13	0	0	0	0	0	0	40	0	2	0.0	0.13	0.02	0
I9A	67.03	0	0	0	0	0	25	40	0	2	0.0	0.03	0.07	0
I10A	66.65	0	0	0	0	0	25	40	0	1	0.5	0.15	0.05	0
I11A	102.40	0	0	0	0	0	50	40	10	0	2.0	0.40	0.08	0
I12A	72.18	0	0	0	0	0	0	60	10	0	2.0	0.18	0.08	0
I13A	88.63	0	0	0	0	30	25	20	10	1	1.0	1.63	0.70	0
I14A	83.98	0	0	0	0	0	0	80	0	1	0.5	2.48	0.15	0
I15B	161.48	0	0	0	0	30	50	60	20	0	0.5	0.98	0.18	0
I18B	148.20	0	0	0	0	0	25	120	0	1	1.5	0.70	0.81	0
I20B	130.60	0	0	0	0	0	25	80	20	3	1.5	1.10	1.38	0
I22B	131.28	0	0	0	0	0	25	80	20	4	1.5	0.78	1.70	0
I25B	208.67	0	0	0	0	0	0	180	20	2	5.0	1.67	3.76	0
I27B	207.71	0	0	0	0	0	0	180	20	3	3.0	1.71	6.16	0
I28B	211.63	0	0	0	0	0	50	120	30	5	2.0	4.63	6.27	0
I30B	201.93	0	0	0	0	30	50	100	10	4	2.0	5.93	13.79	0
I33C	260.70	0	0	0	0	30	50	100	70	2	1.5	7.20	11.77	0
I35C	250.83	0	0	0	0	0	0	220	20	4	3.0	3.83	24.22	0
I37C	243.77	0	0	0	0	0	0	200	30	6	6.5	1.27	28.94	0
I38C	214.96	0	0	0	0	0	0	180	20	6	1.5	7.46	63.74	0
I45C	277.41	0	0	0	0	0	0	220	40	5	5.0	7.41	109.63	0

Table 5: Results of the heuristic for the 20 instances of Scenario 0

fleet available to not transfer flights to the next day, and changes three previously assigned helicopters. This solution results in two delays of type I and seven delays of type II. The heuristic solution for instance I35C includes the rescheduling of the entourage flight and saves one helicopter. Note that the solution changes six previously assigned helicopters, and involves four delays of type I and two of type II. For instance I37C, it also reschedules all flights, saving one helicopter and changing 13 previously assigned helicopters. This solution results in six and three delays of types I and II, respectively. For I38C, the heuristic solution uses only 9 of the 11 available helicopters, changing three previously assigned helicopters. This solution implies in six delays of type I and two delays of type II. The last instance, I45C, was assembled by combining the instances I38C and I37C. According to the company's history, I37C is a subsequent scheduling of I38C, which has 7 transferred flight (see Table 1) from I38C (which were table flights of the current day). This is because the company solution was unable to reschedule all table flights, thus forming I45C ($38 + 7 = 45$). The heuristic solution was able to reschedule all transferred and table flights using the whole fleet of helicopters and changing ten previously assigned helicopters. This solution implies in five and four delays of types I and II, in this order. These results reinforce the effectiveness of the heuristic approach when solving real-life instances, indicating its potential to help decision making in practice.

Table 6 shows the relative gaps (in percentages) of the solution values obtained by the heuristic (f^{heur}) with respect to the best lower (f^*_{LB}) and upper (f^*) bounds obtained by the optimization models (as presented in Table 4). The values of columns Gap_{LB} correspond to the optimality gap with respect to the best lower bound given by $Gap_{LB} = 100\times(f^{heur} - f^*_{LB})/(f^*_{LB} + 10^{-10})$, whereas the values of columns Gap_{UB} refer to the gap with respect to the best upper bound given by $Gap_{UB} = 100\times(f^{heur} - f^*)/(f^* + 10^{-10})$. The Gap_{UB} values indicate that the solutions obtained by the heuristic for the larger instances are significantly better than the best solutions obtained by the models. The average gap with respect to the upper bound was -35.25% for the instances of airport C, reaching -59.26% and -70.48% for the two largest instances (I38C and I145C). For the smaller instances, the solutions obtained by the models are superior, but it is worth remarking that they correspond to proven optimal solutions. We note that there are large

Gap_{LB} values in the table as the models have weak linear relaxations (as indicated by the results of Tables 2 and 3).

Inst	Gap_{LB}	Gap_{UB}	Inst	Gap_{LB}	Gap_{UB}	Inst	Gap_{LB}	Gap_{UB}
I8A	0.000	0.000	I15B	77.646	11.859	I33C	71.864	-21.683
I9A	0.000	0.000	I18B	16.997	0.189	I35C	33.335	-18.830
I10A	0.015	0.015	I20B	43.817	20.658	I37C	61.330	-6.018
I11A	12.490	12.490	I22B	18.068	5.948	I38C	65.609	-59.266
I12A	1.391	1.391	I25B	54.399	-8.144	I45C	77.259	-70.475
I13A	30.090	30.090	I27B	42.335	11.385			
I14A	0.359	0.359	I28B	0.728	-16.093			
			I30B	64.841	-28.503			

Table 6: Gaps of the heuristic solutions with respect to the best lower and upper bounds obtained by the models

A better way to evaluate the solution quality of the heuristic method could be to perform a computational experiment with the B&C approaches using a longer run-time limit. In this way, we chose the largest instance in the set (I45C) and ran the best B&C configuration for each model, with the time limit of $24h$. Table 7 shows the best results obtained for each model from this experiment. Note that f^* has decreased considerably and that f^*_{LB} has increased slightly. Still, no approach has managed to allocate all flights in the schedules, which reinforces the quality of the heuristic solution, at least in the practical scope of application.

Inst	f^*_{LB}	f^*	w_1f_1	w_2f_2	w_3f_3	w_4f_4	w_5f_5	w_6f_6	w_7f_7	w_8f_8	w_9f_9	$w_{10}f_{10}$	$w_{11}f_{11}$	Time	nR	Conf.
I45C	51.66	560.21	0	0	0	240	0	0	220	90	2	3	5.206	24h	3	1R
I45C	158.10	506.08	0	0	0	160	0	0	220	110	2	5.5	8.582	24h	2	2D

Table 7: Best results from the computational experiment with the time limit of 24h for instance I45C

5.2.2. Results for the instances of Scenarios 1-8

Table 8 presents the results of experiments with instances of Scenarios 1 to 8. The first column in the table shows the scenario number and the remaining columns have the same meaning as in Table 5. The presented results for each scenario correspond to the average (arithmetic mean) over three randomly generated instances for each original instance I37C, I38C and I45C, resulting in nine instances per scenario.

Comparing the results of Scenario 1 with the results obtained for the real-life instances (Scenario 0) also using the heuristic, we note an increase in the reassignments of helicopters to flights, as expected. In some cases, we see a reduction in the total number of helicopters used, still rescheduling all flights. In other cases, the whole fleet had to be used in order to reschedule all flights. Regarding Scenario 2, mostly when comparing the availability of the fleet to its actual usage, the heuristic was able to use less expensive helicopters (i.e., more normal than pool, and more pool than spot helicopters) while rescheduling all flights. It is worth mentioning that the penalties related to the use of aircraft are not comparative, because the random generation made the normal type decrease and the pool and spot types increase (before with zero count) in the instances. Consequently, even if the comparison was made between optimal values of both scenarios, the real would be better, for example. Scenario 3 also shows that even after modifying the types of flights, the heuristic was also able to reschedule all flights. In some instances, the solutions reduced the flight delays and the helicopter reassignments by using more helicopters. In Scenario 4, we note that the changes in the departure times and duration of the flights did not significantly affect the heuristic solutions

Scen	Inst	f	w_1f_1	w_2f_2	w_3f_3	w_4f_4	w_5f_5	w_6f_6	w_7f_7	w_8f_8	w_9f_9	$w_{10}f_{10}$	$w_{11}f_{11}$	Time	nR
	I37C	245.79	0	0	0	0	0	0	193.33	36.67	6.00	8.17	1.62	21.09	0
1	I38C	235.39	0	0	0	0	0	0	193.33	26.67	5.00	3.50	6.89	52.79	0
	I45C	266.85	0	0	0	0	0	0	220.00	30.00	4.33	4.83	7.68	103.24	0
	I37C	328.55	0	0	0	0	130	75	60.00	53.33	2.67	6.00	1.55	15.82	0
2	I38C	259.49	0	0	0	0	100	58.33	60.00	26.67	4.33	2.67	7.49	36.13	0
	I45C	339.37	0	0	0	0	130	58.33	86.67	46.67	5.67	4.50	7.54	96.98	0
	I37C	255.15	0	0	0	0	0	0	213.33	26.67	7.67	6.17	1.32	26.43	0
3	I38C	236.38	0	0	0	0	0	0	206.67	16.67	4.67	3.00	5.38	51.51	0
	I45C	270.31	0	0	0	0	0	0	220.00	33.33	6.00	4.50	6.47	96.29	0
	I37C	277.65	0	0	0	0	0	0	220.00	43.33	6.00	7.17	1.15	19.44	0
4	I38C	212.04	0	0	0	0	0	0	186.67	13.33	4.33	3.50	4.21	53.00	0
	I45C	253.63	0	0	0	0	0	0	220.00	16.67	5.33	5.50	6.13	73.69	0
	I37C	241.92	0	0	0	0	0	0	180.00	46.67	7.33	6.33	1.59	30.10	0
5	I38C	225.82	0	0	0	0	0	0	186.67	23.33	5.33	3.00	7.49	59.09	0
	I45C	274.13	0	0	0	0	0	0	206.67	46.67	6.67	6.00	8.13	130.74	0
	I37C	252.68	0	0	0	0	0	0	213.33	26.67	4.00	7.50	1.18	18.64	0
6	I38C	221.53	0	0	0	0	0	0	186.67	20.00	5.67	2.00	7.20	39.95	0
	I45C	275.45	0	0	0	0	0	0	220.00	36.67	5.67	5.33	7.78	76.58	0
	I37C	244.13	0	0	0	0	0	0	200.00	36.67	2.33	3.50	1.63	24.78	0
7	I38C	263.52	0	0	0	0	0	0	213.33	36.67	4.00	3.50	6.02	56.79	0
	I45C	290.44	0	0	0	0	0	0	213.33	56.67	6.67	6.33	7.44	117.49	0
	I37C	246.81	0	0	0	0	0	0	193.33	40.00	6.00	5.83	1.64	26.93	0
8	I38C	167.04	0	0	0	0	0	0	153.33	3.33	1.33	1.67	7.38	51.40	0
	I45C	265.72	0	0	0	0	0	0	213.33	36.67	4.33	4.17	7.22	82.67	0
	Average	**256.24**	**0**	**0**	**0**	**0**	**15**	**7.99**	**185.83**	**32.50**	**5.06**	**4.78**	**5.09**	**56.73**	**0**

Table 8: Results for the simulated instances of Scenarios 1-8

corresponding to reschedules that include all flights. In particular, delay's type II and the linear delay tend to decrease, while the helicopter reassignments increase.

As expected, Scenario 5 shows that the heuristic solutions were sensitive to changes in the minimum time between consecutive flights, time on the ground and the limits for delays I and II, once decreasing $d^{II}_{\max}$ and increasing t^{a} and tat make the schedule more restrictive, or the opposite makes it easier for allocation flights. Nevertheless, the heuristic approach still allocates all flights. The results of Scenario 6 show that the heuristic was still able to reschedule all flights under a few changes in the availability of some helicopters. In Scenario 7, we note that the delay's type II and linear increased with the changes in the flight destinations, mainly because of the precedence constraints between flights to the same maritime unit. Scenario 8 indicates the impact of having more transferred flights. Although transferred flights have a high weight in the objective function and it is contained in precedence constraints, they tend to be easier to allocate as they are not considered in maximum delay rule. This increases the set of feasible solutions in the present heuristic, which generates the possibility of less use of aircraft. Hence, the heuristic managed to have lower values for I38C and I45C.

5.2.3. Results for the instances of Scenario 9

The presentation of the results of Scenario 9 was divided into two parts. The first considers all instances of Table 1, while the second focuses only on instance I45C in order to detail the results of one instance. This instance was chosen because it is the largest one, representing the most complex operating situation.

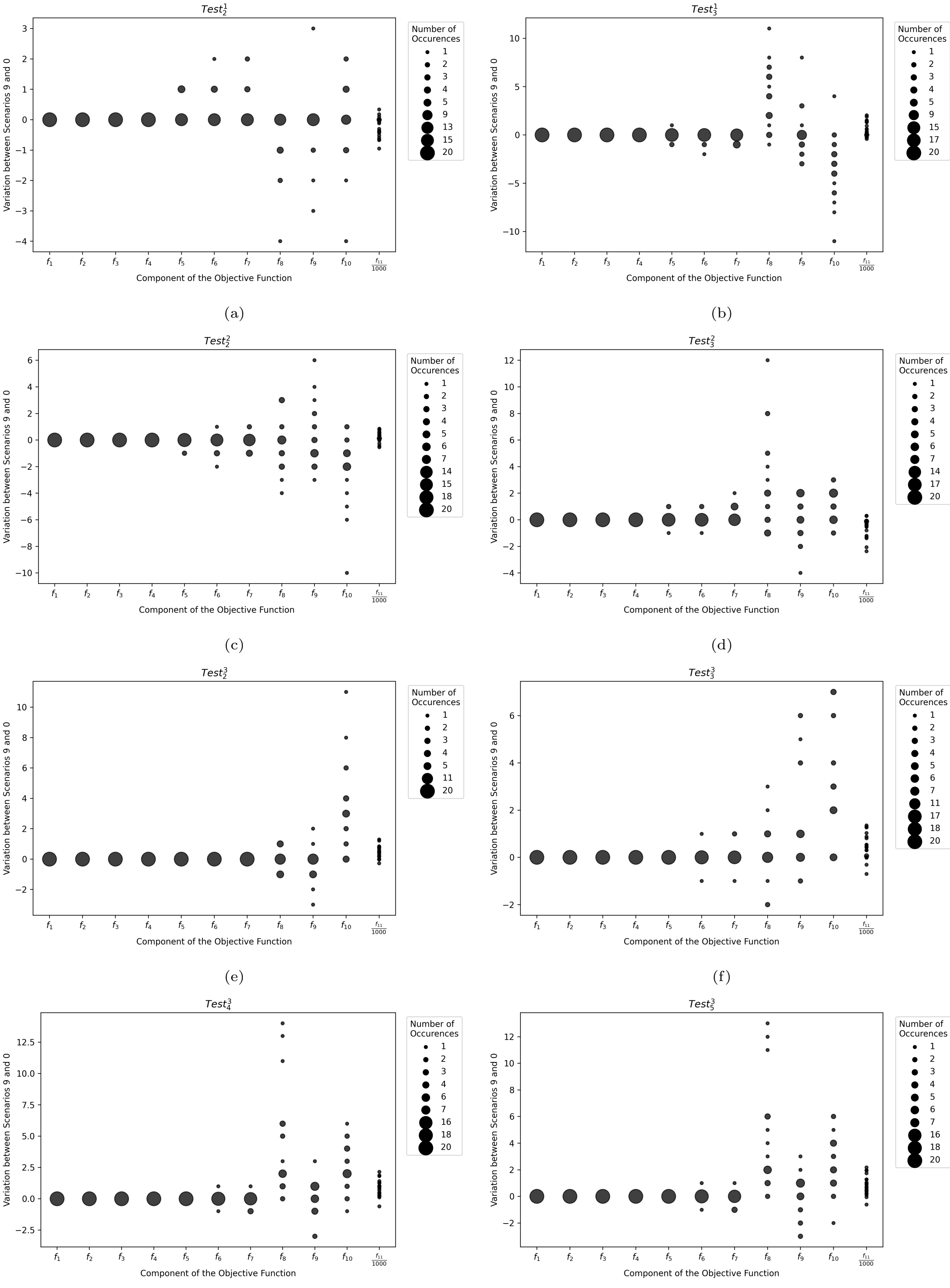
In both parts, the results are shown in the form of variation, specifically the differences of the values of the 11 objective function terms ($f_1, f_2, \ldots, f_{11}$) between Scenarios 9 and 0.

The graphs in Figures 8 depict some of the most interesting results of the first part of the experiments. They present the number of occurrences of each variation level for each objective function term. According to the results of $Test_2^1$ presented in Figure 8a, when canceling the fleet usage penalties, all flights continued to be scheduled (f_1, f_2, f_3 and f_4 in the figure), the fleet utilization (f_5, f_6 and f_7) and the change of helicopters (f_{10}) tend to increase, while the delays (f_8, f_9 and f_{11}) tend to decrease. This was expected as the increase of the fleet utilization tends to increase the change of helicopters and thus, the rate of flights performed by an aircraft is reduced. In $Test_3^1$ (Figure 8b), the best value of the objective function found by the heuristic was reached by increasing the delays, as the delay penalties are zeroed in this test. The increase of the delays allows the heuristic to increase the number of flights performed by each helicopter and reduce the total aircraft utilization.

There were no differences between the solutions of Scenarios 9 and 0, after equalizing the values for $\mathcal{F}_1$ in $Test_1^2$. This is explained as the solutions of Scenario 0 have already allocated all flights. However, leveling the values of $\mathcal{F}_2$ in $Test_2^2$ (Figure 8c) allowed to achieve reductions in delays of type II and helicopter reassignments, due to changes in relation to the use of the fleet. When equalizing the values of $\mathcal{F}_3$ in $Test_3^2$ (Figure 8d), $w_{11}f_{11}$ dominates w_8f_8 and w_9f_9 (since $D >> B^I$ and $D >> B^{II}$) and as expected, the linear delays are reduced.

In $Test_1^3$, the linear delay is not penalized and this increases f_{11}, as expected, but does not reduce significantly the other delays. Canceling the weight ranges for $Test_2^3$ (Figure 8e) increases the linear delay and the change of helicopters to reduce delays of types I and II. About $Test_3^3$ (Figure 8f), annulling the delay type I too provoked variations in the delay of type II and in the use of the fleet by increasing the parcels w_9f_9 to $w_{11}f_{11}$. By eliminating the delay and helicopters' change penalties, the heuristic maximizes the helicopter utilization. This is noted on the results of $Test_4^3$ presented in Figure 8g. $Test_5^3$ (Figure 8h) eliminates the f_7 to f_{11} terms of the objective function. This allows the heuristic to change the values on those factors without changing the remaining (non-zero) indicators. There were no changes in the solutions given by both $Test_6^3$ and $Test_7^3$ compared to the ones of $Test_5^3$.

Table 9 presents the results for the second part of Scenario 9 experiments using instance I45C. We note in $Test_2^1$ that its results are practically the same as the ones of Scenario 0. This is because the aircraft usage was already maximum for I45C in Scenario 0, thus not allowing the reduction of delays. For $Test_3^1$, canceling the weights associated with the delays implies an increase in all delays (type I, type II and linear), as expected, while the change of helicopters decreases. In $Test_1^2$, as the solution of I45C in Scenario 0 have already allocated all flights, matching the values of $\mathcal{F}_1$ does not result in any change in the I45C solution, as expected. In $Test_2^2$, because all available helicopters of I45C are of the normal fleet, matching the values of $\mathcal{F}_2$ cannot result in any change in the solution. In $Test_3^2$, since $w_{11}f_{11}$ dominates w_8f_8 and w_9f_9 in this scenario, it is expected that there will be a decrease in the linear delay and tiebreak conditions will occur in relation to the counts of the delays' types I and II. The heuristic, in this sense, behaved as expected. About $Test_1^3$, there was an increase in f_{11} and the heuristic managed to reduce the delays of types I and II. In $Test_2^3$, f_{11} and f_{10} increase. As f_8 dominates the change of helicopters, the heuristic achieves a reduction in



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8: Analysis of the main result variations between Scenarios 9 and 0

Test type	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}
$Test_2^1$	0	0	0	0	0	0	0	0	0	0	-73
$Test_3^1$	0	0	0	0	0	0	0	9	1	-9	1,046
$Test_1^2$	0	0	0	0	0	0	0	0	0	0	0
$Test_2^2$	0	0	0	0	0	0	0	0	0	0	0
$Test_3^2$	0	0	0	0	0	0	0	12	-2	2	-1,303
$Test_1^3$	0	0	0	0	0	0	0	-2	-4	1	738
$Test_2^3$	0	0	0	0	0	0	0	-1	0	6	760
$Test_3^3$	0	0	0	0	0	0	0	9	1	-9	1,046
$Test_4^3$	0	0	0	0	0	0	0	14	-4	7	-561
$Test_5^3$	0	0	0	0	0	0	0	14	-4	7	-561
$Test_6^3$	0	0	0	0	0	0	0	14	-4	7	-561
$Test_7^3$	0	0	0	0	0	0	0	14	-4	7	-561

Table 9: Result variations between Scenarios 9 and 0 for instance I45C

this regard. In $Test_3^3$, f_8 and f_{11} increased even more if compared to the previous test. For this optimization criterion, the heuristic focused on the aircraft reassignments, since the other parcels remained the same. In $Test_4^3$, as expected, there was only a change in relation to the null weight scores. It is important to highlight that there was an opposite behavior between f_8, f_{10} and f_9, f_{11}. In $Test_t^3, \forall t = 5, 6, 7$, the previous behavior was maintained in these tests. This was already expected because the solution of Scenario 0 uses all aircraft and the I45C only has a normal type of fleet.

5.3. Practical relevance of the proposed approaches

To illustrate the impact of including transferred flights from previous days into the rescheduling of table flights of the day, we plot two time-space diagrams for instance I9A in Figures 9 and 10. The diagram in Figure 9 presents the originally planned schedule of the seven table flights i_1, i_2, ..., i_7, without considering the transferred flights i_8 and i_9 (i.e., the diagram only shows the original flight timetable of the day). The optimal reschedule obtained with the two models and the heuristic for the nine flights is given in the diagram of Figure 10. The schedule of these flights uses three helicopters, named as PR-LCR, PR-LDG and PR-SHL. Similarly to Figures 1 and 2, the maritime units MU_1 to MU_5 related to these nine flights are depicted in the vertical axis of Figures 9 and 10, whereas the scheduled/rescheduled departure times of the flights are presented in the horizontal axis. Figure 10 also presents the arrival time of flight i_9 at the airport.

In Fig. 9, flights i_1, i_2 and i_3 are assigned to helicopter PR-LCR, flights i_4 and i_6 to helicopter PR-LDG and flights i_5 and i_7 to helicopter PR-SHL. Note that this schedule does not result in flight delays. The total penalty of the schedule is 65, resulting from 40 for the use of helicopters PR-LCR and PR-LCD of the normal fleet plus 25 for the use of helicopter PR-SHL of the pool fleet. The resulting reschedule to include flights i_8 and i_9, presented in Fig. 10, has a similar flight-helicopter assignment: flights i_1, i_2 and i_3 are still assigned to helicopter PR-LCR, flights i_9, i_4 and i_6 to helicopter PR-LDG and flights i_8, i_5 and i_7 to helicopter PR-SHL. However, this reschedule results in a delay of 15 minutes in flight i_1 due to the precedence order constraint between transferred flight i_9 and table flight i_1, given that both flights are related to the same maritime unit MU_1. Indeed, the rescheduling of these flights should ensure that flight i_9 lands at least 15 minutes before flight i_1 at unit MU_1. The reschedule of Fig. 10 also results in a delay of 14 minutes for flight i_4. This is because both flights i_9 and i_4 were assigned to the same helicopter PR-LDG, which needs to remain a minimum time interval on the ground of the airport (45 minutes) between any two flights. Observe

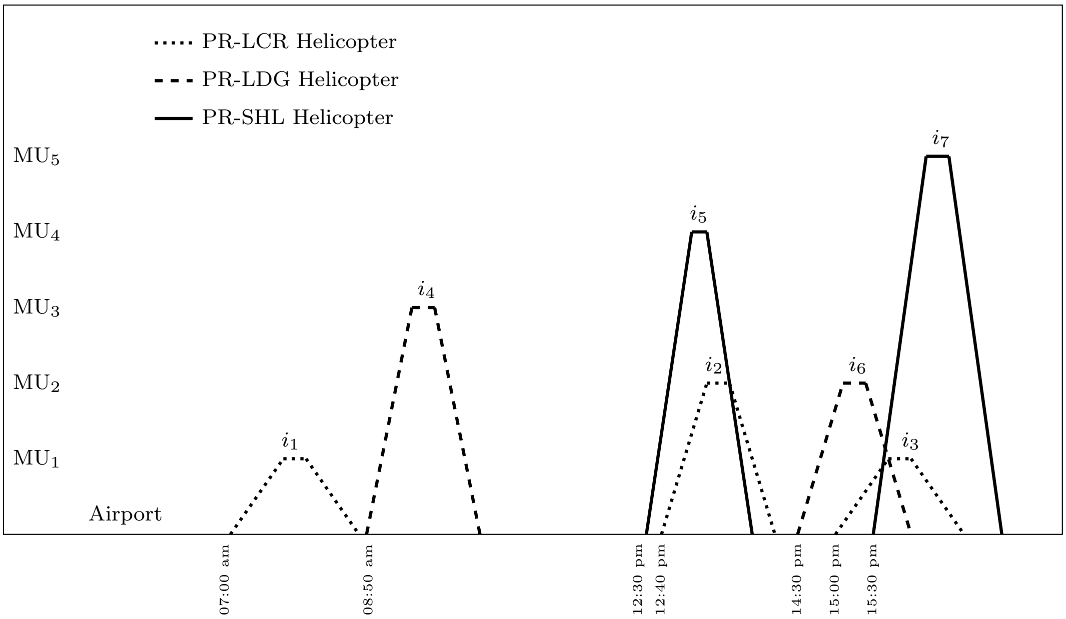


Figure 9: Time-space diagram illustrating the originally planned schedule of instance I9A without the transferred flights

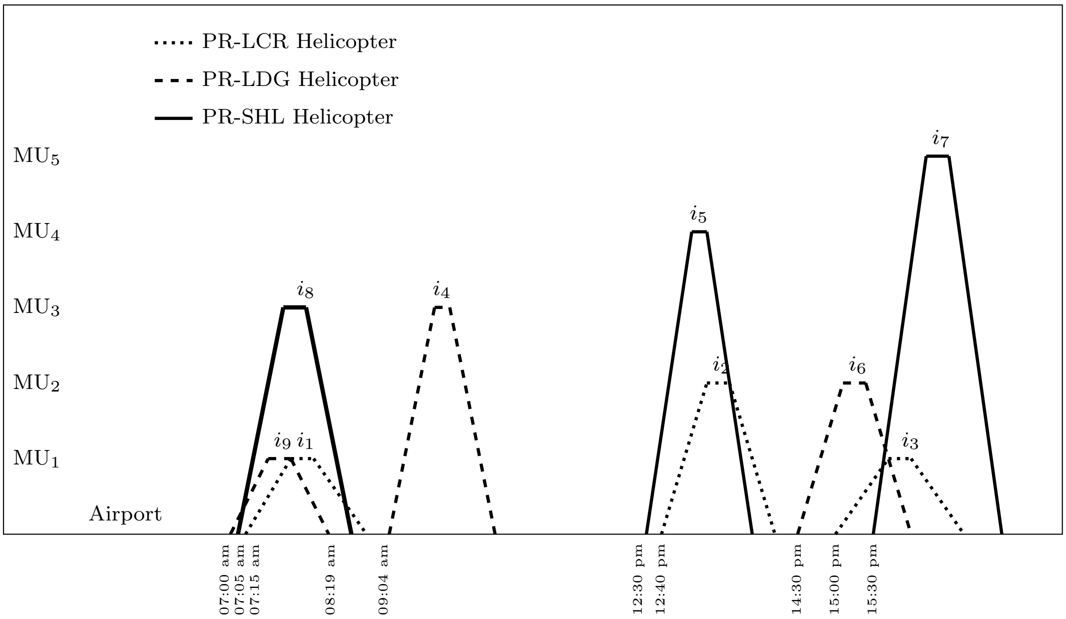


Figure 10: Time-space diagram illustrating the reschedule of instance I9A with the transferred flights

in Figure 10 that as the planned arrival time of flight i_9 at the airport is 8:19 am, the planned departure time of flight i_4 from the airport cannot be before 9:04 am. As shown in Table 4, the total penalty of this reschedule is 67.03 (40 for the use of helicopters PR-LCR and PR-LCD of the normal fleet, 25 for using helicopter PR-SHL of the pool fleet, 2 for the two type-I delays of flights i_1 and i_4, 0.034 for the total delay including 15 minutes of flight i_1, 14 minutes of flight i_4 and 5 minutes of the transferred flight i_8).

When comparing the solutions of Table 4 with the manual solutions actually carried out by the company for these problem instances, in practice, the benefits of solving the models or applying the heuristic become evident. As an example, instance I10A corresponds to a day with 8 originally scheduled table flights and 2 transferred flights from previous days. After rescheduling some table flights to recover these two transferred flights, the company operators were unable to find a feasible solution including all 10 flights - the company reschedule included only 9 flights and the last one had to be transferred to the next day. The solution of the models and the heuristic, on the other hand, provides a feasible reschedule including all the 10 flights and hence, no flights are transferred to the next day.

6. Concluding remarks

We addressed a real-life short-term flight rescheduling problem for personnel transportation to and from different maritime units in the context of the oil industry. This type of aircraft recovery problem is complex and difficult to solve in practice because of different problem characteristics that should be taken into account, in addition to several specific constraints regarding the airport, maritime units, flights and helicopters involved in the real-life situation. We present two MIP formulations, a network flow model and an event-based model, to fully represent the addressed problem, as well as an effective heuristic approach, based on constructive and local search heuristics, which found high-quality solutions for realistically sized problem instances, within acceptable computational times.

The results of computational experiments with instances created from real-life data provided by the company show the potential of this heuristic approach to produce effective daily reschedules in a few minutes, recovering all pending flights of previous days without further transferring table flights to the next day nor rescheduling them with significant delays. When comparing the solutions obtained by the approach with the manual solutions carried out by the company for some real problem instances, the practical benefits of using the approach becomes evident, as the company is primarily interested in solutions that minimize unassigned flights and are obtained within relatively short computer runtimes. For example, while the company operators were unable to find feasible reschedules including all pending flights of these instances, the reschedules found by the approach, on the other hand, include all flights and hence, no flights need to be transferred to the next day. As the runtime is less than three minutes, the approach can be used to obtain an initial solution to human analysis, which can improve the solution considering non-modelable aspects. A sensitivity analysis was also carried out with instances based on eight different scenarios, which confirmed these results and showed the robustness of proposed method.

Interesting perspectives for future research include: *i*) developing tailor-made branch-and-cut approaches for the network flow and event-based formulations with focus on solving to proven optimality realistic-sized instances in reasonable computational times; *ii*) investigating alternative MIP formulations, possibly originated from decomposition techniques, which could show stronger linear relaxation bounds and hence lead to improved exact approaches; *iii*) using the proposed construction and local search heuristics in the context of a metaheuristic approach; *iv*) extending the event-based model to include landing events and other types of events, such as mechanical failures of aircraft, absence of crew members, accommodation of disrupted passengers, adverse weather conditions, environmental protection, and particular restrictions due to coronavirus pandemic; *v*) extending the present study to address more general cases of the problem, such as flights taking-off from more than one airport, uncertainties in some time parameters of the problem (e.g., flight duration), helicopter routing and passenger allocation decisions, merging and splitting of flights, among others.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions of revision, as well the Brazilian National Petroleum Agency (ANP), São Paulo Research Foundation (FAPESP), the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES) and the Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support to carry out this research.

References

Akturk MS, Atamturk A, Gurel S. Aircraft rescheduling with cruise speed control. Operations Research 2014;62(4):829–45. doi:10.1287/opre.2014.1279.

Ali KM, Nidhal R. Novel approach for the continuous flight rescheduling problem resolution. International Journal of Advanced Trends in Computer Science and Engineering 2019;8(1):107–12. doi:10.30534/ijatcse/2019/2081.12019.

Arıkan U, Gürel S, Aktürk MS. Integrated aircraft and passenger recovery with cruise time controllability. Annals of Operations Research 2016;236(2):295–317. doi:10.1007/s10479-013-1424-2.

Bard JF, Yu G, Arguello MF. Optimizing aircraft routings in response to groundings and delays. IIE Transactions 2001;33(10):931–47. doi:10.1080/07408170108936885.

Bastos Y, Fleck J, Martinelli R. A stochastic programming approach for offshore flight scheduling. In: 15th IFAC Workshop on Discrete Event Systems. 2020. p. 1–7.

Belobaba P, Odoni A, Barnhart C. The Global Airline Industry. John Wiley & Sons, Ltd, 2015.

Brunner JO. Rescheduling of flights during ground delay programs with consideration of passenger and crew connections. Transportation Research Part E: Logistics and Transportation Review 2014;72:236–52. doi:https://doi.org/10.1016/j.tre.2014.10.004.

D'Ariano A, Pistelli M, Pacciarelli D. Aircraft retiming and rerouting in vicinity of airports. IET Intelligent Transport Systems 2012;6:433–443(10). URL: https://digital-library.theiet.org/content/journals/10.1049/iet-its.2011.0182.

Eggenberg N, Salani M, Bierlaire M. Constraint-specific recovery network for solving airline recovery problems. Computers & Operations Research 2010;37(6):1014–26. doi:10.1016/j.cor.2009.08.006.

Erkan H, Erkip NK, Özge Safak . Collaborative decision making for air traffic management: A generic mathematical program for the rescheduling problem. Computers & Industrial Engineering 2019;137:106016.

Filar JA, Manyem P, Panton DM, White K. A model for adaptive rescheduling of flights in emergencies (marfe). Journal of Industrial & Management Optimization 2007;3(2):335–56. doi:10.3934/jimo.2007.3.335.

Galvão RD, Guimarães J. The control of helicopter operations in the brazilian oil industry: Issues in the design and implementation of a computerized system. European Journal of Operational Research 1990;49(2):266–70. Operational Research in Developing Countries.

Gao M, Chi H, Xu B, Ding R. Flight rescheduling responding to large-area flight delays. Kybernetes 2012;41(10):1483–96. doi:10.1108/03684921211276693.

Gribkovskaia I, Halskau OS, Kovalyov MY. Minimizing takeoff and landing risk in helicopter pickup and delivery operations. Omega 2015;55:73 – 80. doi:https://doi.org/10.1016/j.omega.2015.02.010.

Hermeto TS, Hermeto NDSS, Hawson PN. Sao tome aircraft parking area redesign: An international infrastructure regulation study case. In: 23nd ATRS World Conference Amsterdam, The Netherlands. 2019. .

Hu Y, Liao H, Zhang S, Song Y. Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft. Expert Systems with Applications 2017;83:283–99. doi:10.1016/j.eswa.2017.04.031.

Hu Y, Song Y, Zhao K, Xu B. Integrated recovery of aircraft and passengers after airline operation disruption based on a grasp algorithm. Transportation Research Part E: Logistics and Transportation Review 2016;87:97–112. doi:10.1016/j.tre.2016.01.002.

Hu Y, Xu B, Bard JF, Chi H, Gao M. Optimization of multi-fleet aircraft routing considering passenger transiting under airline disruption. Computers Industrial Engineering 2015;80:132 –44.

Jafari N, Zegordi SH. Simultaneous recovery model for aircraft and passengers. Journal of the Franklin Institute 2011;348(7):1638–55. doi:https://doi.org/10.1016/j.jfranklin.2010.03.012.

Jozefowiez N, Mancel C, Mora-Camino F. A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions. Journal of the Operational Research Society 2013;64(3):384–95. doi:10.1057/jors.2012.20.

Kammoun MA, Rezg N. An efficient hybrid approach for resolving the aircraft routing and rescheduling problem. Journal of Air Transport Management 2018;71:73–87. doi:10.1016/j.jairtraman.2018.06.005.

Kammoun MA, Rezg N, Achour Z. New approach for air traffic management based on control theory. International Journal of Production Research 2014;52(6):1711–27. doi:10.1080/00207543.2013.848306.

Kammoun MA, Rezg N, Achour Z, Rezig S. State Space Search for Safe Time Petri Nets Based on Binary Decision Diagrams Tools: Application to Air Traffic Flow Management Problem. Studies in Informatics and Control 2016;25(1):39–50.

Liang Z, Xiao F, Qian X, Zhou L, Jin X, Lu X, Karichery S. A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. Transportation Research Part B: Methodological 2018;113:70–90. doi:10.1016/j.trb.2018.05.007.

Menezes F, Porto O, Reis ML, Moreno L, Aragão MPd, Uchoa E, Abeledo H, Nascimento NCd. Optimizing helicopter transport of oil rig crews at petrobras. INFORMS Journal on Applied Analytics 2010;40(5):408–16.

Moreno L, de Aragão MP, Uchoa E. Column generation based heuristic for a helicopter routing problem. In: Àlvarez C, Serna M, editors. Experimental Algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006. p. 219–30.

Qian F, Gribkovskaia I, Halskau OS. Helicopter routing in the norwegian oil industry: Including safety concerns for passenger transport. International Journal of Physical Distribution & Logistics Management 2011;41(4):401–15. doi:doi:10.1108/09600031111131959.

Qian F, Gribkovskaia I, Laporte G, Halskau OS. Passenger and pilot risk minimization in offshore helicopter transportation. Omega 2012;40(5):584–93. doi:https://doi.org/10.1016/j.omega.2011.11.003.

Qian F, Strusevich V, Gribkovskaia I, Halskau OS. Minimization of passenger takeoff and landing risk in offshore helicopter transportation: Models, approaches and analysis. Omega 2015;51:93–106. doi:https://doi.org/10.1016/j.omega.2014.09.002.

Rosenberger JM, Johnson EL, Nemhauser GL. Rerouting aircraft for airline recovery. Transportation Science 2003;37(4):408–21. URL: https://doi.org/10.1287/trsc.37.4.408.23271. doi:10.1287/trsc.37.4.408.23271. arXiv:https://doi.org/10.1287/trsc.37.4.408.23271.

Sinclair K, Cordeau J, Laporte G. A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. Computers & Operations Research 2016;65:42–52. doi:10.1016/j.cor.2015.06.014.

Sinclair K, Cordeau JF, Laporte G. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. European Journal of Operational Research 2014;233(1):234–45. doi:10.1016/j.ejor.2013.08.034.

Teodorovic D, Guberinic S. Optimal dispatching strategy on an airline network after a schedule perturbation. European Journal of Operational Research 1984;15(2):178–82.

Thengvall BG, Bard JF, Yu G. Balancing user preferences for aircraft schedule recovery during irregular operations. IIE Transactions 2000;32(3):181–93. doi:10.1023/A:1007618928820.

Thengvall BG, Yu G, Bard JF. Multiple fleet aircraft schedule recovery following hub closures. Transportation Research Part A: Policy and Practice 2001;35(4):289–308. doi:10.1016/S0965-8564(99)00059-2.

Zhang C. Two-Stage Heuristic Algorithm for Aircraft Recovery Problem. Discrete Dynamics in Nature and Society 2017;2017:12. doi:10.1155/2017/9575719.

Zhang D, Lau HH, Yu C. A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. Computers & Industrial Engineering 2015;87:436–53. doi:10.1016/j.cie.2015.05.033.

Zhang D, Yu C, Desai J, Lau HH. A math-heuristic algorithm for the integrated air service recovery. Transportation Research Part B: Methodological 2016;84:211–36. doi:10.1016/j.trb.2015.11.016.

Zhu B, Zhu J, Gao Q. A stochastic programming approach on aircraft recovery problem. Mathematical Problems in Engineering 2015;2015:680609. doi:10.1155/2015/680609.

Appendix A. Further details on the heuristic approach

To show in details the overall execution scheme of the proposed heuristic, we present the flowchart in Fig. A.11. The main steps of the method have been enumerated in blocks, where Block 2 initializes the main variables, such as k, $OF_0 = +\infty$, $CondRelax$ (a flag that indicates that the relaxed phase is active), $QF = 0$, and $ST = R$ (they are parameters to be used in the repairing stage); Blocks 4-9 refer to the cycle of the relaxed construction stage (Tr_k is a vector that stores the aircraft type on the current iteration k); Block 10 shows how the repairing stage works specifically; and Block 11 represents the call for the improvement phase.

Moreover, the six procedures of the improvement phase described in Subsection 4.2 are detailed in Algorithms 3 to 8. In the following section, we introduce the additional notation used in these algorithms. Let $altIH_{i,h}^{k}$ be the alternative regarding the insertion of flight i in aircraft h of the iteration k, which was stored in the construction phase, as mentioned in Subsection 4.1. Consider $\mathcal{HP}$ as the set containing only the aircraft allocated in the current schedule, and W_l as the lth weight term of the objective function, this is, $W_l = w_l f_l$, $\forall l = 1, \ldots, 11$. Finally, we define $findIndex($"$flight\_indice''$, "$aircraft\_indice''$, "o : $origin$/d : $destiny''$) as a function that returns the index of the corresponding precedent (parameter 'o') or subsequent (parameter 'd') flight from another flight (parameter "$flight\_indice$") allocated on the same aircraft (parameter "$aircraft\_indice$"). As an example, let $Y_{3,7}^{2,5} = 1$ be one of the components of the solution in iteration $k = 2$; then, $findIndex(7, 2, \text{'o'}) = 3$ and $findIndex(3, 2, \text{'d'}) = 7$. On the other hand, if $Y_{3,7}^{2,5} = 0$, this function returns -1. It is important to mention that every time a given set Pr_h is modified in the algorithms, set PR^k is updated accordingly.

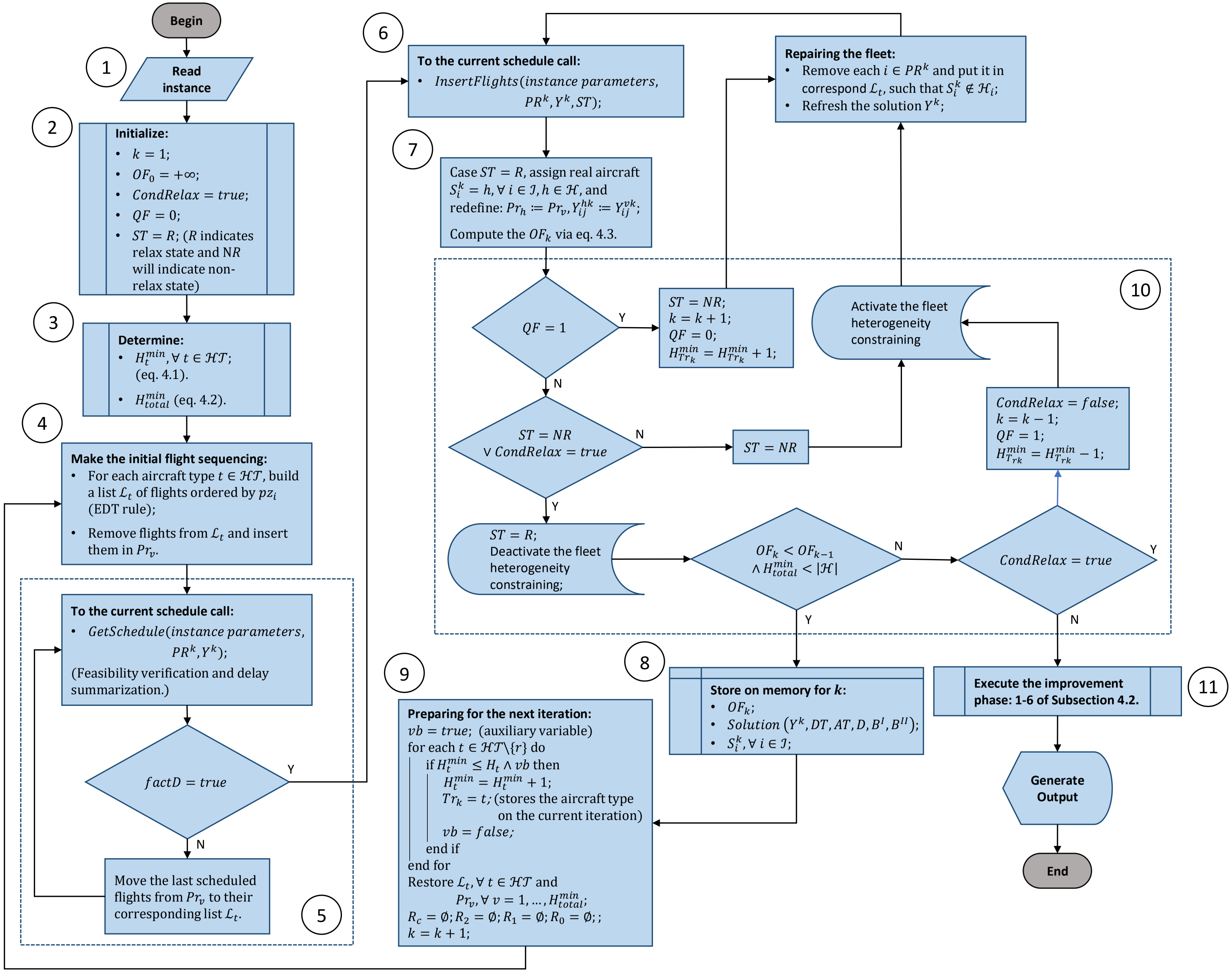


Figure A.11: Flowchart of the heuristic approach

Algorithm 3: *Reschedule previously scheduled flights to accommodate transferred flights*

Input: instance parameters, PR^k, current solution Y^k, OF_k.
Output: PR^k, Y^k, OF_k.

Let $b, TD, i', j', b', h', i^*, j^*, b^*, h^*, v0, v1, v2, v3$, be auxiliary variables;
if $|PR^k| < |\mathcal{I}|$ **then**
  foreach $i \in \mathcal{I} \backslash PR^k$ **do** // for transferred flights
    $TD \leftarrow +\infty; i^* \leftarrow 0$;
    foreach $i' \in PR^k$ **do**
      foreach $h \in \mathcal{HP} \cap \mathcal{H}_i$ **do**
        if $h = S^k_{i'} \wedge i \neq i' \wedge altIH^k_{i,h} = 1$ **then**
          $j \leftarrow findIndex(i', h, \text{‘o’})$;
          $b \leftarrow findIndex(i', h, \text{‘d’})$;
          $Y^{h,k}_{j,i'} \leftarrow 0; Y^{h,k}_{i',b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i'\}$;
          $Y^{h,k}_{j,i} \leftarrow 1; Y^{h,k}_{i,b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i\}$; // does the temporary change
          foreach $h' \in \mathcal{HP} \cap \mathcal{H}_{i'}$ **do**
            if $h \neq h' \wedge altIH^k_{i',h'} = 1$ **then**
              $b' \leftarrow |\mathcal{I}|+1$;
              $j' \leftarrow findIndex(b', h', \text{‘o’})$;
              while $j' \geq 0$ **do**
                $Y^{h',k}_{j',b'} \leftarrow 0$; $Y^{h',k}_{j',i'} \leftarrow 1$; $Y^{h',k}_{i',b'} \leftarrow 1$; $Pr_{h'} \leftarrow Pr_{h'} \cup \{i'\}$;
                $GetSchedule(instance\ parameters, PR^k, Y^k)$;
                if $\sum_{l \in \mathcal{I}} D_l < TD \wedge factD = true$ **then**
                  $TD \leftarrow \sum_{l \in \mathcal{I}} D_l$;
                  $j^* \leftarrow j; i^* \leftarrow i$;
                  $b^* \leftarrow b; h^* \leftarrow h$;
                  $v1 \leftarrow j'; v2 \leftarrow i'$;
                  $v3 \leftarrow b'; v0 \leftarrow h'$;
                $Y^{h',k}_{j',b'} \leftarrow 1$; $Y^{h',k}_{j',i'} \leftarrow 0$; $Y^{h',k}_{i',b'} \leftarrow 0$; $Pr_{h'} \leftarrow Pr_{h'} \backslash \{i'\}$;
                $b' \leftarrow j'$;
                $j' \leftarrow findIndex(b', h', \text{‘o’})$;
          $Y^{h,k}_{j,i'} \leftarrow 1; Y^{h,k}_{i',b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i'\}$;
          $Y^{h,k}_{j,i} \leftarrow 0; Y^{h,k}_{i,b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i\}$; // undoes the temporary change
    if $i^* > 0$ **then** // determines the exchange
      $Y^{h^*,k}_{j^*,v2} \leftarrow 0; Y^{h^*,k}_{v2,b^*} \leftarrow 0; Pr_{h^*} \leftarrow Pr_{h^*} \backslash \{v2\}$;
      $Y^{h^*,k}_{j^*,i^*} \leftarrow 1; Y^{h^*,k}_{i^*,b^*} \leftarrow 1; Pr_{h^*} \leftarrow Pr_{h^*} \cup \{i^*\}$;
      $Y^{v0,k}_{v1,v3} \leftarrow 0; Y^{v0,k}_{v1,v2} \leftarrow 1; Y^{v0,k}_{v2,v3} \leftarrow 1; Pr_{v0} \leftarrow Pr_{v0} \cup \{v2\}$;
      Remove the flight i^* of corresponding $\mathcal{R}$ set;
      $S^k_{i^*} \leftarrow h^*; S^k_{v2} \leftarrow v0$;
      $altIH^k_{i^*,h^*} \leftarrow 0; altIH^k_{v2,h^*} \leftarrow 1; altIH^k_{v2,v0} \leftarrow 0$;
Update all the terms W and OF_k;

Algorithm 4: *Swap unscheduled by scheduled flights*

Input: instance parameters, PR^k, current solution Y^k, OF_k.
Output: PR^k, Y^k, OF_k.

Let $b, FV1, FV2, i', i^*, b^*, v1, v2$, be auxiliary variables;
if $|PR^k| < |\mathcal{I}|$ **then**
  foreach $h \in \mathcal{HP}$ **do**
    foreach $i \in Pr_h$ **do**
      $FV1 \leftarrow OF_k; j \leftarrow findIndex(i, h, \text{'o'}); b \leftarrow findIndex(i, h, \text{'d'})$;
      $Y^{h,k}_{j,i} \leftarrow 0; Y^{h,k}_{i,b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i\}$;
      select i
        case $\in \mathcal{I}_c : \mathcal{R}_C \leftarrow \mathcal{R}_C \cup \{i\}$;
        case $\in \mathcal{I}_2 : \mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{i\}$;
        case $\in \mathcal{I}_1 : \mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup \{i\}$;
        case $\in \mathcal{I}_0 : \mathcal{R}_0 \leftarrow \mathcal{R}_0 \cup \{i\}$;
      if $s_i \neq h \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v1 \leftarrow -1$; // if there was an aircraft change
      else $v1 \leftarrow 0$;
      $i^* \leftarrow 0$;
      foreach $i' \in \mathcal{I} \backslash PR^k$ **do**
        if $h \in \mathcal{H}_{i'} \wedge altIH^k_{i',h} = 1$ **then**
          $Y^{h,k}_{j,i'} \leftarrow 1; Y^{h,k}_{i',b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i'\}$;
          select i'
            case $\in \mathcal{I}_c : \mathcal{R}_C \leftarrow \mathcal{R}_C \backslash \{i'\}$;
            case $\in \mathcal{I}_2 : \mathcal{R}_2 \leftarrow \mathcal{R}_2 \backslash \{i'\}$;
            case $\in \mathcal{I}_1 : \mathcal{R}_1 \leftarrow \mathcal{R}_1 \backslash \{i'\}$;
            case $\in \mathcal{I}_0 : \mathcal{R}_0 \leftarrow \mathcal{R}_0 \backslash \{i'\}$;
          $GetSchedule(instance\ parameters, PR^k, Y^k)$;
          if $s_{i'} \neq h \wedge i' \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v2 \leftarrow 1$; // if there will be an aircraft change
          else $v2 \leftarrow 0$;
          $FV2 \leftarrow w1.|\mathcal{R}_C| + w2.|\mathcal{R}_2| + w3.|\mathcal{R}_1| + w4.|\mathcal{R}_0| + W5 + W6 + W7$
          $+ w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + [W10 + w10.(v1 + v2)] + w11. \sum_{l \in \mathcal{I}} D_l$;
          if $FV1 > FV2 \wedge factD = true$ **then**
            $FV1 \leftarrow FV2$;
            $i^* \leftarrow i'; b^* \leftarrow (v1 + v2)$;
          $Y^{h,k}_{j,i'} \leftarrow 0; Y^{h,k}_{i',b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i'\}$;
          select i'
            case $\in \mathcal{I}_c : \mathcal{R}_C \leftarrow \mathcal{R}_C \cup \{i'\}$;
            case $\in \mathcal{I}_2 : \mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{i'\}$;
            case $\in \mathcal{I}_1 : \mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup \{i'\}$;
            case $\in \mathcal{I}_0 : \mathcal{R}_0 \leftarrow \mathcal{R}_0 \cup \{i'\}$;
      if $i^* > 0$ **then**
        $Y^{h,k}_{j,i^*} \leftarrow 1; Y^{h,k}_{i^*,b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i^*\}$;
        select i^*
          case $\in \mathcal{I}_c : \mathcal{R}_C \leftarrow \mathcal{R}_C \backslash \{i^*\}$;
          case $\in \mathcal{I}_2 : \mathcal{R}_2 \leftarrow \mathcal{R}_2 \backslash \{i^*\}$;
          case $\in \mathcal{I}_1 : \mathcal{R}_1 \leftarrow \mathcal{R}_1 \backslash \{i^*\}$;
          case $\in \mathcal{I}_0 : \mathcal{R}_0 \leftarrow \mathcal{R}_0 \backslash \{i^*\}$;
        $GetSchedule(instance\ parameters, PR^k, Y^k)$;
        $W10 \leftarrow W10 + w10.b^*$;
        $OF_k \leftarrow w1.|\mathcal{R}_C| + w2.|\mathcal{R}_2| + w3.|\mathcal{R}_1| + w4.|\mathcal{R}_0| + W5 + W6 + W7$
        $+ w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + W10 + w11. \sum_{l \in \mathcal{I}} D_l$;
        $S^k_{i^*} \leftarrow S^k_i; S^k_i \leftarrow -1$; // where "-1" corresponds no aircraft allocated
        $altIH^k_{i^*,h} \leftarrow 0; altIH^k_{i,h} \leftarrow 1$;
      else
        $Y^{h,k}_{j,i} \leftarrow 1; Y^{h,k}_{i,b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i\}$;
        select i
          case $\in \mathcal{I}_c : \mathcal{R}_C \leftarrow \mathcal{R}_C \backslash \{i\}$;
          case $\in \mathcal{I}_2 : \mathcal{R}_2 \leftarrow \mathcal{R}_2 \backslash \{i\}$;
          case $\in \mathcal{I}_1 : \mathcal{R}_1 \leftarrow \mathcal{R}_1 \backslash \{i\}$;
          case $\in \mathcal{I}_0 : \mathcal{R}_0 \leftarrow \mathcal{R}_0 \backslash \{i\}$;
  Update the terms $W8, W9, W11$;

Algorithm 5: *Transfer flights to other aircraft*

Input: instance parameters, PR^k, current solution Y^k, OF_k.
Output: PR^k, Y^k, OF_k.

Let $b, FV1, FV2, j', b', h', i^*, j^*, b^*, h^*, v0, v1, v2$, be auxiliary variables;
Let *continue* be a variable that gives “true” if there is benefits to change flights, or “false”, otherwise;
do
  Set $continue \leftarrow false$;
  $FV1 \leftarrow W8 + W9 + W10 + W11$;
  foreach $h \in \mathcal{HP}$ **do**
    foreach $i \in Pr_h$ **do**
      $j \leftarrow findIndex(i, h, \text{‘o’}); b \leftarrow findIndex(i, h, \text{‘d’})$;
      $Y^{h,k}_{j,i} \leftarrow 0; Y^{h,k}_{i,b} \leftarrow 0; Y^{h,k}_{j,b} \leftarrow 1$;
      if $s_i \neq h \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v1 \leftarrow -1$; // if there was an aircraft change
      else $v1 \leftarrow 0$;
      foreach $h' \in \mathcal{HP} \cap \mathcal{H}_i$ **do**
        if $h' \neq h \wedge altIH^k_{i,h'} = 1$ **then**
          $b' \leftarrow |\mathcal{I}|+1$;
          $j' \leftarrow findIndex(b', h', \text{‘o’})$;
          while $j' \geq 0$ **do**
            $Y^{h',k}_{j',i} \leftarrow 1; Y^{h',k}_{i,b'} \leftarrow 1;\ Y^{h',k}_{j',b'} \leftarrow 0$;
            $GetSchedule(instance\ parameters, PR^k, Y^k)$;
            if $s_i \neq h' \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v2 \leftarrow 1$; // if there will be an aircraft change
            else $v2 \leftarrow 0$;
            $FV2 \leftarrow w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + [W10 + w10.(v1 + v2)] + w11. \sum_{l \in \mathcal{I}} D_l$;
            if $FV2 < FV1 \wedge factD = true$ **then**
              $FV1 \leftarrow FV2$;
              $i^* \leftarrow i; v0 \leftarrow h; j^* \leftarrow j'$;
              $b^* \leftarrow b'; h^* \leftarrow h'$;
              $continue \leftarrow true$;
            $Y^{h',k}_{j',i} \leftarrow 0; Y^{h',k}_{i,b'} \leftarrow 0;\ Y^{h',k}_{j',b'} \leftarrow 1$;
            $b' \leftarrow j'$;
            $j' \leftarrow findIndex(b', h', \text{‘o’})$;
      $Y^{h,k}_{j,i} \leftarrow 1; Y^{h,k}_{i,b} \leftarrow 1; Y^{h,k}_{j,b} \leftarrow 0$;
  if $continue = true$ **then**
    $j \leftarrow findIndex(i^*, v0, \text{‘o’}); k \leftarrow findIndex(i^*, v0, \text{‘d’})$;
    $Y^{v0,k}_{j,i^*} \leftarrow 0; Y^{v0,k}_{i^*,b} \leftarrow 0; Y^{v0,k}_{j,b} \leftarrow 1; Pr_{v0} \leftarrow Pr_{v0} \backslash \{i^*\}$;
    $Y^{h^*,k}_{j^*,i^*} \leftarrow 1; Y^{h^*,k}_{i^*,b^*} \leftarrow 1; Y^{h^*,k}_{j^*,b^*} \leftarrow 0; Pr_{h^*} \leftarrow Pr_{h^*} \cup \{i^*\}$;
    $GetSchedule(instance\ parameters, PR^k, Y^k)$;
    if $s_{i^*} \neq v0 \wedge i^* \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v1 \leftarrow -1$;
    else $v1 \leftarrow 0$;
    if $s_{i^*} \neq h^* \wedge i^* \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $v2 \leftarrow 1$;
    else $v2 \leftarrow 0$;
    $OF_k \leftarrow OF_k + \left[w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + w10.(v1 + v2) + w11. \sum_{l \in \mathcal{I}} D_l\right]$
    $- (W8 + W9 + W10 + W11)$;
    Update the terms $W8, W9, W10, W11$;
    $S^k_{i^*} \leftarrow h^*$;
    $altIH^k_{i^*,h^*} \leftarrow 0; altIH^k_{i^*,v0} \leftarrow 1$;
while $continue = true$;

Algorithm 6: *Inter-aircraft flight swapping*

Input: instance parameters, PR^k, current solution Y^k, OF_k.

Output: PR^k, Y^k, OF_k.

Let $b, FV1, FV2, dif, i', j', b', h', i^*, j^*, b^*, h^*, v0, v1, v2, v3$, be auxiliary variables;
Let *continue* be a variable that gives "true" if there is benefits to change flights, or "false", otherwise;
do
Set $continue \leftarrow false$;
$FV1 \leftarrow W8 + W9 + W10 + W11$;
foreach $i \in PR^k$ **do**
foreach $h \in \mathcal{HP}$ **do**
if $altIH^k_{i,h} = 1$ **then**
$v0 \leftarrow S^k_i; j \leftarrow findIndex(i, v0, \text{'o'}); b \leftarrow findIndex(i, v0, \text{'d'})$;
foreach $v2 \in Pr_h$ **do**
if $v2 \neq i \wedge h \in \mathcal{H}_i \wedge v0 \in \mathcal{H}_{v2}$ **then**
$Y^{v0,k}_{j,i} \leftarrow 0; Y^{v0,k}_{i,b} \leftarrow 0; Pr_{v0} \leftarrow Pr_{v0} \backslash \{i\}$;
$Y^{h,k}_{v1,v2} \leftarrow 0; Y^{h,k}_{v2,v3} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{v2\}$;
$Y^{v0,k}_{j,v2} \leftarrow 1; Y^{v0,k}_{v2,b} \leftarrow 1; Pr_{v0} \leftarrow Pr_{v0} \cup \{v2\}$;
$Y^{h,k}_{v1,i} \leftarrow 1; Y^{h,k}_{i,v3} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i\}$;
$GetSchedule(instance\ parameters, PR^k, Y^k)$;
$dif \leftarrow 0$;
if $s_i \neq v0 \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif - 1$;
if $s_{v2} \neq h \wedge v2 \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif - 1$;
if $s_i \neq h \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif + 1$;
if $s_{v2} \neq v0 \wedge v2 \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif + 1$;
$FV2 \leftarrow w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + (W10 + w10.dif) + w11. \sum_{l \in \mathcal{I}} D_l$;
if $FV2 < FV1 \wedge factD = true$ **then**
$FV1 \leftarrow FV2$;
$i^* \leftarrow v2; j^* \leftarrow v1; b^* \leftarrow v3; h^* \leftarrow h$;
$i' \leftarrow i; j' \leftarrow j; b' \leftarrow b; h' \leftarrow v0$;
$continue \leftarrow true$;
$Y^{v0,k}_{j,i} \leftarrow 1; Y^{v0,k}_{i,b} \leftarrow 1; Pr_{v0} \leftarrow Pr_{v0} \cup \{i\}$;
$Y^{h,k}_{v1,v2} \leftarrow 1; Y^{h,k}_{v2,v3} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{v2\}$;
$Y^{v0,k}_{j,v2} \leftarrow 0; Y^{v0,k}_{v2,b} \leftarrow 0; Pr_{v0} \leftarrow Pr_{v0} \backslash \{v2\}$;
$Y^{h,k}_{v1,i} \leftarrow 0; Y^{h,k}_{i,v3} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i\}$;
if $continuar = true$ **then**
$Y^{h',k}_{j',i'} \leftarrow 0; Y^{h',k}_{i',b'} \leftarrow 0; Pr_{h'} \leftarrow Pr_{h'} \backslash \{i'\}$;
$Y^{h^*,k}_{j^*,i^*} \leftarrow 0; Y^{h^*,k}_{i^*,b^*} \leftarrow 0; Pr_{h^*} \leftarrow Pr_{h^*} \backslash \{i^*\}$;
$Y^{h',k}_{j',i^*} \leftarrow 1; Y^{h',k}_{i^*,b'} \leftarrow 1; Pr_{h'} \leftarrow Pr_{h'} \cup \{i^*\}$;
$Y^{h^*,k}_{j^*,i'} \leftarrow 1; Y^{h^*,k}_{i',b^*} \leftarrow 1; Pr_{h^*} \leftarrow Pr_{h^*} \cup \{i'\}$;
$GetSchedule(instance\ parameters, PR^k, Y^k)$;
$dif \leftarrow 0$;
if $s_{i'} \neq h' \wedge i' \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif - 1$;
if $s_{i^*} \neq h^* \wedge i^* \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif - 1$;
if $s_{i'} \neq h^* \wedge i' \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif + 1$;
if $s_{i^*} \neq h' \wedge i^* \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $dif \leftarrow dif + 1$;
$OF_k \leftarrow OF_k + (w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + w10.dif + w11. \sum_{l \in \mathcal{I}} D_l) - (W8 + W9 + W10 + W11)$;
Update the terms $W8, W9, W10, W11$;
$v0 \leftarrow S^k_{i'}; S^k_{i'} \leftarrow S^k_{i^*}; S^k_{i^*} \leftarrow v0$;
$altIH^k_{i^*,h'} \leftarrow 0; altIH^k_{i^*,h^*} \leftarrow 1$;
$altIH^k_{i',h^*} \leftarrow 0; altIH^k_{i',h'} \leftarrow 1$;
while $continue = true$;

Algorithm 7: *Intra-aircraft flight swapping*

Input: instance parameters, PR^k, current solution Y^k, OF_k.

Output: Y^k, OF_k.

Let $FV1, FV2, v1, v2$ be auxiliary variables, and $\overline{Y}$ be an auxiliary solution;
foreach $h \in \mathcal{HP}$ **do**
    $v1 \leftarrow 0$;
    while $v1 < |\mathcal{I}|$ **do**
        for $i' \leftarrow 0$ **to** $|\mathcal{I}|+2$, $step + 1$ **do**
            for $j' \leftarrow 0$ **to** $|\mathcal{I}|+2$, $step + 1$ **do**
                foreach $h' \in \mathcal{H}$ **do**
                    $\overline{Y}^{h'}_{i',j'} \leftarrow Y^{h',k}_{i',j'}$;
        $FV1 \leftarrow W8 + W9 + W11$;
        $v2 \leftarrow v1$;
        $i \leftarrow findIndex(v1, h, \text{'d'}); j \leftarrow findIndex(i, h, \text{'d'})$;
        while $0 < j < |\mathcal{I}|$ **do**
            $\overline{Y}^h_{v1,i} \leftarrow 0; \overline{Y}^h_{i,j} \leftarrow 0; \overline{Y}^h_{v1,j} \leftarrow 1$;
            $v1 \leftarrow j$;
            $j \leftarrow findIndex(v1, h, \text{'d'})$;
            $\overline{Y}^h_{v1,i} \leftarrow 1; \overline{Y}^h_{i,j} \leftarrow 1; \overline{Y}^h_{v1,j} \leftarrow 0$;
            $GetSchedule(instance\ parameters, PR^k, Y^k)$;
            $FV2 \leftarrow w8. \sum_{l \in \mathcal{I}} B^{II}_l + w9. \sum_{l \in \mathcal{I}} B^I_l + w11. \sum_{l \in \mathcal{I}} D_l$;
            if $FV2 < FV1 \wedge factD = true$ **then**
                $OF_k \leftarrow OF_k + (FV2 - FV1)$;
                $FV1 \leftarrow FV2$;
                for $i^* \leftarrow 0$ **to** $|\mathcal{I}|+2$, $step + 1$ **do**
                    for $j^* \leftarrow 0$ **to** $|\mathcal{I}|+2$, $step + 1$ **do**
                        foreach $h^* \in \mathcal{H}$ **do**
                            $Y^{h^*,k}_{i^*,j^*} \leftarrow \overline{Y}^{h^*}_{i^*,j^*}$;
                Update the terms $W8, W9, W11$;
        $v1 \leftarrow findIndex(v2, h, \text{'d'})$;

Algorithm 8: *Reduce delay types*

Input: instance parameters, PR^k, Y^k, OF_k.
Output: $DT, AT, D, B^I, B^{II}, OF_k, factD, improve$.

Let $improve$ and $factD$ be variables related to the feasible condition improvement of the schedule;
Let $FV1, FV2$ and $\overline{DT}_i, \overline{AT}_i, \overline{D}_i, \overline{B}_i^I, \overline{B}_i^{II}$, $\forall\, i \in \mathcal{I}$, be auxiliary variables;
Let $p_i \in \mathcal{P}$ be the destination of flight i, $\forall\, i \in \mathcal{I}$;
Set $improve \leftarrow false$;
foreach $i \in \mathcal{I}$ **do**
    $\overline{DT}_i \leftarrow 0; \overline{AT}_i \leftarrow 0; \overline{D}_i \leftarrow 0; \overline{B}_i^I \leftarrow 0; \overline{B}_i^{II} \leftarrow 0;$
    if $i \in PR^k$ **then**
        $\overline{DT}_i \leftarrow DT_i;$
        $\overline{AT}_i \leftarrow AT_i;$
Let $\mathcal{O}$ be an ordered list of all flights $i \in \mathcal{I} \mid D_i > 0$ sorted in non-descending order of delay D_i;
foreach $j \in \mathcal{O}$ **do** // simulates a zero delay for flight j
    $\overline{DT}_j \leftarrow r_j;$
    $\overline{AT}_j \leftarrow r_j + tf_j;$
    Set $factD \leftarrow true$;
    foreach $i \in \mathcal{I}$ **do** // if necessary, delay other flights instead of flight j
        if $(i,j) \in PR^k \wedge \overline{DT}_i < DT_j \wedge i \neq j$ **then**
            if $p_i = p_j \wedge \overline{DT}_j - \overline{DT}_i < t_j^u$ **then**
                $\overline{DT}_i \leftarrow r_j + t_j^{\mathrm{u}};$
            if $\overline{DT}_j - \overline{DT}_i < t^a$ **then**
                $\overline{DT}_i \leftarrow r_j + t^{\mathrm{a}};$
            if $\overline{AT}_i \neq \overline{DT}_i + tf_i$ **then**
                $\overline{AT}_i \leftarrow \overline{DT}_i + tf_i;$
    foreach $j \in \mathcal{I}$ **do** // checking the feasibility
        foreach $i \in \mathcal{I}$ **do**
            if $(j,i) \in PR^k \wedge \overline{DT}_j \geq \overline{DT}_i \wedge j \neq i$ **then**
                if $p_j = p_i \wedge \overline{DT}_j - \overline{DT}_i < t_i^u$ **then** $factD \leftarrow false$;
                if $\overline{DT}_j - \overline{DT}_i < t^a$ **then** $factD \leftarrow false$;
                if $\overline{AT}_j \neq \overline{DT}_j + tf_j$ **then** $factD \leftarrow false$;
                foreach $h \in \mathcal{H}$ **do**
                    if $Y_{i,j}^{h,k} = 1 \wedge \overline{DT}_j - \overline{AT}_i < tat$ **then** $factD \leftarrow false$;
    foreach $i \in PR^k$ **do**
        $\overline{D}_i \leftarrow \overline{DT}_i - r_i;$
        if $0 < \overline{D}_i \leq d_{max}^I \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $\overline{B}_i^I \leftarrow 1;$
        if $d_{max}^I < \overline{D}_i \leq d_{max}^{II} \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $\overline{B}_i^{II} \leftarrow 1;$
        if $(\overline{D}_i > d_{max}^{II} \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c) \vee \overline{AT}_i > tw^b$ **then** $factD \leftarrow false$;
        foreach $j \in PR^k$ **do**
            if $\overline{DT}_j > \overline{DT}_i \wedge p_j = p_i \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_c \wedge j \in \mathcal{I}_1 \cup \mathcal{I}_2)$ **then**
                $factD \leftarrow false;$
    $FV1 \leftarrow w8.\sum_{i\in\mathcal{I}} B_i^{II} + w9.\sum_{i\in\mathcal{I}} B_i^I + w11.\sum_{i\in\mathcal{I}} D_i;$
    $FV2 \leftarrow w8.\sum_{i\in\mathcal{I}} \overline{B}_i^{II} + w9.\sum_{i\in\mathcal{I}} \overline{B}_i^I + w11.\sum_{i\in\mathcal{I}} \overline{D}_i;$
    if $FV2 < FV1 \wedge factD = true$ **then**
        foreach $i \in PR^k$ **do**
            $DT_i \leftarrow \overline{DT}_i, AT_i \leftarrow \overline{AT}_i;$
            $D_i \leftarrow \overline{D}_i, B_i^I \leftarrow \overline{B}_i^I, B_i^{II} \leftarrow \overline{B}_i^{II};$
        $improve \leftarrow true;$
    else
        foreach $i \in \mathcal{I}$ **do**
            $\overline{DT}_i \leftarrow DT_i; \overline{AT}_i \leftarrow AT_i;$
            $\overline{D}_i \leftarrow 0, \overline{B}_i^I \leftarrow 0, \overline{B}_i^{II} \leftarrow 0;$
Update OF_k;