Federal University of São Carlos – UFSCar
Exact and Technology Sciences Center – CCET
Graduate Program in Production Engineering – PPGEP

Thiago José dos Santos Vieira

Optimization of aircraft routing with crew pairing for non-scheduled air transportation

Doctoral dissertation

Thiago José dos Santos Vieira

Optimization of aircraft routing with crew pairing for non-scheduled air transportation[1]

Doctoral dissertation submitted to Graduate Program in Production Engineering/UFSCar, as a partial requirement to obtain the degree of Doctor in Production Engineering.

Concentration area: Operations Research.
Supervisor: Prof. Dr. Pedro Augusto Munari Júnior.
Co-supervisor: Prof. Dr. Reinaldo Morabito Neto.

São Carlos
April, 2024

[1] This research was supported by the São Paulo Research Foundation (FAPESP), grant number 2020/11602-5, and the Coordination for the Improvement of Higher Education Personnel (CAPES), finance code 001.

Vieira, Thiago
Optimization of aircraft routing with crew pairing for non-scheduled air transportation / Thiago José dos Santos Vieira – São Carlos, April, 2024.
290f.: il.; 30 cm

Doctoral dissertation – Federal University of São Carlos (UFSCar)
Supervisor: Prof. Dr. Pedro Augusto Munari Júnior / Co-supervisor: Prof. Dr. Reinaldo Morabito Neto
Examination committee members: Prof. Dr. Alysson Machado Costa; Prof. Dr. Franklina Maria Bragion de Toledo; Prof. Dr. Laio Oriel Seman; Prof. Dr. Roberto Fernandes Tavares Neto

1. Non-scheduled air transportation. 2. Dial-a-flight problem. 3. Aircraft recovery problem.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Defesa de Tese de Doutorado do candidato Thiago José dos Santos Vieira, realizada em 30/04/2024.

Comissão Julgadora:

Prof. Dr. Pedro Augusto Munari Junior (UFSCar)

Prof. Dr. Reinaldo Morabito Neto (UFSCar)

Prof. Dr. Alysson Machado Costa (University of Melbourne)

Profa. Dra. Franklina Maria Bragion de Toledo (USP)

Prof. Dr. Laio Oriel Seman (UFSC)

Prof. Dr. Roberto Fernandes Tavares Neto (UFSCar)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia de Produção.

Dedicated to my lovely grandmothers,
Aurora and Edith.

Acknowledgements

First and foremost, I thank God for giving me strength and perseverance in all the steps necessary to conclude this doctorate, and at every moment of my life.

I am thankful to my parents, Ademir and Clarice, for their love, encouragement and trust. To my brother, Lucas, for his affection, company and complicity.

Special thanks to my supervisors, Prof. Pedro Munari and Prof. Reinaldo Morabito, for their friendship, guidance, advice, attention, patience and support over these years. They are examples of professionals and people.

I am grateful to the examination committee members, Prof. Alysson Costa, Prof. Franklina de Toledo, Prof. Laio Seman and Prof. Roberto Tavares, for their valuable contributions to this dissertation.

A lot of thanks to all my friends and colleagues from UFSCar, USP and also to those I met at conferences, who made all the years we have spent together much happier and easier.

I would also like to acknowledge the two companies involved in this research, for providing their historical data and for consolidating my doctoral theme.

I thank very much the Brazilian Agencies, FAPESP and CAPES, for the financial support during this time.

Finally, I wish to thank you, reader, for your interest in my work.

"You never change things by fighting the existing reality.
To change something, build a new model to make the existing reality obsolete."
(Buckminster Fuller)

Abstract

This dissertation addresses aircraft routing problems with crew pairing in the context of non-scheduled air transportation. These problems involve complex decisions in a highly dynamic and costly environment, where various civil aviation regulations must be followed. There is a lack of operations research literature on non-scheduled air transportation, and this type of service has significant differences from conventional (scheduled) transportation. Overall, this research covers real-world problems of two companies belonging to the sector, categorized in academia as a dial-a-flight problem and an aircraft recovery problem. The first refers to fractional ownership services with private aircraft sharing. In this scenario, the customer owns an equity part of aircraft managed by an airline, which entitles him/her to fly a certain amount of miles during the period. We proposed a detailed optimization model, MIP-based heuristics and an exact branch-and-price algorithm. The second problem refers to a rescheduling (recovery) of flights as a way to mitigate the damage arising from past disruptions (adverse weather conditions, mechanical failures, etc.). Given a flight timetable, we need to determine new departure times, redesign routes, reassign flights to different aircraft, and examine potential flight cancellations. We formulated network-flow, event-based and discrete-time models. Additionally, we developed tailored constructive and improvement heuristics. To verify the adequacy and coherence of the approaches, several experiments were performed with real-life data. In the first problem, all instances were solved optimally, and in the second, we were able to generate effective reschedules without canceling flights, in relatively short computing times.

Keywords: Non-scheduled air transportation, Dial-a-flight problem, Aircraft recovery problem, Crew pairing, Aircraft routing.

Resumo

Esta tese aborda problemas de roteamento de aeronaves com emparelhamento de tripulações no âmbito do transporte aéreo não regular. Esses problemas englobam decisões complexas num ambiente altamente dinâmico e custoso, onde várias regras da aviação civil devem ser respeitadas. Há uma escassez na literatura da Pesquisa Operacional a respeito do transporte aéreo não regular, que possui diferenças significativas ao do transporte convencional (regular). Ao todo, esta pesquisa abrange problemáticas reais de duas empresas pertencentes ao setor, categorizadas no meio acadêmico como Problemas de Reserva de Voos e de Recuperação de Aeronaves. O primeiro remete aos serviços de voos sob demanda com o compartilhamento de aeronaves privadas. Neste, o cliente tem posse de uma parte patrimonial da aeronave que fica sob os cuidados da companhia aérea, o que lhe dá direito a uma certa quantia de milhas no período. Nós propusemos um modelo de otimização bem detalhado, métodos heurísticos de programação matemática, como também um método exato de ramificação e preço. O segundo problema refere-se à reprogramação (recuperação) de voos como forma de mitigar o prejuízo oriundo de interrupções passadas (condições climáticas, avarias mecânicas, etc.). A partir de uma tabela de voos pré-definida, são determinados novos horários de decolagem, novas rotas, a realocação de voos às aeronaves, assim como os possíveis cancelamentos dos voos. Nós elaboramos formulações de rede de fluxo, baseada em eventos de decolagem e com tempo discreto. Ademais, desenvolvemos heurísticas construtivas e de melhoria, feitas sob medida. Para verificar a adequação e a coerência das abordagens, foram realizados vários experimentos com dados reais. No primeiro problema, conseguimos resolver todos os exemplares na otimalidade, e no segundo, fomos capazes de gerar reprogramações efetivas, sem cancelamento de voos, em tempos computacionais relativamente curtos.

Palavras-chave: Transporte aéreo não regular, Transporte aéreo sob demanda, Reprogramação de aeronaves, Emparelhamento de tripulações, Roteamento de aeronaves.

List of Figures

List of Tables

List of Algorithms

List of abbreviations and acronyms

ARP aircraft recovery problem

ACO ant colony optimization

ANAC Agência Nacional de Aviação Civil

ALNS adaptive large neighborhood search

B&B branch-and-bound

B&C branch-and-cut

BIP binary integer programming

B&P branch-and-price

BP&C branch-price-and-cut

BD Benders decomposition

CRP crew recovery problem

CG column generation

DAFP dial-a-flight problem

DWD Danzig-Wolfe decomposition

DP dynamic programming

EASA European Union Aviation Safety Agency

EDT earliest due time

FAA Federal Aviation Administration of USA

F&O fix-and-optimize

GDP ground delay program

GRASP greedy randomized adaptive search procedure

GUB generalized upper bound

HFVRPTW heterogeneous fleet vehicle routing problem with time windows

ILP integer linear programming

ILS iterated local search

LP linear programming

LNS large neighborhood search

MIP mixed-integer programming

MANOVA multivariate analysis of variance

MP master problem

OR operations research

OPL Optimization Programming Language

PRP passenger recovery problem

PDCGM primal-dual column generation method

RCESPP resource-contrained elementary shortest-path problem

RINS relaxation induced neighborhood search

R&F relax-and-fix

RMP restricted master problem

SLR systematic literature review

SA simulated annealing

SVNS stochastic variable neighborhood search

SP subproblem

SRC subset row cut

TOD transportation on demand

USA United States of America

WCL window of circadian low

VRP vehicle routing problem

Contents

Chapter 1

Introduction

Air transportation is an important industry driver, stimulating socioeconomic development of regions and promoting domestic and foreign markets as a whole. It facilitates intermodality in business logistics strategies and has intrinsic characteristics associated with speed, autonomy, reliability and safety. However, the logistics in air transportation are well-known for their high complexity in terms of operational planning and high costs involved (LIMA; BELDERRAIN, 2008). Moreover, large airline corporations generally have meager profits, in the order of only 1% of gross revenues (MCCARTNEY, 2012).

Given the low profitability and the presence of an increasingly demanding and competitive field of activity, optimization processes end up playing an important role in supporting decision-making. They commonly consist of quantitative approaches, such as mathematical models and solution methods, which efficiently assist decision-makers to determine routes, times and locations; assign passengers, cargo and crew to flights and aircraft; build work schedules; reduce delays, flight times, available fleet; improve contribution margins, revenues, resource utilization; facilitate the fulfillment of what was planned; offer insights, scenario simulations, systematization and automation of business processes; among many other complex tasks.

In fact, successful cases have been reported in operations research (OR) literature, including tools by ensuring efficient solution algorithms in a reasonable time for companies in this sector (BELOBABA; ODONI; BARNHART, 2015), e.g., schedule design, responsible for constructing flight timetables according to traffic forecasts among cities (ERDMANN et al., 2001); fleet assignment, which defines what type of aircraft will operate a certain route (REXING et al., 2000; SHEBALOV; KLABJAN, 2006; ZEGHAL et al., 2011); tail assignment, consistent with assigning a specific aircraft (identified by its tail number) to each flight leg (LIANG et al., 2015; KHALED et al., 2018); aircraft maintenance routing, a sequencing landing aircraft process to realize corrective and preventive maintenances (LIANG et al., 2011; MAHER; DESAULNIERS; SOUMIS, 2014; AL-THANI; AHMED; HAOUARI, 2016); and crew scheduling, which aims to assign and pair crew members to flights, reconciling flight tasks with break and rest requirements (MERCIER; SOUMIS, 2007; DUNBAR; FROYLAND; WU, 2014; HAOUARI; MANSOUR; SHERALI, 2019).

In conventional airline operations, travel services are provided based on pre-established times and locations, so that customers make their preferred choices from a predictable range of options defined by departure times, seat classes, origin and destination routes, etc. There is a governmental civil aviation authority – which in United States is Federal Aviation Administration of USA (FAA), in Europe, European Union Aviation Safety Agency (EASA), and in Brazil, *Agência Nacional de Aviação Civil (ANAC)*, for instance – that accounts for approvals and concessions to explore international and domestic passenger, cargo charter and mail carrier airlines. This category of services is called *scheduled* air transportation. Conversely, companies engaged in *non-scheduled* air transportation do not need to submit their operational acts of incorporation for prior approval by a regulatory agency, not having as a requirement, the pre-definition of their itineraries, which can or cannot own commercial logistical purposes (IAC-1223, 2000; ICAO, 2009). In this scenario, the flight schedule may be highly dynamic, as the departure/arrival times, origin/destination locations, fleet and crews are planned on a short-term horizon, usually varying from period to period. In a commercial airline, these decisions are made by customers themselves, and the company seeks for responding to demand, and in the non-commercial one, it is the company itself that deliberates schedules regarding the operation convenience and based upon the state of resources provided at that moment. This leads to greater nervousness and unpredictability in comparison with scheduled air transportation airlines (ZWAN; WILS; GHIJS, 2011). It is also relevant to emphasize that, contrary to the literature related to scheduled air transportation, the non-scheduled case is considered a new topic, an activity that is emerging thanks to the increased participation of air taxi services and fractional ownership programs in the market, and due to the practicality that private aviation offers (verticalization effect), therefore, this is a class of problems that has been barely explored.

In view of this, our study aims to apply OR as a scientific method to optimize the routing of different aircraft types, involving flight scheduling with crew pairing in the context of non-scheduled air transportation industry. The integrated optimization of these activities represents a meaningful challenge for decision-makers, as the resulting problems are significantly complex and imply the simultaneous use of multiple resources. Hence, the goal of this dissertation is to propose representative optimization models, as well as effective exact and heuristic solution methods to cope with large-scale realistic problem instances within acceptable computer runtimes. In specific terms, the formulations comprise mixed-integer programming (MIP) models. The exact method is derived from the branch-and-price (B&P) algorithm, whereas for heuristic approaches, we focus on elaborating MIP-based, tailored constructive and local search heuristics. The results of this dissertation have potential to contribute not only to the scientific literature domain, but also to the development of new computational tools, able to help corporations that belong or work in this environment.

As a scope of research, two companies that proffer non-scheduled air transport services are addressed here, for the commercial and non-commercial cases, classified in the literature as variants of dial-a-flight problem and aircraft recovery problem, respectively, in order to identify and validate pertinent requirements that may occur routinely in these operations. Given the particularities of each issue and the crew regulations that govern each situation, the first real-world case arises in the planning processes of on-demand panorama, carried out with a fractional management airline company operating in European and Asian countries; and the second comes from a problem of recovering flights for personnel transportation (mainly teams of employees) to maritime units, faced by a Brazilian oil and gas company.

1.1 Objectives

The dissertation in its essence is stimulated by the main and specifics objectives presented as follows.

1.1.1 Main objective

Demonstrate the potential benefits of decisions rooted in scientific methods from OR literature, which concisely and objectively, propose a set of alternatives and actions capable of representing the problem mathematically, and developing effective practical solution strategies to create optimized aircraft routings in the non-scheduled air transportation context.

1.1.2 Specific objectives

Basically, the dissertation comprehends four specific objectives, arising from unfolding of the main objective, namely:

1. Deepen the study of non-scheduled air transportation literature, seeking to identify related problems, and formulations/solution approaches that have been devised;

2. Create mathematical models that satisfactorily represent the problems and can help in decision-making processes of the involved companies;

3. Develop an exact branch-and-price algorithm; and

4. Elaborate heuristic methods capable of improving the computational time and effort, without compromising the solution quality.

1.2 Motivation and justification

This dissertation addresses challenging problems that integrate complex decisions of highly costly activities, encompassing the simultaneous use of multiple scant resources in an unpredictable and mutable logistical environment. Because these problems are located in a short-term horizon, whereupon the demand is known gradually and can be planned at most a few days in advance, the sector ends up requiring in response, quality solutions at a relatively short computational time, making necessary the development of effective solution approaches.

Furthermore, this study comprises formulations with cost functions that are difficult to handle, where the incorporation of aviation rules results, in its entirety, nonlinear structures with integer variables – configurations that are originally intractable in current terms of direct resolution by general-purpose optimization software. The models are extensions from the formulation of heterogeneous fleet vehicle routing problem with time windows (HFVRPTW), which belongs to the NP-hard class of problems.

Finally, non-scheduled air transportation is considered a recent subject, an activity that has been growing since the last decade, thereby having scarce literature. The raised studies revealed that this research presents problems with particularities that make them unique and quite relevant to air transportation as a whole. It is also worth noting that the potential results of this work can be adapted to multiple types of operations related to the non-scheduled case. Therefore, the research seeks to promote approximation between academia and industry, and has potential to generate partnerships with an impact on society.

1.3 Methodology

Once this dissertation uses OR as a scientific tool to assist logistical planning in the aircraft routing process, the methodological procedures of this research follow the modeling and simulation category of production engineering literature (NAKANO, 2010).

According to the classification proposed by Bertrand and Fransoo (2002), and Morabito and Pureza (2010), this research can be characterized as empirical-quantitative and normative. The research is called empirical-quantitative because it is oriented to develop mathematical models and solution methods for real-life problems, and is named normative because the approaches promote engagement in decision-making processes that actively influence strategic business choices for companies.

The modeling process used to solve the problems under study was the one proposed by Mitroff et al. (1974), a cycle that composes four main and sequential phases for the construction of a quantitative model: conceptualization, formulation, resolution, implementation. In the conceptualization phase, the researcher creates a conceptual formulation of the problem/system, making decisions about variables that need to be included in the scope. In the next phase, the researcher actually develops the quantitative model, thus defining causal relationships between the variables. Afterward, the resolution process begins through computational tests with real instances. Lastly, the results of the mathematical model are validated and then implemented, after which a new cycle can start. Mitroff et al. (1974) argue that a research cycle can arguably begin and end at any of the phases, provided that the research is aware of specific parts of the addressed solution process and, consequently, of claims the researcher can make based upon the research results.

1.4 Organization and major contributions

The remainder of this dissertation is organized as follows, where the structure of each chapter concerns an article style format. We also highlight the main contributions.

Chapter 2 presents a relevant literary background about the problems studied here, dial-a-flight and aircraft recovery. It was built from the application of a systematic literature review method, which was able to do a complete scan of papers published until 2023. The search revealed that our problems and optimization approaches are unprecedented.

In Chapter 3, we address a real-world dial-a-flight problem (DAFP) that represents a challenge for decision-makers, because, in addition to considering the customer preferences, maintenance requests, distinct aircraft types and different operating costs, this problem also incorporates crew assignments, resulting in the fulfillment of several labor rules, common in the FAA guidelines. To effectively representing the problem situation, we proposed a detailed optimization model, enriched by stronger MTZ-based constraints, linearization artifices and variable pre-fixing. Moreover, we developed MIP-based heuristics, solution methods derived from relax-and-fix and fix-and-optimize approaches. In our adaptation/improvement, these heuristics allow forward and backward temporal strategies, admit the regret of previously fixed variables, choose how many variables will be linearly relaxed, among other possibilities. The results of extensive computational experiments using real-life instances show that the additional crew constraints do not compromise the model performance, which can be effectively solved by a general-purpose optimization software. We were capable of solving optimally 80% of the instances with this formulation. The results of MIP-based heuristic methods were promising. They found even better solutions, within relatively short running times.

Chapter 4 focuses on describing a branch-and-price algorithm together with a set partitioning formulation for the same DAFP addressed in Chapter 3. This sophisticated approach comprises, within its branch-and-bound tree, an interior-point method that can offer well-centered primal and dual solutions, as a way to stabilize the column generation procedure, avoiding well-known drawbacks in the literature. Furthermore, we create a tailored labeling algorithm that efficiently solves the resulting subproblems from the reformulation. This algorithm is responsible for attending to all crew rules while it generates the aircraft routes. We can also opt for which branching strategy to follow (two-step or strong branching). Last but not least, we applied a primal MIP-based heuristic to obtain good integer solutions sooner. The branch-and-price algorithm managed to solve all the real-life instances optimally, taking less than 400 seconds on average, which indicates that it is an exact method with an excellent performance in practice.

In Chapter 5, we start the study of a real-world aircraft recovery problem (ARP) in the context of oil industry. This is a complex and challenging problem to solve because of its particular characteristics observed in practice. We need to determine a daily flight reschedule that satisfies several operational constraints and recovers all pending flights, while minimizing flight delays and costs related to helicopter usage and reassignments. We propose two MIP models to formulate the respective 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 with constructive and improvement procedures, able to produce high-quality solutions within acceptable computational times.

Chapter 6 extends the ARP by considering a set of aerodromes (in which is possible to have flight transfers among them) and different maximum durations of the daily working hours of the crew members. Three approaches are proposed, a continuous-time MIP model that faithfully represents the extended problem, a discrete-time simplification of the former model to generate better recovery plans, and a two-phase heuristic to cope with larger realistic problem instances. The first and last approach are extended contents from Chapter 5.

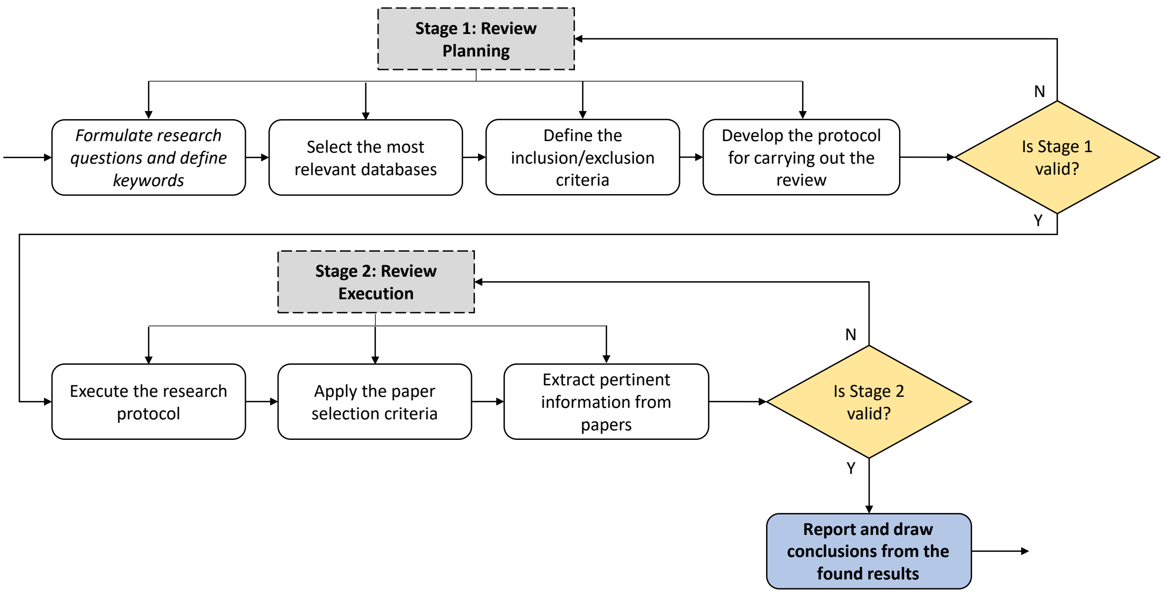
Finally, Chapter 7 shows an overall final discussion and concluding remarks together with perspectives for future research arising from the developments revealed in this study.

Chapter 2

Literature review

All the theoretical background relevant to this dissertation was raised through a systematic literature review (SLR), conducted according to the research stages proposed in Biolchini et al. (2005). The methodological-flow of this SLR is illustrated in Figure 1.

Figure 1 – Steps of the SLR.



Source: Adapted from Biolchini et al. (2005).

As a result of Stage 1 (review planning), we obtain the research protocol, which describes the specificities of how the SLR was carried out to answer the intended research questions – which by the way were: (i) How are our problems classified in the literature? (ii) Are there similar problems? (iii) What solution approaches have been proposed? (iv)

How can we determine the planning of these two problems in order to respect the imposed aviation rules and minimize the inherent operation costs? For the protocol construction, insights acquired from reading other SLRs were also considered, such as Uhlmann and Frazzon (2018), Clausen et al. (2010) and Santana et al. (2023). Figure 2 shows the developed research protocol.

Figure 2 – Research protocol.

DATABASE	Web of Science, Scopus and Compendex
MAIN STRINGS/KEYWORDS	"non-scheduled" OR "dial-a-flight" OR "on-demand air transportation" OR (("air taxi" OR "per-seat" OR "time-shared“ OR "charter flight") AND "scheduling") OR ("fractional" AND "ownership" AND ("management" OR "operations")) OR "aircraft recovery" OR (("flight" OR "aircraft" OR "helicopter") AND ("rescheduling" OR "rerouting")) OR ("irregular operations" AND "aircraft schedule") OR ("airline" AND ("disruption management“ OR ("schedule" AND ("perturbation" OR "disturbance")) OR "flight cancellations")) OR (("airline" OR "aircraft") AND "oil platform transport")
FIELDS	Title, Abstract and Keywords
PERIOD	Until 2023
LANGUAGE	English and Portuguese
INCLUSION/EXCLUSION CRITERIA	Conferences and Journals, Peer reviewed and Completely available; Operations Research area; and Related to DAFP or ARP, either subject or formulation
CONTENTS INCLUSION CRITERIA	Decisions, Rule sets, Formulations and Solution methods

Source: Own authorship.

Once Stage 1 was validated, we proceeded to Stage 2 (review execution). In all, 863 articles were found, leaving 448 after filtering for language, document type, publication stage and research area, and remaining 246 by removing duplicated works in the three databases. After performing the content analysis (title, abstract, conclusion, and for the most relevant articles, the full reading), a total of 101 papers prevailed (included in the dissertation).

Given the conclusion of Stage 2, as an output, our SLR revealed that the DAFP and ARP described here have particularities that make them unique in the literature. Regarding the first problem, we found only five works about DAFP in the last ten years, showing that this literature is very scarce, needing to be better explored. Related to the second problem, we note that there is no paper associated to the oil and gas industry.

Sections 2.1 and 2.2 present the theoretical background obtained for these two problems, first for DAFP and then for ARP. We start with the initial research efforts, and the remaining content is organized according to the type solution methodology, i.e., exact optimization methods, heuristics and metaheuristics, hybrid methods, etc. At the end of each section (specifically, Subsections 2.1.6 and 2.2.6), we exhibit a table that summarizes the main characteristics of the studied contributions. Ultimately, Section 2.3 concludes this chapter by showing our articles that have already been published or submitted.

2.1 The dial-a-flight problem

The category of problems related to transportation on demand (TOD) refers to the locomotion of passengers or goods between specified points of origin and destination at the request of users. More common examples are dial-a-ride transportation services for elderly and disabled people, urban courier activities, emergency vehicle dispatching, and which characterizes our first problem, aircraft sharing services, spread in the literature as dial-a-flight problem (DAFP) (CORDEAU et al., 2007).

In recent years, aircraft sharing has become more popular for a number of reasons. Increased security at airports has resulted in longer waiting times. In addition, the cost-reduction efforts made by airlines have led to staff cuts and reduced workload, affecting the flexibility of the provided service, especially when it concerns smaller regional airports. At the same time, technological advances are paving the way for the development of smaller aircraft, which are therefore cheaper. Because of this, an air taxi system that offers efficient, hassle-free, affordable, and that allows boarding/landing in congested outlying airports, without packed parking lots, long lines, numerous security checkpoints, flight delays, lost luggage, and enables greater freedom of choice and convenience, make air transport on demand more and more attractive.

Basically, DAFPs can be characterized by the presence of three often conflicting goals: (*i*) maximizing the number of requests served, (*ii*) minimizing operational costs and (*iii*) minimize user inconvenience. Service quality is usually measured in terms of excess travel time (i.e., the difference between a user's actual travel time and the minimum possible travel time). Operating costs are mainly related to the number of used vehicles, the total duration of a journey or the total distance traveled by the aircraft.

The daily management of a dial-a-flight system involves making decisions regarding three major aspects: request clustering, aircraft routing and flight scheduling. Request clustering consists of creating a group of customer requests to be served by the same aircraft due to its spatial and temporal proximity. Given this clustering, aircraft routing is tasked of deciding the order in which airports should be visited by each aircraft. Finally, the flight scheduling specifies the exact time that each airport should be visited. These decisions are obviously closely intertwined and a proper management of the system calls for their simultaneous optimization (FOY, 2013).

Related to the form of meeting the demand, dial-a-flight systems can be static or dynamic. In the first case, all requests are known beforehand (for example, at least one day in advance) and routes or decisions are not changed during the execution of planning, while in the second case, requests are unknown and revealed dynamically and aircraft routes must be adjusted in real-time (online) to meet the demand (CORDEAU et al., 2007). In contrast, we can also find different definitions about the word “dynamic” in the literature. According to Ghiani, Laporte and Musmanno (2004), in dynamic problems, all the parameters are considered to be known, even probabilistically, and the decisions must be taken at the beginning of period based on a sequence of changes carried out at instants defined within a time dimension, which is explicitly taken into account along the planning horizon. Therefore, the conflict over this term arises due to the fact that decisions are made in real-time or staged over time. In the present dissertation, we use the first definition (CORDEAU et al., 2007), in view of the second belongs to an older literature (JOHNSON; MONTGOMERY, 1974; BAKER, 1974; SHAPIRO, 2001), not usually cited in more recent works.

2.1.1 Initial efforts

Desaulniers et al. (1997), Keskinocak and Tayur (1998) were the first to explore aircraft routing and scheduling for DAFP. Desaulniers et al. (1997) solved a daily aircraft routing and scheduling problem, which consists of finding a fleet schedule that maximizes the profits. They proposed set partitioning and time-constrained multicommodity network-flow formulations, and obtained good computational results by employing column generation (CG) technique. Keskinocak and Tayur (1998) addressed a time-shared jet aircraft scheduling problem, which can be seen as a DAFP where each aircraft can serve only one customer at a time. The authors presented a 0-1 integer programming model to solve small and medium-sized instances, and also developed a heuristic method based on dynamic programming (DP) for larger instances.

Since then, other works, most of which were motivated by real-life applications, have emerged in the literature.

2.1.2 Exact optimization methods

Martin, Jones and Keskinocak (2003) built a flow-based integer linear programming (ILP) formulation to handle five aspects of fractional fleet management (reservations, scheduling, dispatch, aircraft maintenance, and crew requirements). They reported an 18.7% reduction in positioning legs for some aircraft and a $4.4 million savings in the first year after implementing their procedure.

Courier services also operate through on-demand delivery. Armacost et al. (2004) described an optimization system developed for the scheduling of packages in a North American courier airline. At the time the study was developed, the company was delivering around 13 million packages globally every day. The system consists of a MIP model that determines aircraft routes, fleet assignments and the package routings. Given the weakness of linear relaxations provided by the model, the strategy was to redefine the decision variables as composite variables (combination of two aircraft routes).

Karaesmen et al. (2005) took into account a problem with multiple types of aircraft, scheduled maintenance, and crew constraints. At first, they implemented network-flow and MIP models, for only aircraft scheduling constraints. After, the problem was extended by adding crew daily duty times. Due to the increased computational effort of larger instances, they developed a set partitioning-based formulation, a branch-and-price algorithm and a constructive heuristic. These approaches led to a significant improvement in the aircraft utilization (from 62% to over 70%), diminution of costs due to reduced empty moves, and hence increased profits.

Hicks et al. (2005) implemented a comprehensive three-module optimization system composed by a MIP model to simultaneously maximizes the use of aircraft, crews, and facilities within the scope of DAFP. This system made use of the GENCOL optimizer, which encompasses a CG approach to decompose large-scale mixed-integer nonlinear programming problems. In general, the approach was capable of generating savings of $54 million, while increasing fleet utilization of around 10% for an airline.

The term dial-a-flight problem originates from the work of Espinoza et al. (2008a). Their research dealt with the hiring of executive jets in the USA. They modeled the problem as a multicommodity network-flow formulation with side constraints and developed a variety of techniques to control the network size and to strengthen the quality of linear programming relaxation, proving optimality for instances with up to six airplanes.

Lee et al. (2008) addressed an air taxi service problem with probabilistic variables. They built an approach that describes the aggregate flow of passengers and aircraft without specifying event-level operation in order to define the optimal pricing of travel fares. Specifically, the approach combines two optimization models, a discrete-event model and an aggregate flow model, where the latter model abstracts away from the event-level complexity of the first one. A set of hypothetical scenarios were used to validate the proposed approach, revealing its potential for practical application.

Campbell, Ali and Silverwood (2020) formulated the problem of a tourist airline operating in Botswana/Africa, as multicommodity network-flow model using composite variables. The method takes many of the problem constraints into account at the variable creation stage, reducing the problem size in terms of variables and constraints. As such the method is mostly suitable for highly constrained problems.

Munari and Alvarez (2019) considered a problem in which the aim is to determine airplane routing and scheduling to fulfill a list of flight requests, minimizing the operational costs. They proposed a compact mixed-integer programming formulation, including aircraft maintenance events and service upgrades. One important and novel feature of this model is that it allows the anticipation or postponement of the beginning of flights and maintenance events within a given tolerance, affording more freedom to the decision-making process. Our proposed model with crew assignment for the DAFP is based on the formulation of Munari and Alvarez (2019). Both representations will be discussed further in the next chapter.

2.1.3 (Meta)heuristic methods

Fagerholt, Foss and Horgen (2009) considered an air taxi service problem in Norway. They introduced a heuristic algorithm for a strategic decision support tool that helps estimate the trade-off between fleet size and service. The heuristic method uses insertion operators to solve the problem and a local search stage to improve the incumbent solution.

Mane and Crossley (2012) developed an approach based on the System-of-Systems decomposition (decisions between interdependent systems) dedicated to couple the uncertain aircraft assignment problem of fractional operations and the aircraft design problem of manufacturers. A solution to this type of problem describes an aircraft design that directly improves operations and identifies operating strategies that influence, and take advantage of, the characteristics of the new aircraft.

Recently, Cordeau et al. (2023) studied a DAFP faced by one of the major safari airline companies in Tanzania. The objective is to determine the best set of itineraries for passengers based on their travel requirements by minimizing delays, the number of intermediate stops on routes and operational cost. Given the huge number of requests that the operation comprises, they proposed an adaptive large neighborhood search (ALNS) metaheuristic, enriched with local search operators and a set partitioning model.

2.1.4 Hybrid methods

Espinoza et al. (2008b) developed a parallel local search matheuristic was incorporated to the multicommodity network-flow model, allowing to explore large neighborhoods of the solution space, obtained high-quality solutions for large-scale real-life instances. This method was embedded within the core optimization technology of Espinoza et al. (2008a).

An on-demand air transportation type of problem was addressed by Ronen (2000). A set of revenue trips needed to be partially satisfied by a fleet of aircraft at minimal cost, and any remaining flights were to be sold to other operators. A large set of feasible candidate schedules is generated for each aircraft using heuristics. The best set is then selected using a MIP-type solver.

Similar problems to the one found in Martin, Jones and Keskinocak (2003) were later studied by Yao et al. (2008) and Yang et al. (2008). In the first paper, the authors discussed strategic planning issues, for example, aircraft maintenance, crew swapping, and methods to increase and differentiate demand. They modeled it as a set partitioning formulation and used a rolling horizon approach to solve it, where at each iteration, a CG method was applied to deal with the resulting subproblem. In the second paper, the authors proposed a scheduling decision support tool that couples a network-flow model and a heuristic algorithm aimed at increasing aircraft utilization. Additionally, these authors presented a set partitioning formulation and a B&P algorithm to solve the problem. These methods are improvements of those presented in Karaesmen et al. (2005). Zwan, Wils and Ghijs (2011) adopted the Yang's set partitioning model (YANG et al., 2008) and provided a detailed description of aircraft routing problem for per-aircraft air taxi operator of a Belgian company. Contrary to the CG used in Yang et al. (2008), they used a K-shortest path alike algorithm to generate feasible routing pools for set partitioning. The instance included 225 airports over 72 h time horizon. They achieved an estimated cost reduction of 12% on their routing plans with respect to using a human dispatcher.

Related to the work of Espinoza et al. (2008a), Engineer, Nemhauser and Savelsbergh (2011) introduced a CG approach making use of a DP that operates on the time-expanded network underlying the previous multicommodity flow model. The DP method alternates between a forward and backward search employing bounds derived from the previous search to prune the search space and remove irrelevant paths during advancing iterations, ensuring that an optimal path is found at the end of the procedure. They provide solutions for instances with up to 200 airplanes.

2.1.5 Other methods

In Maheshwari and DeLaurentis (2021), given the stochastic and dynamic nature of their DAFP, a hierarchical Markov decision process framework was proposed to enable the implementation of reinforcement learning techniques. They demonstrated the functioning of the two layers through simple example problems and a case study for a hypothetical dial-a-flight service, located in the Chicago Metropolitan area.

In Sumarti, Brahmandita and Aqsha (2022), a stochastic simulation scheme has been created to generate requests for an optimization model, in order to determine optimal flight pairings that minimizes the operational cost of a fractional aircraft ownership company from Indonesia. By means of assumptions and calculations in the simulation, the approach was able to determine the number of aircraft needed to be owned so the business will profit.

2.1.6 Overview

Compared to the 21 works revealed by the SLR, our DAFP is one of the few (together only with Martin, Jones and Keskinocak (2003), Karaesmen et al. (2005), Yang et al. (2008)) that covers the three most common conflicting goals of this literature (as mentioned at the beginning of Section 2.1). Once all requests must be served (either by the airline itself or by a third party), aircraft upgrade and crew overtime costs are minimized, and the main objective of the problem is to reduce positioning flights (i.e., trying to ensure that there is always an aircraft available at the time and location chosen by a customer), therefore, we also seek (in other words): "maximizing the number of requests served", "minimizing operational costs" and "minimizing user inconvenience". Additionally, our problem is the richest in terms of crew retraction. It presents a complete detail of legislation rules that are in the most followed civilian aviation regulatory agencies (such as the FAA and EASA). Consequently, our mathematical formulation ends up being the most accurate among those analyzed. Regarding solution approaches, we are the first to use R&F and F&O heuristics in this context, which have adaptations and enhancements never seen before. Our B&P algorithm is the only one that includes a method to stabilize the column generation procedure.

Table 1 gives a summary of the DAFP literature in a chronological order. For each paper, we describe the optimization objective; inform if the problem has heterogeneous fleet, aircraft maintenance or/and crew regulation with yes (Y) or no (N) indication; detail the developed solution approaches (accurate description as used by the authors) and categorize them into exact (EX), (meta)heuristic (MH), hybrid (HM) or other method (O), considering the most relevant approach if there is more than one type of method; identify whether most data are real-life instances (RL) or not (G); and reveal the size of the problem in relation to the aspects of time horizon, number of aircraft, types of aircraft (fleet) and flights (or requests). These instance aspects were collected according to the way the authors reported them in the paper (total, average or largest number), or even through a range (when the authors present tables with general information of the data). In the last row of the table, we put our DAFP to be compared with the other contributions.

We conclude this section by showing three charts in Figures 3, which illustrate the number of DAFP articles published in relation to journals, countries and over the years, respectively.

Table 1 – Overview and classification for literature focusing on dial-a-flight problem.

Paper	Objective	Problem Characteristics			Solution		Data	Dimensions			
		Multifleet	Maintenance	Crew	Approach	Type		Time horizon	Aircraft	Fleet	Flights
Desaulniers et al. (1997)	Aircraft fleet profits	Y	N	N	Set partitioning/Time-constrained model and DWD with B&B	EX	RL	24 h	142	15	635
Keskinocak and Tayur (1998)	Empty flight hours and amount of subcontracting	N	Y	N	0-1 ILP formulation and Divide-and-conquer heuristic	MH	G	72 h	10-30	1	20-100
Ronen (2000)	Cost of schedule, selling off flight and keeping aircraft idle	Y	Y	Y	Set partitioning model embedded in Schedule generation method	HM	RL	24 h-48 h	48		50-98
Martin, Jones and Keskinocak (2003)	Schedule, ferry and outsourcing costs	Y	Y	Y	ILP formulation/CPLEX	EX	RL	48 h-72 h	> 100		
Armacost et al. (2004)	Total flying cost	Y	N	N	Next-day-air network design MIP formulation	EX	RL	24 h	160	7	
Hicks et al. (2005)	Total flying cost	Y	Y	Y	Multicommodity network-flow model/GENCOL	EX	RL	24 h-72 h	25		
Karaesmen et al. (2005)	Repositioning legs, hidden operations and subcontracting costs	Y	Y	Y	MIP models, Constructive heuristic and B&P method	EX	RL	24 h	100		300
Lee et al. (2008)	Revenues	N	N	N	Discrete-event/aggregate flow model with Gradient algorithm	EX	G	1 week	4-10	1	
Espinoza et al. (2008a)	Total flying cost	Y	N	N	Multicommodity network-flow with side constaints/CPLEX	EX	RL	24 h	4-8		43-82
Espinoza et al. (2008b)	Total flying cost	Y	N	N	Parallel local search matheuristic	HM	RL	24 h	> 150		
Yang et al. (2008)	Repositioning legs, hidden operations and subcontracting costs	Y	Y	Y	Network-flow formulation, Constructive heuristic and B&P method	EX	RL	24 h-48 h	37-211		53-400
Yao et al. (2008)	Operating costs	Y	Y	Y	Rolling horizon approach and CG algorithm	HM	RL	72 h	35-75		42-341
Fagerholt, Foss and Horgen (2009)	Trade-off between fleet size and service	N	N	N	Heuristic with insertion operators and Local search methods	MH	RL	24 h	5-20	1	1-200
Engineer et al. (2011)	Transportation costs	N	N	N	Relaxation-based dynamic programming-based CG algorithm	HM	RL	24 h	10-200	1	60-1600
Zwan, Wils and Ghijs (2011)	Total flying cost	Y	Y	Y	Set partitioning model, CG with K-shortest path alike algorithm	HM	RL	72 h			
Mane and Crossley (2012)	Total flying cost	N	Y	Y	System of systems decomposition-based and Simulated Annealing heuristic	MH	G	24 h		1	200-1600
Munari and Alvarez (2019)	Ferry and aircraft upgrade costs	Y	Y	N	Network-flow model/GLPK	EX	RL	72 h	19-50	6	35-109
Campbell, Ali and Silverwood (2020)	Total flying cost	Y	N	N	Multicommodity network-flow/CPLEX	EX	RL	24 h	17	2	200
Maheshwari and DeLaurentis (2021)	Operating costs	N	N	N	Hierarchical Markov decision process formulation	O	RL	< 24 h	10	1	
Sumarti, Brahmandita and Aqsha (2022)	Operating costs	N	Y	Y	Stochastic simulation scheme with Mathematical model	O	G	1 month	5	1	
Cordeau et al. (2023)	Delays, number of intermediate stops and operational costs	Y	N	N	ALNS enriched with Local search operators/Set partitioning model	HM	RL	24 h	7-15		91-343
Our DAFP	**Ferry, aircraft upgrade, outsourcing and overtime costs**	**Y**	**Y**	**Y**	**Network-flow model/CPLEX, R&F-F&O heuristic and B&P method**	**EX**	**RL**	**24 h**	**13-56**	**3-7**	**23-149**

Note: Abbreviations used in the table: EX: Exact method; MH: (Meta)heuristic; HM: Hybrid method; O: Other methods; Y: Included or mentioned; N: Not included nor considered; " ": Not mentioned or not relevant; G: Generated data; RL: Real-life data.

Source: Own authorship.

2.2 The aircraft recovery problem

Many times, daily flight schedules cannot be made as previously planned, whether as a result of mechanical failures, crew reassignments, airport closure, variations on travel times and ground times, unavailable fuel supply, problems with the air traffic control, or even because of inclement weather conditions. These unexpected events can be defined as disruptions, and if they are not addressed in a timely and appropriate manner, they will affect crew connections and passenger itineraries and may result in significant damage to the airline's profitability and image. Consequently, air companies will typically reschedule a number of aircraft taking flight plans, aircraft routings, maintenance schedules, crew status and other pertinent information into account. Rescheduling practices include aircraft swapping, flight cancellation, use of standby aircraft, and departure time holding. These adjustments must satisfy maintenance exigencies, station departure curfew restrictions and aircraft balance requirements, especially at the beginning and end of a recovery period. In most cases, it is critical for the company to "recover" their operations as quickly as possible, which may be in a few hours or the end of the day at the latest.

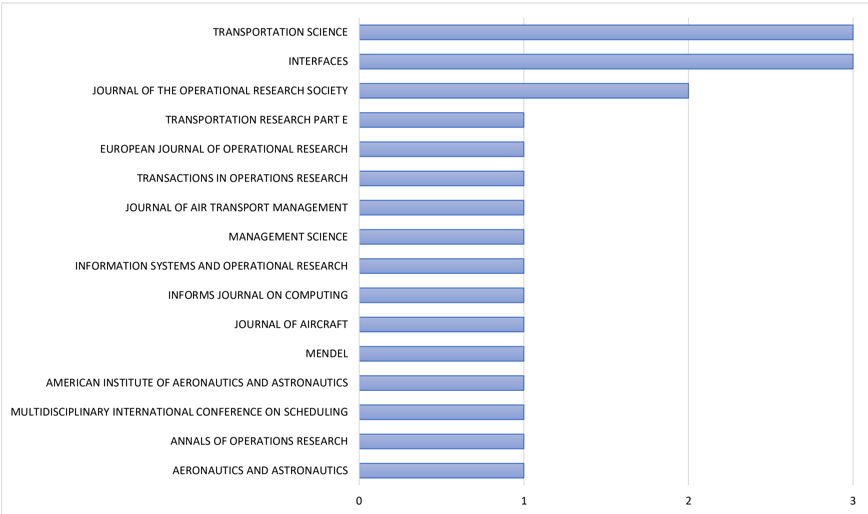
Problems that comprise disruption occurrences are called Airline Recovery Problems. According to Belobaba, Odoni and Barnhart (2015), this category is composed of three subproblems: aircraft recovery problem (ARP), crew recovery problem (CRP) and passenger recovery problem (PRP). The first one corresponds to the second problem addressed in this dissertation. ARP is the recovery process that focuses on the aircraft resource and can be formulated as follows: given a flight schedule and a set of disruptions, determine which flights to delay or cancel, and reassign the available aircraft to the flights with respect to a preferred purpose. The objective is often minimizing the operating costs, maximizing the profit or minimizing the time required to return to the original schedule.

Before 2009, the majority of airline recovery publications focused on aircraft recovery, in part because: (*i*) aircraft are the most constraining and expensive resource; and (*ii*) aircraft recovery is a smaller and simpler problem than crew recovery (which involves complex regulations and pilots' preferences). Despite this, ARP is still an active research subject, where the efforts have been focused on better representing real-world networks and decreasing the computation time (HASSAN; SANTOS; VINK, 2020).

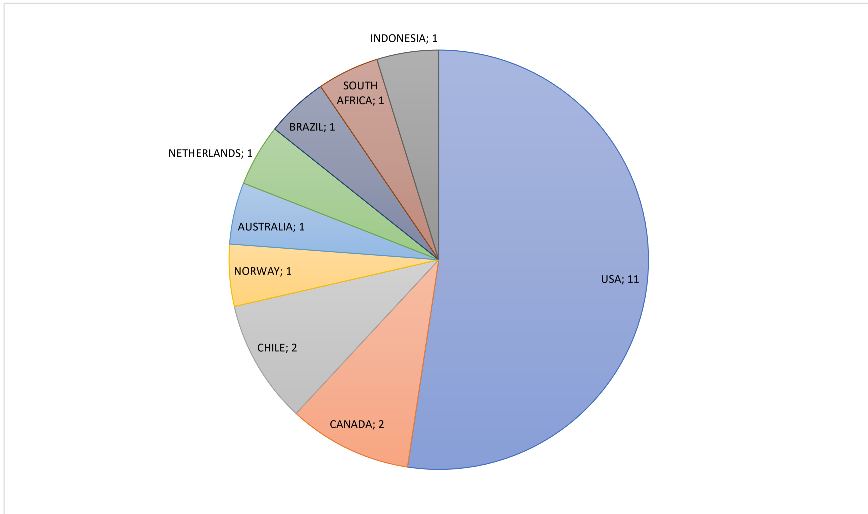
2.2.1 Initial efforts

In a broadest understanding, the first study on ARP was proposed by Teodorović and Guberinić (1984). In this work, one or more aircraft are unavailable and the objective was to minimize the total passenger delays by reassigning and retiming the flights. The authors devised a heuristic that sequentially constructs the chain of flights to be flown by each aircraft. Their solution assumed a single fleet type and ignores all maintenance constraints. They presented a very simple example with only eight flights. Teodorović

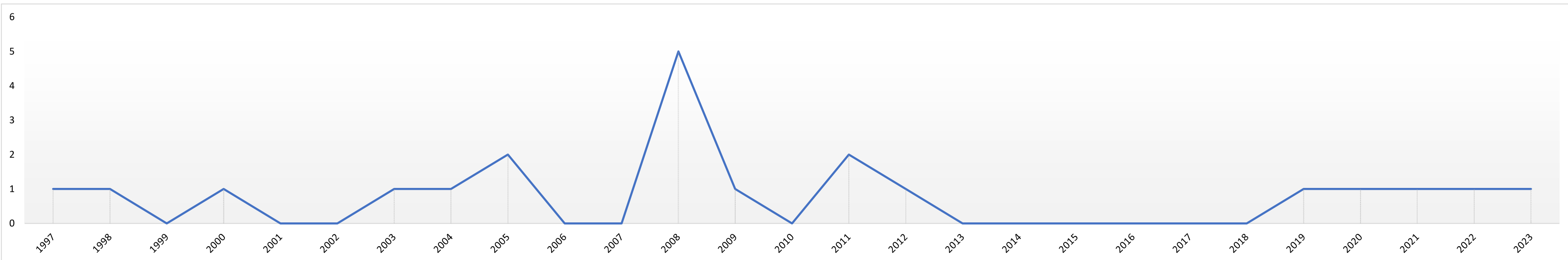
Figure 3 – Number of DAFP articles published by journals, countries and over the years.



(a)



(b)



(c)

Source: Own authorship.

and Stojković (1990) extended the study to also consider airport curfews. The described method was tested on a small example of 14 aircraft and 80 flights. Teodorović and Stojković (1995) further extended their model to include crew considerations too. The proposed method was tested on 240 different randomly generated numerical examples.

2.2.2 Exact optimization methods

Jarrah et al. (1993) presented two network-flow models for solving the ARP, and used flight delays, cancellations, aircraft swappings, and reserve aircraft as recovery strategies. In their approach, flight cancellations were not allowed for one model whereas delaying flights were not considered in the other model, and the authors suggested the possibility of finding better solutions by combining delays and cancellations to capture their interactions. The authors argued that the models are general minimum-cost networks which involve multiple sources and sinks. However, the source nodes can be combined to one master source node using arcs with flows bounded by one. For this transformation, they implemented the Busacker Gowen's dual algorithm (BUSACKER; GOWEN, 1961). To assess the cost of delaying or canceling a flight, the authors constructed a disutility function, guided by the total number of passengers, the number of passengers with a downstream connection, lost crew time, and disruption of maintenance. The Busacker-Gowen's algorithm was also used to treat three real-life scenarios from airports in the USA, producing solutions with shorter delays than the ones in practice. The models were subsequently integrated into the decision support system of the United Airlines (RAKSHIT; KRISHNAMURTHY; YU, 1996).

The papers by Cao and Kanafani (1997a, 1997b) are basically extensions from the work of Rakshit, Krishnamurthy and Yu (1996). In Cao and Kanafani (1997a), a 0-1 quadratic programming formulation was presented to maximize flight revenues minus swapping and delaying costs. Both delays and cancellations, ferrying (i.e., flying a deadhead aircraft to a station for the next operation), and multiple aircraft type swapping were taken into consideration in their model. From a special structure found in the quadratic formulation, Cao and Kanafani (1997b) proposed a linear programming approximation algorithm and reported good computational results at a reasonable CPU times.

Mathaisel (1996) described a novel approach on the integration of computer science and operations research techniques in airline industry. The study focused on development of a decision support system for the dispatchers in airline operations control centers. Several optimization methods were embedded in the environment. In cases of cancellations or significant flight delays, aircraft rescheduling alternatives minimizing the effects of the disruption are generated. The problem was represented by a network flow model, and the solution procedure was based on an out-of-kilter network flow algorithm.

Ground delay program (GDP) is one of the several programs that the FAA is administering for efficient and equitable use of scarce airspace and airport capacity. In poor weather conditions, the FAA may decide that the number of planned arrivals at an airport will exceed the airport's capacity. In such cases, GDP is initiated, and usually the arrival times of these flights are delayed. Luo and Yu (1997) addressed such disruptions. The performance measure was the percentage of flights that are delayed more than 15 minutes. The problem was modeled as an ILP problem. Valid inequalities and variable reduction methods are used to solve the problem.

Thengvall, Bard and Yu (2000) used a single-commodity network-flow model for recovery following a hub closure. Using the framework based on a time-space network with flight arcs, ground arcs, and overnight arcs, the authors added arcs for ferrying, arcs for time-shifted copies of original flights (accommodating delays), protection arcs, and through-flight arcs. This model can handle delays, cancellations, as well as swaps among different fleets. The authors solved the problem with various objective functions such as minimizing total cancellation and delay costs, and preserving as many of the original aircraft routings as possible. After, Thengvall, Yu and Bard (2001), Thengvall, Bard and Yu (2003) extended this work to consider the closure of a hub, as well as multiple fleets. Three mixed-integer programming models were introduced: two so-called preference models, which are based on timeline networks for every subfleet, and a model based on time-bands.

In Rosenberger, Johnson and Nemhauser (2003), an ARP was modeled as a set partitioning problem with capacity constraints. Prior to solving the model, a pre-processing heuristic determined which aircraft should be subjected to rerouting and rescheduling. Tests comprising real-life instances with 32 to 96 aircraft and 139 to 407 flights were presented.

In Xiuli and Jinfu (2007), a grey programming method was introduced to solve a flight schedule recovery problem. The term "grey" represents the fact that the formulation contains both white (deterministic) and black (stochastic) parts. The white part has "white numbers" (i.e., precise values), while the black part is defined by "grey numbers" (a family of time series random variables, where only upper and lower bounds are known). The authors contributed to the development of a mathematical model for the problem including grey parameters, and a grey simulation technology rooted on a shortest path algorithm to get optimal results. The feasibility and effectiveness of the model and solution method were validated by the simulation results.

Filar et al. (2007) sought to solve the ARP from a "common good" point of view. They started from the premise that with a judicious choice of interventions (for example, propagation of delays and flight cancellations) to minimize the harmful effects of interruptions in the schedule experienced by all stakeholders: passengers, airlines, airport companies and air traffic regulatory agencies. Among its characteristics, the use of the time discretization strategy – as in Bard, Yu and Arguello (2001) – is noteworthy, and its adaptability to different situations as well.

Jafari and Zegordi (2010) presented an assignment model for solving the ARP and reassigning disrupted passengers simultaneously, using sequential recovery stages within the time window (a rolling horizon algorithm). The model examines possible flight retiming, aircraft swapping, overflying, ferrying, utilization of reserve aircraft, cancellation and passenger reassignment to generate an efficient schedule recovery plan. The method uses aircraft rotations and passenger itineraries instead of flights, without consider maintenance constraints. Jafari and Zegordi (2011) extended the work by incorporating more operational rules. Due to the high complexity of the algorithm, the method was only tested on disruptions with 13 aircraft of 2 fleet types. The authors did not demonstrate that the method was computationally efficient, nor did they show that the model can deal with disruptions that reflect operations of a larger airline.

Gao et al. (2012) dealt with an ARP whose flights have features that prioritize each other relatively, such as passenger class and flight status. Depending on the priority, different costs related to delays and cancellations are applied. The authors presented an optimum polynomial-time algorithm which was tested with a problem instance with 8 flights and 2 airports.

Akturk, Atamturk and Gurel (2014) were the first to include cruising speed in the decisions in a mathematical model. Even though the increase of cruising speed shortens the travel time, hence contributing to reducing delays, it also increments fuel consumption in a non-linear trend. The trade-off between speed and consumption was significant and should be considered when one wants to minimize both economical costs from the rescheduling and CO_2 emissions. The authors proposed a mixed-integer conic quadratic model and concluded that allowing cruising speed provides better results both in terms of cost and delay.

The approximated delay costs considering the random flying time around the planned flying time is introduced by the time-band approximation model of Xu et al. (2015) to recover flight operations and minimize the delay and cancellation costs. Their results demonstrate that the model on flight operations recovery with the random flying time may be more efficient than the model on flight operations recovery with the planned flying time.

Wu et al. (2017) adopted an iterative fixed-point method for ILP to generate feasible flight routes that are used to construct an aircraft reassignment in response to the grounding of one aircraft. Two division methods are proposed to divide the solution space into several independent segments and implement a distributed computation. Comparison with CPLEX CP Optimizer showed that the iterative fixed-point method was essential to find an aircraft reassignment when unexpected events happen, and the second division method was more promising when dealing with long haul airline disruption problem.

Arıkan, Gürel and Aktürk (2017) developed a new flight network representation for an ARP integrated to crew and passenger recovery decisions. The problem size in their flight network is kept within limits so that real-time solutions can be provided since it does not require discretization of departure times and cruise speed decisions. Comparable to Akturk, Atamturk and Gurel (2014), the authors implemented aircraft cruise speed control and proposed a conic quadratic MIP formulation, which evaluated the passenger delay costs more realistically by its explicit representation. Using the approach, they managed to achieve solutions with a maximum optimality gap of 1.5% for networks having 288 and 473 flights.

Erkan, Erkip and Safak (2019) proposed a generic mathematical model to solve an ARP under a collaborative decision environment. The studied ARP considers a single fleet and time slots at the airports. The proposed model has the least amount of essential constraints and allows stakeholders to expand it and use it for different purposes, adding new constraints, or using different objective functions according to their purposes. For validation, the authors defined a base case, with optimization objectives considered common to any stakeholder, using real data from 6 h of operation at the Minneapolis airport, USA. Applying a general-purpose optimization solver with a time limit of 20 min, improved solutions were obtained compared to the rescheduling performed by the airport over the same period.

Sun, Liu and Zhang (2021) addressed integrated aircraft and passenger recovery for airline schedule disruption by developing a MIP model and proposing a solution algorithm based on the modification of time-band network and generation of candidate passenger itineraries. They also introduced an inter-modal concept to expand airline aviation networks by including ground transportation modes. The results of the numerical experiments showed that their modeling and solution algorithm considerably reduced the number of disrupted passengers and total disruption cost.

Liu, Sun and Zhang (2022) developed a MIP model to incorporate high-speed rail (HSR) transport mode into an aviation network for aircraft recovery purposes. The air-rail inter-modal strategy focuses on occasional operational integration of existing airside and HSR infrastructure capacities. Experimental computations are performed with flight schedule operation data from one of Chinese airlines and railway resource data from China's high-speed railway network. By comparing recovery outcomes for a pure aviation network and an air-rail intermodal network, the air-rail inter-modal strategy was shown to help reduce the number of cancelled flights and the total disruption cost.

In Cadarso and Vaze (2023), an ARP that imposes monetary compensations to passengers in case of disruptions was studied. They proposed a passenger response model with nonlinear cost terms to recover airline schedules, aircraft, and passenger itineraries while endogenizing the impacts of airlines' decisions on passenger compensation and response. Afterward, they linearized the objective function, in combination with a delayed constraint generation for ensuring aircraft maintenance feasibility and an acceleration technique that penalizes deviations from planned schedules.

Khiabani et al. (2023) proposed a mixed-integer linear programming model to represent their integrated aircraft and crew recovery problem. The formulation considers individual flight legs instead of strings, which leads to more accurate schedules and better solutions. Furthermore, a Benders decomposition approach is used to solve the proposed model.

2.2.3 (Meta)heuristic methods

Argüello, Bard and Yu (1997) proposed a model including turnaround time, airport time curfew, aircraft balance and compulsory maintenance schedules. The authors also proposed a greedy randomized adaptive search procedure (GRASP) algorithm, tested on real-life instances up to 16 aircraft and 42 flights. The model was based on the formulation presented in Arguello (1997), after being used to derive the model by Bard, Yu and Arguello (2001).

Løve et al. (2002) implemented a steepest ascent local search heuristic using connection network representation for an ARP with flight schedules extracted from data of British Airways. On average, less than 10 seconds were required to find a feasible revised flight schedule that includes all planned flights on a given day.

Løve et al. (2005) proposed several heuristic approaches based on a network representation with the aim of handling problems of a realistic size (about 100 aircraft and 500 flights) in real-time (no more than three minutes). The approach tried to balance the trade-off between delays, cancellations and swaps. They tested their approach with disruptions from British Airways, and revised flight schedules with good quality were generated in less than 10 seconds on average.

Tang et al. (2009) extended the resource assignment model of Teodorović and Stojković (1995) and presented a greedy random simulated annealing algorithm to deal with aircraft recovery issue of multi-fleet. The modified model took into account aircraft flow imbalance and flights merger strategy. In the algorithm, neighbors are obtained by comparing aircraft route pairs, either of which is disturbed. Empirical results demonstrated the ability of the new model and algorithm to quickly explore a wide range of unbalanced scenarios and to produce an optimal or near-optimal solution in time.

Qiang, Xiao-wei and Jin-fu (2009) developed a greedy simulated annealing algorithm, combining characteristics of GRASP and simulated annealing (SA). The combination of heuristics improves the efficiency of the neighborhood selection and decreases the probability of local optima. The objective of the model is to minimize the total passenger delay time. On the same thought thread, a combination of GRASP and ant colony optimization (ACO) was used by Xiuli and Yanchi (2012). Compared to the original GRASP algorithm, it provides a high global optimization capability. The authors state that the model was tested on a multifleet network with 50 aircraft and more than 5 aircraft types.

D'Ariano, Pistelli and Pacciarelli (2012) modeled an ARP as a job-shop scheduling problem, in which a job (in the ARP's case, an aircraft) must perform a prescribed sequence of operations on specific machines (i.e., airport resources), including additional real-world constraints. Two solution methods were adopted and compared. The first combines the use of a set of heuristics to determine a good feasible solution, which is then used as the starting point in a routine based on a B&B algorithm. The second applies a tabu search algorithm, disturbing the pre-defined routes for each aircraft and searching for a better solution in the vicinity. Using real data from the Fiumicino airport in Rome, Italy, it was observed that the tabu search algorithm performed better.

Wu and Le (2012) developed a model based on flight strings instead of traditional individual flights. They transform these strings into a time-space model that considers maintenance constraints and regulations. The model is solved with a heuristic that was developed by the authors called the iterative tree growing with node combination. The model is tested on a dataset from China Airlines consisting of 170 flights, 5 fleets, 35 aircraft, and 51 airports.

Zhang, Lau and Yu (2015) proposed a two-stage heuristic for the integrated aircraft and crew recovery problem. In the first stage, the aircraft recovery with partial crew considerations model is built. This model is based on the traditional multi-commodity network model for the aircraft schedule recovery problem. In the second stage, the crew schedule recovery with partial aircraft consideration model is built. The authors propose a new multi-commodity model for the crew schedule recovery. The two stages are run iteratively until no improvement is found. The algorithm improved the solutions of the other two algorithms for all scenarios. Although the algorithm had a higher run-time, it never exceeds 72 s.

Zhao and Chen (2018) presented a weight-table heuristic algorithm for the ARP. The authors only consider disruptions from airport closures due to bad weather conditions. All common disruption recovery options are considered, however, maintenance constraints are not included in the model. A single case study consisting of 6 aircraft and 31 flights. The computation times are not presented.

Hu et al. (2021) constructed a two-objective MIP model and a heuristic with multi-directional and stochastic variable neighborhood search (SVNS) algorithm for an ARP considering passengers' recovery with willingness (circumstance in which a passenger obeys the airline assignment) under various itinerary disruption situations. The two objectives of optimization are to minimize airline's recovery cost and reduce passenger recovery loss. Experiments with the data collected from Air China showed the stability and efficiency of the approaches.

Ji et al. (2021) proposed a build-in flight feasibility verification algorithm to improve the rescheduling of an ARP with lexicographic preference of flight priorities. A novel model of the feasibility verification problem was given, which is equivalent to the formulation of a maximum clique problem for networks. The authors tested their algorithm on the real data gathered from a Chinese airline company, and the experiments revealed that the algorithm ran fairly quickly and could be plugged into other scheduling algorithms easily.

Evler et al. (2022) presented a rolling horizon algorithm that incorporate a network delay formulation with concepts of integrated aircraft, crew, and passenger recovery decisions. The model determines assignment of aircraft to flight routes and integrates it with allocation of scarce resources to aircraft turnarounds at the central hub airport, allowing to estimate the delay propagation in an airline network. When evaluating different scenarios, the authors achieved a full recovery of the flight schedule at low and moderate delay situations. Despite the lower efficiency of turnaround recovery in medium and high delay scenarios, the combination of flexible aircraft assignments and ground operations still generated additional cost savings of at least 21%.

Lee, Lee and Moon (2022) proposed Q-learning and Double Q-learning algorithms with a reinforcement approach for ARP derived from the real-world case of an airline in South Korea. The proposed approach presents an artificial environment of daily flight schedules and the Markov decision process for aircraft recoveries. Computational experiments showed that reinforcement learning algorithms recover disrupted flight schedules effectively, and that their approaches flexibly adapt to various objectives and realistic conditions.

Xu, Wandelt and Sun (2023) presented a novel approach to integrated airline recovery under decisions with the uncertainty of epidemic transmission probability captured through a Wasserstein distance-based ambiguity set. To efficiently solve the model, they elaborated a B&C algorithm combined with a large neighborhood search heuristic to iteratively add infection cost-related cuts based on the established epidemic propagation network.

Recently, Ding et al. (2023) built a mixed integer nonlinear programming (MINP) model for an integrated airline recovery problem. Given the computational challenge of solving the model, they develop three solution approaches, a CPLEX-based exact solution technique with adequate preprocessing techniques, a variable neighborhood search (VNS) algorithm with well-designed operators and state evaluator, and a Deep Reinforcement Learning (DRL) incorporated with the VNS to pre-trained policy and value function based on transfer learning.

Wang et al. (2023) developed a data-driven heuristic method to solve an ARP of China South Airlines from Beijing/Chine. Inspired by the data analysis results divided into different scenarios according to their delay reasons, the heuristic was enhanced to imitate dispatcher actions based on two basic operations: swapping the tail numbers of two flights and resetting their flight departure times.

Other ARPs solved by heuristic approaches can be consulted in Andersson (2006); Liu et al. (2006), Liu, Jeng and Chang (2008); Liu, Chen and Chou (2010); Le, Gao and

Zhan (2013); Sousa et al. (2015); Zhu, Zhu and Gao (2015); Xu and Han (2016); Hu et al. (2017); Zhang (2017); Khaled et al. (2018); Šarčević, Rocha and Castro (2018); Lin and Wang (2018); for example.

2.2.4 Hybrid methods

Yan and Yang (1996) made a decision support framework to handle schedule perturbations in airline industry. The authors assumed a single fleet type and focus on disruptions occurred due to an aircraft breakdown. The problem was represented on a time-space network. Based on this representation, they proposed ARP basic models with side constraints. The models were solved using lagrangian relaxation and a subgradient algorithm, and the computational study considered real-life problems of a major Taiwan air carrier. Yan and Tu (1997) extended the models for the case of heterogeneous fleets.

Clarke (1997) proposed a comprehensive framework for reassigning operational aircraft to scheduled flights in the aftermath of irregularities. Multiple aircraft type swapping, flight delays and cancellations, as well as the impact of air traffic management initiatives and crew availability were incorporated in the modeling. About the solution methods, he developed a tree-search heuristic and a set packing-based optimal solution algorithm.

Using the connection network as the underlying network, Andersson and Värbrand (2004) based their approach on the set packing problem with generalized upper bound (GUB) constraints, which ensures that each aircraft is assigned exactly one route. The problem was solved with a lagrangian relaxation-based heuristic and a method based on the Danzig-Wolfe decomposition (DWD).

Le and Wu (2013) extended the work presented in Le, Gao and Zhan (2013) to include crew recovery. As in the previous work, the authors use flight strings to represent a sequence of flights. An iterative tree-growing algorithm with nodes combination method is proposed to speed up the computational time. The authors consider maintenance requirements and pilot union regulations. A case study using data from a Chinese airline is presented.

Eggenberg, Salani and Bierlaire (2010) introduced a constraint-specific approach that simultaneously considered the aircraft, crews, and passengers. A different recovery network was generated for each kind of resource to reduce the problem scale. A set partitioning model was then created to embed the resources in one recovery scheme. Subsequently, a CG approach was used to solve the model.

Vos, Santos and Omondi (2015) presented an innovative dynamic modeling framework to the aircraft schedule recovery problem. The framework relies on the combined usage of an efficient aircraft selection algorithm and a LP model based on parallel aircraft specific time-space networks. This approach allows for very detailed aircraft specific constraints, and thereby closely portrays reality. The results showed that in all the test cases a solution was found within a short and appropriated time window, confirming the validity of the model framework for real-life application.

Maher (2016) studied an ARP integrated with crew recovery schedules. He proposed a column-and-row generation framework that extends existing generics B&P methods until then and reduces the problem size. This extension considers multiple secondary variables and linking constraints. The proposed algorithm is compared to a standard column generation approach. On average, the column-and-row generation method had a 27% lower run-time. The authors tested the method on both a point-to-point and a hub-and-spoke network with 262 and 442 flights, respectively.

Zhang et al. (2016) developed a three-stage sequential heuristic framework to solve the integrated aircraft and passenger recovery problem. In the first stage, the flight schedules and aircraft rotations are recovered. The next two steps iteratively solve the flight rescheduling problem and the passenger recovery problem. A time-space network representation is used together with a MIP formulation of the model. The proposed algorithm is tested based on the same data sets used by the ROADEF 2009 challenge. The algorithm can beat the finalists of the challenge on all datasets.

Liang et al. (2018) developed a framework where a master problem was used to select routes and subproblems were used to generate routes. Airport capacity constraints are explicitly considered in the master problem while maintenance constraints are in the subproblems. In the suggested framework, aircraft are allowed to swap their planned maintenance, if all constraints are satisfied. The approach is based on a CG framework. The proposed framework is validated and tested on eight real-world scenarios, which are based on the scenarios used as benchmark problems.

Vink et al. (2020) extended the work from Vos, Santos and Omondi (2015) by considering passengers' itineraries and aircraft maintenance requirements when solving the ARP. The authors modeled passengers' delay costs by pre-computing a delay cost matrix for both direct and connecting passengers. Maintenance constraints are directly considered and parallel-time space networks are used to track the route of each aircraft. The problem was formulated as a MIP model that was dynamically solved (i.e., a recovery solution is produced every time new information about disruptions is made available). The selection algorithm proved to be efficient, providing an initial solution within a couple of seconds and producing a near-optimal solution in 22 s on average.

Shao et al. (2020) developed a multi-objective model that provides new flight schedules aiming to minimize the total flight delay and the sum of the probabilities of six operational risks (airspace control, flight collisions, ground service, aircraft parking, ground control and taxi conflicts). Since such risks are each the result of different operational factors, their probability values act as bounds for such factors. To solve the model, the authors used a multi-start algorithm with intelligent neighborhood selection. The method proposed by the authors was able to reduce the total delay of flights by up to 56% and risks by up to

5% of the value proposed by the current method of operation.

Yetimoğlu and Aktürk (2021) addressed aircraft and passenger recovery problems simultaneously and proposed a novel matheuristic algorithm to solve them. The objective is to produce recovery plans that maximize the airline's profit while mitigating the passenger dissatisfaction, also considering seat capacity limitations. In their computational results, they used a major U.S. airline's daily schedule and clearly demonstrated an optimal trade-off between operating and passenger-related costs.

Huang et al. (2022) made an iterative cost-driven copy generation method for the ARP. It incorporates a copy evaluation method which assesses each eligible copy by using the dual information of the ARP LP relaxation, as well as a copy filtration method which effectively controls the number of copies that are generated and added to the ARP model to control the problem size. Their computational experiments using real airline data demonstrated that the proposed method is able to iteratively reduce the total recovery cost and can lead to very promising recovery solutions in less time.

Yan and Chen (2022) employed a network flow technique to construct an optimization model that aims to minimize the total operating costs and efficiently deal with fight rescheduling problems (gate reassignment) after a typhoon disruption event. To efficiently solve realistically larger problems, a heuristic algorithm was developed. To evaluate the model and the heuristic algorithm, a case study based on the operations of a major Taiwan airline was performed. The test results demonstrates that the model and the algorithm could be useful references in actual operations.

Zhao, Bard and Bickel (2023) proposed a two-stage approach composed of a single-commodity and a multi-commodity network model with side constraints, determining first, the flights that are most likely to be affected, and then adjusting their schedules to achieve system-wide optimality. They also developed a rolling horizon approach that provides hourly updates in a manner that reflects current practices in irregular airline operations.

In Eshkevari, Komijan and Baradaran (2023), a new bi-objective aircraft/crew recovery model was formulated to minimize a recovery cost regarding crew swapping, flight cancellation, flight delay, and crew deadhead costs. To solve the model, they adopted the tabu search method for their context ARP, which was able to reach optimal and close-optimal in a short computational time, especially in large-scale problems.

2.2.5 Other methods

Arias et al. (2013) proposed a combined methodology using simulation and optimization techniques to cope with the stochastic aircraft recovery problem (SARP). The approach solves the SARP through the rescheduling of the flight plan using delays, swaps, and cancellations. The main objective of the optimization model is to restore as much as possible the original flight schedule, minimizing the total delay and the number of canceled flights. By applying simulation techniques, the robustness of the given solution is assessed.

Mota, Mota and Serrano (2015) described a methodology for the SARP, which considers the stochastic nature of air transportation systems. The methodology is based on the large neighborhood search (LNS) metaheuristic, combined with a simulation run at different stages to ensure robustness. The proposed method was tested on several instances with different characteristics, some of which were obtained from real data provided by a Spanish airline.

In recent years, a few papers have been published using simulation-based approaches to solve the ARP. Rhodes-Leader et al. (2018a), Rhodes-Leader et al. (2018b) and Rhodes-Leader et al. (2022) combined a symbiotic simulation system, i.e., a simulation approach that matches a high-fidelity simulation model with a low-fidelity physical model for the benefit of both. In their case, the authors propose an adapted version of the ILP model presented by Zhang, Lau and Yu (2015) to reduce the complexity of the solution space considered for the simulation model. The ILP model generates a set of reasonable solutions that are then used as initial solutions in the simulation model to guarantee a faster and more effective high-fidelity simulation system. The difference among the works is that Rhodes-Leader et al. (2018b) extends Rhodes-Leader et al. (2018a) by considering the maintenance schedules, and Rhodes-Leader et al. (2022) by incorporating uncertainties within the decision-making process.

Wang et al. (2019) developed a simulation model based on the dynamic system framework, even as recovery scheduling for two types of disruptions: bad weather and unexpected maintenance. They provided numerical test results on an operation example from China Eastern Airlines. Their simulation model can be used to rapidly test and compare many different recovery schedules obtained from the algorithms embedded in the simulation model and proposed by the schedulers, based on which it can select and propose some good schedules.

A SARP was considered in a recent paper by Lee, Marla and Jacquillat (2020). The authors propose an innovative reactive and proactive approach to solve the SARP. By forecasting systematic delays at hub airports, their study optimizes recovery actions that respond to both realized disruptions and anticipated future disruptions. The authors combine a stochastic queuing model to capture airport congestion, with a commercial flight planning tool, and with a dynamic integer programming framework to model the disruption recovery. A solution based on a look-ahead approximation and sample average approximation is proposed to solve the modeling framework.

2.2.6 Overview

Some features of the ARP considered in the present dissertation 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; 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. There are also papers on the helicopter transportation of oil rig crew members in the context of oil companies (GALVÃO; GUIMARÃES, 1990; MORENO; ARAGÃO; UCHOA, 2006; MENEZES et al., 2010; QIAN; GRIBKOVSKAIA; HALSKAU, 2011; QIAN et al., 2012; QIAN et al., 2015; GRIBKOVSKAIA; HALSKAU; KOVALYOV, 2015), but most of them focus on helicopter routing and passenger allocation decisions or minimizing operation safety risks (i.e., they are not ARPs), as pointed out in Bastos, Fleck and Martinelli (2020). Concerning solution methods, we have developed novel representations such as takeoff event-based and discrete-time formulations, as well as heuristic approaches never seen before in the literature.

Table 2 summarizes the ARP relevant materials according to some main features, including our ARP (see the last line). When comparing it with Table 1, we noticed the inclusion of a new column, called "Network". Each paper is classified by this column in accordance with the type of network used to represent the respective problem, common in the ARP literature (CLAUSEN et al., 2010):

- Connection network (CN): it is an activity-on-node network, where flight legs correspond to nodes in the network, and connections between flight legs are consonant with directed edges (arcs) amongst the nodes. Hence, a node i, representing the flight leg l_i, is connected by a directed edge (i, j) to a node j, which represents the flight leg l_j, if it is feasible to fly l_j immediately after l_i using a same aircraft;

- Time-line network (TLN): a.k.a time-space network, it is an activity on-edge network, where directed edges correspond to activities of an aircraft, and schedule information is represented explicitly by event nodes. This representation has a node for each event, an event being an arrival or a departure of an aircraft at a particular station;

- Time-band network (TBN): it is proposed by Arguello (1997), where the network can be constructed dynamically as disruptions occur, for a certain recovery time period. The idea is to partition the recovery period into time-bands or discrete intervals. Station (airport) activity is then aggregated during each of the resulting time intervals. This aggregation allows us to model the flight connections to which an aircraft may be assigned, and to approximate the costs associated with the possible connections. The output of the transformation is a network positioned on a two-dimensional plane, in which one axis represents time, and the other, space or station location. A node in the transformation network represents the possible activity at a station during a specific segment of time. Because the network is time-based, nodes are placed according to the segment of time they portray. Arcs directed into a node symbolize the arrival of an aircraft during a specific time segment for a scheduled flight. Arcs originating at a node denote specific flights that an available aircraft at the node may service.

Similar to previous section, we illustrate in Figures 4 the number of ARP articles published by journals, countries and years. As the resulting distinct count of journals was 50, we decided to display only the first 15 newspapers in Figure 4b.

Table 2 – Overview and classification for literature focusing aircraft recovery problem. (Continued on next page)

Paper	Network	Objective	Problem Characteristics			Solution		Data	Dimensions		
			Multifleet	Maintenance	Crew	Approach	Type		Aircraft	Fleet	Flights
Teodorović and Guberinić (1984)	CN	Delay time	N	N	N	Heuristic with B&B algorithm	HM	G	3	1	8
Teodorović and Stojković (1990)	CN	Delay and canx	N	N	N	Heuristic based on Dynamic programming	HM	G	14	1	80
Jarrah et al. (1993), Rakshit et al. (1996)	TLN	Delay, swap and ferrying	Y	N	N	Network-flow models and Busacker-Gowen's algorithm	EX	RL		9	
Teodorović and Stojković (1995)	CN	Delay and canx	N	Y	Y	First-in-first-out and Dynamic programming	HM	G		1	80
Mathaisel (1996)	TLN	Revenue loss and operating cost	N	Y	Y	Out-of-Kilter algorithm	EX			1	
Yan and Yang (1996)	TLN	Revenue minus costs	N	N	N	Lagrangian relaxation with Sub-gradient methods	HM	RL	17	1	39
Clarke (1997)	CN	Revenue minus costs	Y	Y	Y	Tree-search heuristic with B&B algorithm	HM	RL	177	4	612
Yan and Tu (1997)	TLN	Revenue minus costs	Y	N	N	Lagrangian relaxation with Sub-gradient methods	HM	RL	273	3	3
Cao and Kanafani (1997a, 1997b)	TLN	Revenue minus costs	N	N	N	0-1 quadratic programming formulation and Active set algorithm	EX	G	162	1	504
Luo and Yu (1997)		Number of delayed flights	N	N	N	ILP formulation and Lexicographic heuristic	EX	RL		1	71
Argüello, Bard and Yu (1997)	TBN	Route and canx costs	Y	N	N	GRASP heuristic	MH	RL	16	1	42
Thengvall, Bard and Yu (2000)	TLN	Revenue minus cost	N	N	N	Single-commodity network-flow model and Rounding heuristic	EX	RL	27	1	162
Thengvall, Yu and Bard (2001, 2003)	TLN/TBN	Revenue minus cost	Y	N	N	Multicommodity network formulations and Bundle algorithm	EX	RL	332	12	2921
Bard, Yu and Arguello (2001)	TBN	Delay and canx	N	N	N	Transformation network method with MIP model	HM	RL	27	1	162
Løve et al. (2002)	CN	Revenue minus delay and canx costs	Y	N	N	Steepest ascent local search heuristic	MH	RL	20-200		80-753
Rosenberger, Johnson and Nemhauser (2003)	CN	Delay and canx	N	Y	N	Aircraft selection with MIP model/CPLEX	EX	G	96	1	407
Andersson and Värbrand (2004)	CN	Canx and fleet swap	Y	N	N	DWD and Lagrangian relaxation-based heuristic	HM	RL	30	5	215
Løve et al. (2005)	TLN	Revenue minus costs	N	N	N	Constructive and Steepest ascent local search heuristics	MH	RL	80	1	340
Andersson (2006)		Canx and fleet swap	Y	N	N	Tabu search and Simulated annealing metaheuristics	MH	RL	30	5	215
Liu et al. (2006, 2008)		Delay, canx and assignment	Y	N	N	Multi-objective genetic algorithm	MH	RL	7		70
Filar et al. (2007)		Delay and canx	Y	N	N	Discrete-time MIP model/CPLEX	EX	RL		4	517
Xiuli and Jinfu (2007)		Ferrying costs	N	N	N	Grey programming method	EX	G	6	1	20
Qiang, Xiao-wei and Jin-fu (2009)		Delay, swap and canx	N	N	N	GRASP and SA heuristic	HM		30	1	149
Tang et al. (2009)	CN	Delay, canx and aircraft flow imbalance costs	Y	N	N	Resource assignment model and Greedy random simulated annealing algorithm	MH	G	30-50		149-200

Table 2 – Overview and classification for literature focusing aircraft recovery problem. (Continued on next page)

Paper	Network	Objective	Problem Characteristics			Solution		Data	Dimensions		
			Multifleet	Maintenance	Crew	Approach	Type		Aircraft	Fleet	Flights
Eggenberg, Salani and Bierlaire (2010)	TBN	Delay, swap, canx, plus maintenance cost	Y	Y	Y	Dynamic programming with CG	HM	RL	100	1	760
Liu, Chen and Chou (2010)	CN	Delay, swap and canx	N	N	N	Hybrid multiobjective genetic algorithm	MH	RL	7	1	84
Jafari and Zegordi (2010, 2011)	CN	Delay, swap, canx and ferrying	Y	N	N	Rolling horizon time framework with MIP model	EX	RL	13	2	100
D'Ariano, Pistelli and Pacciarelli (2012)	CN	Delay time	Y	N	N	Alternative graph models and Tabu search with B&B algorithm	MH	RL	6-32		44-259
Gao et al. (2012)	CN	Delay time	N	N	N	MIP models and optimum polynomial algorithm	EX	G	1	1	8
Wu and Le (2012)	TLN	Canx and assignment	Y	Y	N	Iterative tree-growing with Node combination method	MH	RL	35	5	170
Xiuli and Yanchi (2012)		Canx and assignment	Y	Y	N	GRASP combined with Ant colony heuristic	MH	RL	50	5	
Le, Gao and Zhan (2013)		Delay and AC swap	Y	N	N	Time Window Modelling and Genetic Algorithm	MH	RL	6	3	30
Arias et al. (2013)		Canx and assignment		N	N	Constraint programming with Simulation	O	RL	11		51
Le and Wu (2013)	TLN	Delay, swap and canx	Y	Y	Y	Iterative tree-growing with Node combination method	HM	RL	35	4	170
Akturk, Atamturk and Gurel (2014)	TLN	Delay, swap and ferrying	Y	N	N	Conic quadratic MIP model	EX	RL	60	6	207
Brunner (2014)		Delay and canx		N	Y	ILP model/CPLEX	EX	RL			79
Vos, Santos and Omondi (2015)	TLN	Delay, swap and canx	Y	N	N	Aircraft Selection Heuristic with MIP model	HM	RL	43		
Sousa et al. (2015)	CN	Delay, swap and canx	N	N	N	Dynamic aircraft scheduling with Ant colony optimization	MH	RL	72	1	5722
Zhu, Zhu and Gao (2015)		Delay, swap and canx	N	N	N	Stochastic greedy simulated annealing algorithm	MH	RL	6	1	23
Xu et al. (2015)	TBN	Delay and canx	N	N	N	Time-band approximation with MIP model /CPLEX	EX	G	3	1	11
Mota, Mota and Serrano (2015)		Delay and AC swap	N	N	N	Constraint programming with LNS heuristic and Simulation	O	RL	40	1	163
Xu and Han (2016)	TBN	Flight canx		N	N	Weighted time-band approximation with MIP model	MH	RL	60		254
Wu et al. (2017)	CN	Delay, swap and canx	N	N	N	Distributed fixed-point integer programming	EX	RL	27	1	162
Hu et al. (2017)	CN	Delay, swap and canx	Y	N	N	Neighborhood search algorithm with constraints	MH	RL	104	1	410
Zhang (2017)	CN	Delay, swap and canx	N	Y	N	Two stage heuristic for line of flight reduction	MH	RL	44	1	638
Arıkan, Gürel and Aktürk (2017)	TLN	Delay, swap and ferrying	Y	N	Y	Conic quadratic MIP model	EX	RL		6	1429
Khaled et al. (2018)	TLN	Flight canx and AC swap	N	Y	N	Multi-objective LP with e-constaint for Pareto frontier	MH	RL	10	1	111

Table 2 – Overview and classification for literature focusing aircraft recovery problem. (Continued on next page)

Paper	Network	Objective	Problem Characteristics			Solution		Data	Dimensions		
			Multifleet	Maintenance	Crew	Approach	Type		Aircraft	Fleet	Flights
Šarčević, Rocha and Castro (2018)		Delay, swap and canx		N	N	Artificial Bee Colony algorithm implemented in MASDIMA	MH				
Liang et al. (2018)	CN	Delay, swap, canx, plus maintenance cost	N	Y	N	Column generation-based heuristic	HM	RL	44	1	638
Zhao and Chen (2018)	TLN	Delay, swap and canx		N	N	Weight-table based heuristic algorithm	MH	RL	6		32
Lin and Wang (2018)		Delay and AC swap	Y	N	N	Sequential decision algorithm	MH	RL	151	9	749
Rhodes-Leader et al. (2018a, 2022)	TLN	Delay, swap and canx	N	N	N	High-fidelity simulation with a low-fidelity integer program	O	RL	5-102	1	> 83
Rhodes-Leader et al. (2018b)	TLN	Delay, swap and canx	N	Y	N	High-fidelity simulation with a low-fidelity integer program	O	RL	8	1	54
Erkan, Erkip and Safak (2019)	CN	Delay time	N	N	N	MIP model/CPLEX	EX	RL		1	
Wang et al. (2019)		Delay, canx and ferrying	N	N	N	Simulation-based approach	O	RL	628		5071
Lee, Marla and Jacquillat (2020)	TLN	Canx, assignment and fuel costs	Y	N	N	Dynamic stochastic integer programming framework	O	RL		3	852
Shao et al. (2020)		Delay time and risks probabilities	Y	N	N	Multiobjective mathematical model with Multistart algorithm/Matlab	HM	RL			215
Vink et al. (2020)	TLN	Operating and disruption costs	Y	Y	N	Aircraft selection heuristic with MIP model	HM	RL	100	2	600
Ji et al. (2021)		Delay time	Y	N	N	Rescheduling algorithm with build-in flight feasibility verification routine	MH	RL			300
Hu et al. (2021)	TLN	Airline recovery cost and passenger recovery loss	Y	N	N	Two-objective MIP model and Heuristic with multi-directional/SVNS algorithm	MH	RL	276	11	1038
Sun, Liu and Zhang (2021)	TBN	Operating costs	N	N	N	MIP model/CPLEX and solution algorithm based on TBN modification	EX	RL	80	1	302
Yetimoğlu and Aktürk (2021)	TLN	Airline profit	Y	Y	N	Variable fixing matheuristic algorithm	HM	RL	53	6	208
Evler et al. (2022)	CN	Operating and delay costs	Y	Y	Y	Rolling horizon algorithm with network delay model	MH	RL	17	2	85
Huang et al. (2022)	TLN	Operating costs	Y	Y	N	Iterative cost-driven copy generation approach	HM	RL	4-162		56-789
Lee, Lee and Moon (2022)	TLN	Delay time	Y	N	N	Q-learning and Double Q-learning algorithms with Markov decision process	MH	RL	20	4	70
Liu, Sun and Zhang (2022)	TBN	Delay, swap and canx costs	N	N	N	MIP model/CPLEX	EX	RL	118	1	295
Yan and Chen (2022)	TLN	Operating costs	N	N	N	Dynamic application framework with Network flow model	HM	RL	< 33	1	< 250
Zhao, Bard and Bickel (2023)	TLN	Delay, curfew and canx costs	N	Y	N	Two-stage and Rolling horizon heuristic	HM	RL	73	1	207
Xu, Wandelt and Sun (2023)	CN	Delay, canx, ferrying plus disease infection	Y	N	Y	B&C algorithm and LNS heuristic	MH	G	27-93	2	81-230
Eshkevari, Komijan and Baradaran (2023)	CN	Delay, swap, canx and ferrying costs	Y	Y	Y	Bi-objective MIP model and Tabu search heuristic	HM	G			
Ding et al. (2023)	CN	Delay, canx and fuel costs	Y	Y	Y	DRL guided variable neighborhood search algorithm	MH	G	> 28		16-931

Table 2 – Overview and classification for literature focusing aircraft recovery problem. (Continued)

Paper	Network	Objective	Problem Characteristics			Solution		Data	Dimensions		
			Multifleet	Maintenance	Crew	Approach	Type		Aircraft	Fleet	Flights
Wang et al. (2023)	CN	Delay and canx	Y	N	N	MIP model and Data-driven heuristic method	MH	RL	35		537
Cadarso and Vaze (2023)	TLN	Delay, canx and fuel costs	Y	Y	Y	B&B algorithm with nonlinear cost terms	EX	RL	19	5	1074
Khiabani et al. (2023)		Delay, swap, canx and ferrying costs	Y	N	Y	MIP model and Benders' decomposition algorithm	EX	RL	4-34		15-111
Our ARP	**CN/TLN**	**Delay, AC usage/swap and canx**	**Y**	**N**	**Y**	**MIP formulations and heuristic methods**	**MH**	**RL**	**3-18**	**2-3**	**8-90**

Note: Abbreviations used in the table: Canx: Cancellation; AC: aircraft; CN: connection network; TLN: time-line network; TBN: time-band network; EX: Exact method; MH: (Meta)heuristic; HM: Hybrid method; O: Other methods; Y: Included or mentioned; N: Not included nor considered; " ": Not mentioned or not relevant; G: Generated data; RL: Real-life data.

Source: Own authorship.

2.3 Our publications

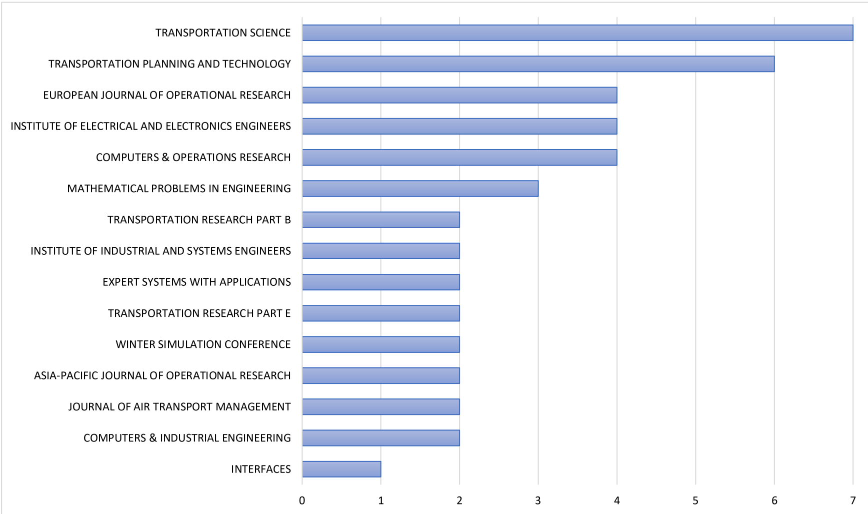
Finally, it is worth mentioning that Tables 1 and 2 do not include works from international conferences and journals that are part of the content produced by this dissertation, as well as those related to which the present author had co-authorship or direct participation. Hitherto, five articles have been written, the first referring to a DAFP, and the remaining four, to different variants of ARP:

- de Campos, Vieira and Munari (2021) (Computational Logistics: 12th International Conference - ICCL 2021): a branch-and-cut solution method was developed to deal with a DAFP that requires the inclusion of a minimum rest time for the crews into each duty period, by dynamically adding cuts/constraints. The strategy used for this is derived on a dynamic programming algorithm (labeling), dedicated to separate cuts that guarantee the feasibility regarding crew legislation. Computational experimentation with real-life data showed that method obtained optimal solutions for all instances in less than five minutes;

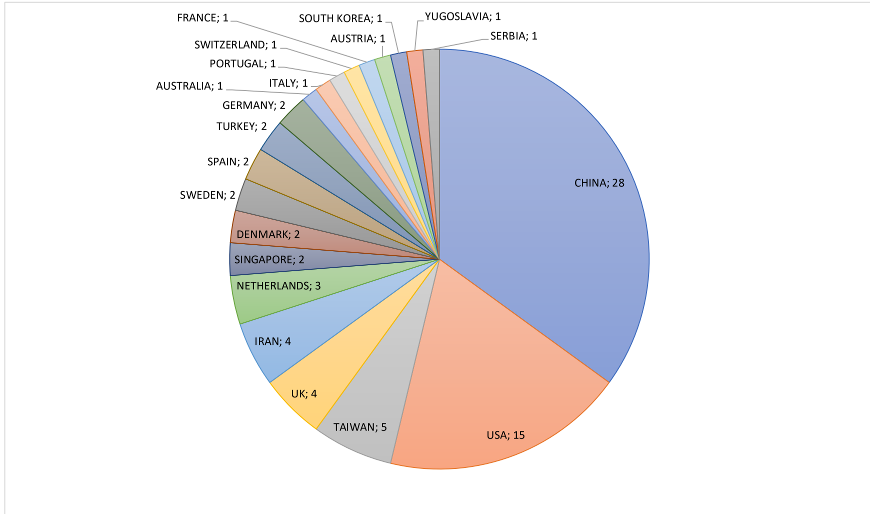
- Vieira et al. (2021) (Transportation Research Part E): their ARP considers an aerodrome with single runway and includes features such as heterogeneous fleet, time windows, safety briefing, minimum aircraft turnaround time, mandatory flight precedence, minimum time interval between consecutive landings on a maritime unit and maximum allowed delays. Two MIP models were proposed to formulate the problem with all relevant characteristics, one based on extension of the traditional network-flow model from HFVRPTW, and other that relies on a novel takeoff event-based representation of the problem. Additionally, the article brought an effective tailored heuristic approach that has constructive and improvement procedures. The results of computational experiments with real-life data provided by an oil company revealed the potential of the proposed approaches to support decision-making, mainly the heuristic, which recovered all pending and current flights without significant delays, taking just a few minutes.

- De La Vega et al. (2022a) (International Transactions in Operational Research): the logistics panorama was expanded by including crew-related constraints such as workday hours and lunch break intervals. The authors proposed a network-flow model with continuous-time as a faithful form of representation, and from this model, they developed a discrete-time simplification and also some simple solution approaches (variants of this second formulation). According to their results, the continuous-time approach is effective for producing full flight recovery only in small-sized problem instances within an hour of runtime. On the other hand, the discrete-time approaches reached solutions with no transferred flights for larger instances, indicating their use in practice.

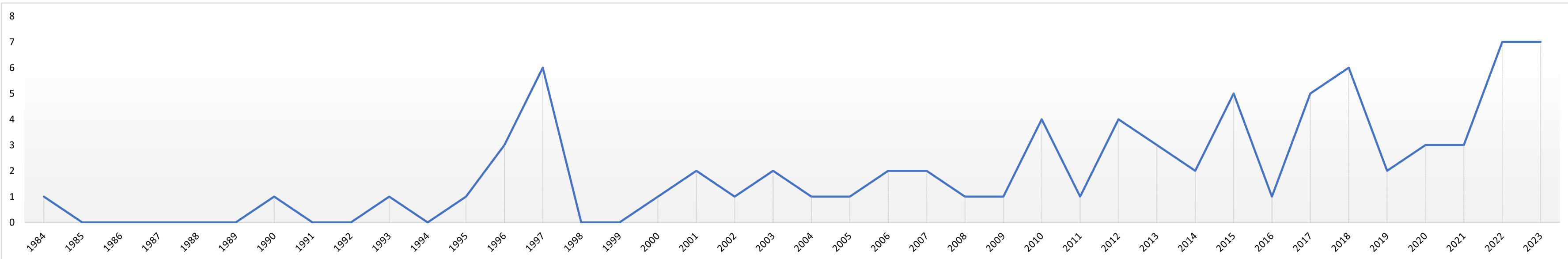
Figure 4 – Number of ARP articles published by journals, countries and over the years.



(a)



(b)



(c)

Source: Own authorship.

- De La Vega et al. (2022b) (EURO Journal on Transportation and Logistics): they extended the problem to the case of multiple aerodromes. In addition to encompassing restrictions from the first two works, this ARP must consider the issue of flight transfers among aerodromes (local-transfers). To solve real-world instances, the previously continuous-time network-flow model from De La Vega et al. (2022a) and tailored heuristic method from Vieira et al. (2021) were extended, and a MIP-based local search heuristic was built. The results obtained by the model using general-purpose optimization software and the employ of heuristics were promising, showing recovery plans with complete flight allocations, few local-transfers, low helicopter usage and small delays.

- Fantazzini et al. (2024) (International Transactions in Operational Research): the ARP addressed in Vieira et al. (2021) is studied from another perspective, in which, the objective function becomes having three non-conflicting hierarchical criteria, lexicographically defined by the penalties: flight cancellations, aircraft usage and departure delays. They develop four different solution approaches using hierarchical goal programming from a discrete-time ILP formulation, aided by enhancements and valid inequalities. Computational experiments with both real-world and simulated instances demonstrated that the application of goal programming significantly reduces the runtime when compared to solving the discrete-time model with single-scaled objective function as a unique 0-1 ILP formulation, consistently delivering optimal or near-optimal solutions within the time limit.

Chapter 3

Aircraft routing with crew assignment for on-demand air transportation: detailed optimization model and MIP-based heuristics

On-demand private flight service is a business in the non-scheduled transportation sector that has significantly grown in the last decades. These operational service types are referred to in the OR literature as dial-a-flight problems (DAFPs) (ENGINEER; NEMHAUSER; SAVELSBERGH, 2011; CAMPBELL; ALI; SILVERWOOD, 2020; CORDEAU et al., 2023). This class of problems represents air taxi transportation activities with fractional ownership aircraft programs, where the operational planning must respond to a demand of travel requests made by customers themselves, considering the state of resources provided at present, which usually happens a few days in advance.

As with other services, an important decision in this problem context is the assignment and scheduling of crew members into flights. Although each country has its own regulations regarding crew labor and rest rules, in the international literature, authors usually follow the FAA guidelines to develop their formulations (SHEBALOV; KLABJAN, 2006; HAOUARI; MANSOUR; SHERALI, 2019). Some of the main requirements are: the maximum duration of a workday (duty); maximum flight time in a duty; minimum rest time between two duties (layover); and minimum work break duration. In spite of being economically advantageous and having a social appeal, few studies integrate aircraft allocation with their respective crews (MARTIN; JONES; KESKINOCAK, 2003; HICKS et al., 2005; YAO et al., 2008; YANG et al., 2008; ZWAN; WILS; GHIJS, 2011).

In this chapter, we address a real-world DAFP motivated by the case of a company that operates in the non-scheduled air transportation sector. We propose a compact mixed-integer programming model that effectively represents the aircraft routing with crew assignment problem faced by the company. This model extends the formulation of Munari and Alvarez (2019), proposed for aircraft routing (without incorporating crew regulation) to minimize operating costs arising from positioning flights and aircraft upgrades. As a way to enforce fundamental and highly relevant crew requirements, this extension contemplates: minimum rest periods, breaks (split duties), maximum flight time, pilot time windows, as well as overtime payments and the possibility of outsourcing for customer requests. Computational experiments using real-world data provided by the company reveal that the proposed model can be effectively solved by a general-purpose MIP solver. To further reduce computing times and improve the quality of solutions for instances not solved to optimality, we developed two well-established MIP-based methods, namely the relax-and-fix and fix-and-optimize heuristics, which are based on the proposed model. The results confirmed the high potential of these approaches for improving decision-making in practice. Thereby, the proposed optimization approaches have the potential to support realistic and complex decisions related to the DAFP in practice, generating significant reductions in operational costs, while taking into account the aviation labor rules.

This chapter consists of the following parts. In Section 3.1, we describe our DAFP. Section 3.2 introduces the formulation of Munari and Alvarez (2019), hereafter referred to as the base model (Subsection 3.2.1), and exhibits the proposed model by including new constraints that ensure crew regulations (Subsection 3.2.2). In Section 3.3, we present our version of relax-and-fix (Subsection 3.3.1) and fix-and-optimize (Subsection 3.3.2) heuristics. Ultimately, the results of computational experiments carried out with the approaches are shown in Section 3.4.

3.1 Problem description

In this logistics panorama, we address a type of service contract in which the contracted company is an airline that has a heterogeneous fleet composed, for example, of private jets, turboprops and helicopters. The company operates in different airports distributed in several European and Asian countries. On the other hand, the contractor is a customer who owns equity part of an aircraft, being responsible for monthly or annual payments to cover outgoings incurred in the operation – such as those related to maintenance, crew salaries, fuel consumption, control and management services, etc. – which are much lower when compared to the costs of owning an aircraft. This contract entitles the customer to a certain amount of mileage for flights in each period.

Unlike traditional airlines, where several operational requirements are already pre-established, this service allows the customer to make a travel request (live leg) and decide: (*i*) the aircraft type, (*ii*) the origin and destination airports, and (*iii*) the departure time of his/her flight. With this, the company must designate an aircraft of the requisitioned type to perform this travel request at the scheduled time. In some circumstances, an aircraft of the required type may not be promptly available at the airport and date desired by the customer. In this situation, it is possible to designate an aircraft of a superior type (upgrade), if available, or take a positioning flight by an aircraft of the same requested type from another airport (a.k.a. ferry leg, deadhead or non-revenue flight). An upgrade incurs additional costs, as the customer is charged according to the aircraft type specified in the request. Nevertheless, an upgrade can be used strategically by the company to obtain savings with respect to positioning a farther away aircraft of the type chosen. Ferry legs do not provide revenue, as the customer only pays for the requested leg. In practice, they represent more than 35% of total travel times for this market segment (YAO et al., 2008).

The company is responsible for the fleet maintenance. Periodically, each aircraft must go through a planned checking and repair process, becoming unavailable until it is finished. Although the start time of a maintenance event is pre-scheduled, the company is typically allowed to advance or delay this time within a relatively large margin of 24 hours, whereas, a customer request can only be delayed up to 15 min (by comparison). Thus, maintenance can be seen as a request in which the aircraft is stationary at a single airport for a certain period and presents a comprehensive time window. It is meaningful to emphasize that we do not need to consider the number of passengers and aircraft capacity in this problem, as the number of passengers is checked at the time a customer makes a request.

In the aircraft itineraries, before each flight event, a minimum turnaround time (generally taking 20 minutes) is required for refueling, cleaning service, embarkation/disembarkation of passengers, among other activities. When a planning horizon starts, programmers also need to take into account the moment that each aircraft becomes available for use, therefore representing an opening time window.

Hence, the design of aircraft routes has to ensure that all requested flights are served and all maintenance events are satisfied, while minimizing the costs related to upgrading and positioning flights. Additionally, it is crucial that each aircraft route has a feasible crew allocation, which adds even more complexity to this planning process. Concerning crew regulation, we consider six operational topics of aviation labor rights:

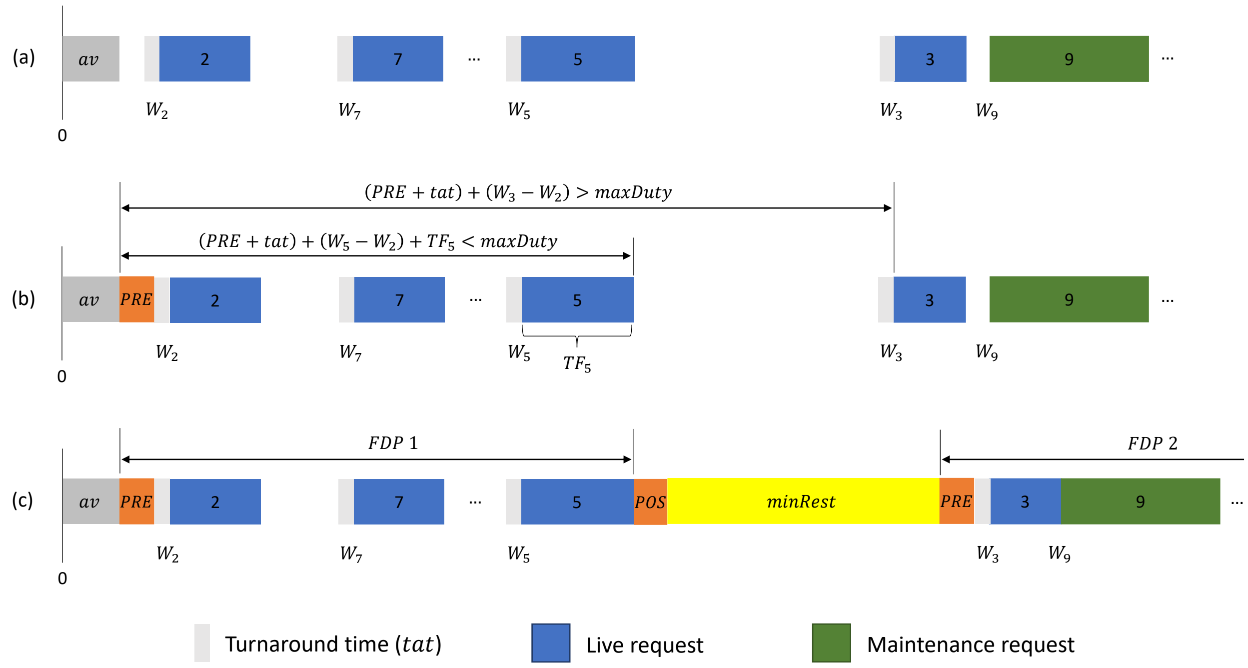
i) Duty and resting times. Being one of the most important crew requirements, the maximum allowed time without rest in a single duty ($maxDuty$) is typically defined as 13 hours in the studied company case. This rule is guaranteed by international labor standards due to the risks associated with fatigue. Moreover, the crew should have at least 10 hours of uninterrupted rest between two consecutive duties ($minRest$). The inclusion of $minRest$ establishes the end of the current duty and the beginning of next one. Other important elements considered by company are the crew presentation times to prepare a plane and analyze the weather conditions and itinerary. The first occurs at the beginning of a duty (PRE), usually taking 40 minutes, being counted within the crew's duty. The second happens at the end of a duty (POS), lasting 30 minutes and is neither counted in the duty nor the rest time (a policy known as Flight Duty Period - FDP). Thus, after each rest period, there is a presentation time in the start and end of the duty.

As an example, Figure 5 depicts the insertion of minimum rest in the schedule of an aircraft. Each Gantt chart in this figure represents a situation, indicated in the left-hand side by (a), (b) and (c). Situation (a) shows the route taken by an aircraft without considering, at this first moment, the insertion of $minRest$, PRE or POS. After the aircraft becomes available (av), it stays on the ground for the minimum turnaround time (the light gray rectangle) and begins by serving live request 2, then 7, and so on (W_r portrays the time that a request r started). In situation (b), we have the same route, but considering PRE after av, resulting in a slight delay in request 2. As displayed, the total time calculated from PRE until the execution of request 5 (where TF_5 is the travel time) is less than $maxDuty$. In turn, if this time is extended to request 3, it ends up exceeding $maxDuty$. Therefore, $POS + minRest + PRE$ must be inserted between requests 5 and 3, establishing the end of the crew's workday (FDP 1) and the beginning of a new one (FDP 2), as illustrated in situation (c). It is worthy mentioning that these rest insertions cannot be performed before the available time for flying av of an aircraft and after its turnaround time tat, as a flight event must occur immediately after these times. Figure 6 exhibits the possibilities for inserting the minimum rest.

ii) Duty breaks. Another rule involves the inclusion of breaks within a duty period. A break is imposed depending on the ground time length (interval in which an aircraft is not used, remaining on the ground between a landing and the next takeoff) that the crew is on standby, i.e., idle waiting for a next flight. According to the crew operation manual of the company, if the ground time is greater than 90 minutes, but less than 6 hours, the time exceeding 90 minutes is to be reduced by 50% (see situation (a) in Figure 7). Likewise, if the ground time is longer than 6 hours, but does not exceed $minRest$, then all this time is counted as one hour (situation (b) illustrated in Figure 7). Both situations correspond to reductions in a duty period, called split duty practices.

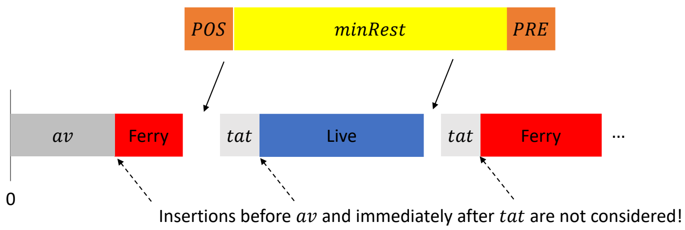
iii) Rest and break during maintenance. A particularity on maintenance requests is that, since the aircraft is parked during an entire repair process, the crew can rest

Figure 5 – Examples of situations for the insertion of minimum rest.



Source: Own authorship.

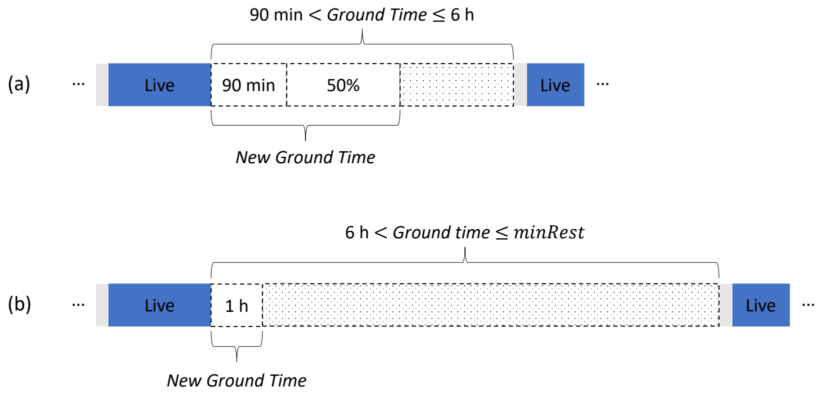
Figure 6 – The possibilities for inserting the minimum rest time.



Source: Own authorship.

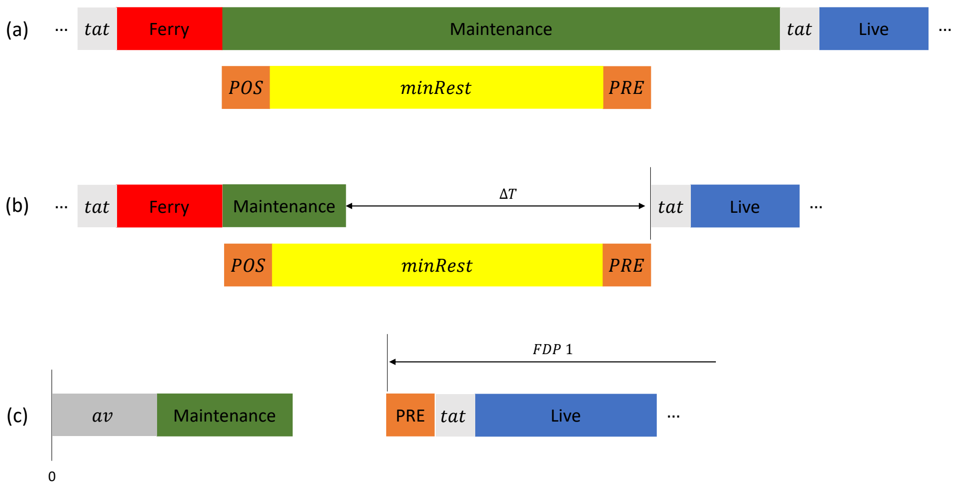
throughout this event (because maintenance events occur in parallel) and, hence, the company can take advantage of the maintenance time. In Figure 8, situation (a) illustrates a case in which the maintenance lasts longer than $POS + minRest + PRE$, thus it is interesting to extend the crew's free time until the end of this event, as there is no reason to keep the pilots on standby if the aircraft is not ready yet. Conversely, in situation (b), the duration of maintenance is less than $POS + minRest + PRE$, thus the crew rests for the minimum required time, even if the aircraft is available earlier. Nevertheless, it is beneficial to perform the maintenance in parallel to part of the rest, as this promotes time-saving and feasibility of solutions that would be impossible otherwise. Furthermore, a crew's workday only begins with a flight event, that way as seen in situation (c), if the aircraft's route starts with a maintenance request, it is not counted in the crew FDP. In this context, the maintenance duration can also be counted as ground time in the split duty.

Figure 7 – Illustration of duty breaks and split duty practices.



Source: Own authorship.

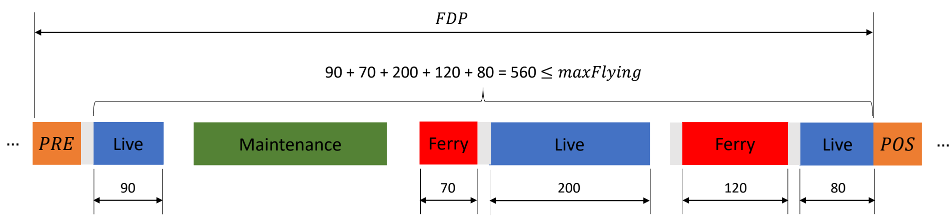
Figure 8 – Resting time in a maintenance request.



Source: Own authorship.

iv) Flying time. Similar to ground times, flight events can directly impact an aircraft's schedule. For each duty period, there is a maximum flight time of 10 hours permitted for the crew activities ($maxflying$). Hence, the duration of live and ferry legs cannot exceed $maxflying$. Figure 9 illustrates, as example, an aircraft with 560 min of total flying time in its FDP.

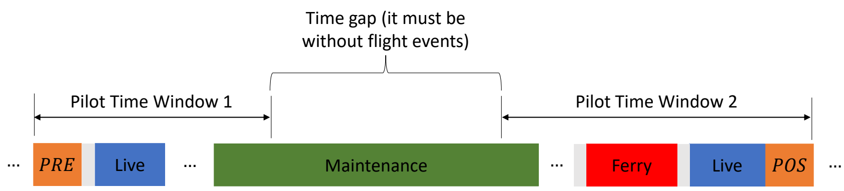
Figure 9 – An example of how flying time is calculated.



Source: Own authorship.

v) Pilot rostering. This crew requirement is related to the time window of a pilot-in-chief and its crew team. In general, every week, there is a rest period of 36 hours including two local nights that closes the pilots' work schedules, named as *H36 Rest Period*. This makes the practice of changing shifts of pilots, which enables happening a gap between subsequent time windows t and $t+1$ for a given aircraft (see Figure 10). In the company, the same time windows are used for the first officer and cabin hostess in the team of the pilot-in-chief.

Figure 10 – Exemplifying an arrangement of two subsequent time windows for pilots at an aircraft.



Source: Own authorship.

vi) Outsourcing and overtime issues. The problem also considers the possibility of outsourcing customer requests to another company, if it is not possible to service them with the current resources. Moreover, overtime pay happens when the time accumulation in duty exceeds $maxDuty$ or $maxflying$. It can be the result of an undesirable allocation of flights, or the existence of a specific flight (live or ferry leg) whose duration naturally already exceeds $maxDuty$ or $maxflying$. Thereby, the overtime costs are used to the detriment of rest time, which corresponds to a percentage increase $overtPerc$ (typically defined as 150%) in the travel cost of an aircraft type.

3.2 Mathematical formulations

To present the two formulations (Subsections 3.2.1 and 3.2.2), firstly, we define the common notation and parameters of our DAFP for both models. This input data comes from a list of available airports (where there are the geographical coordinates of each one), list of aircraft fleet (with their respective types, travel times between any two airports, taking into account calculation from great circle mapper, turnaround times and the initial locations of each aircraft), and a list of requests (required departure times, aircraft types and origin and destination airports):

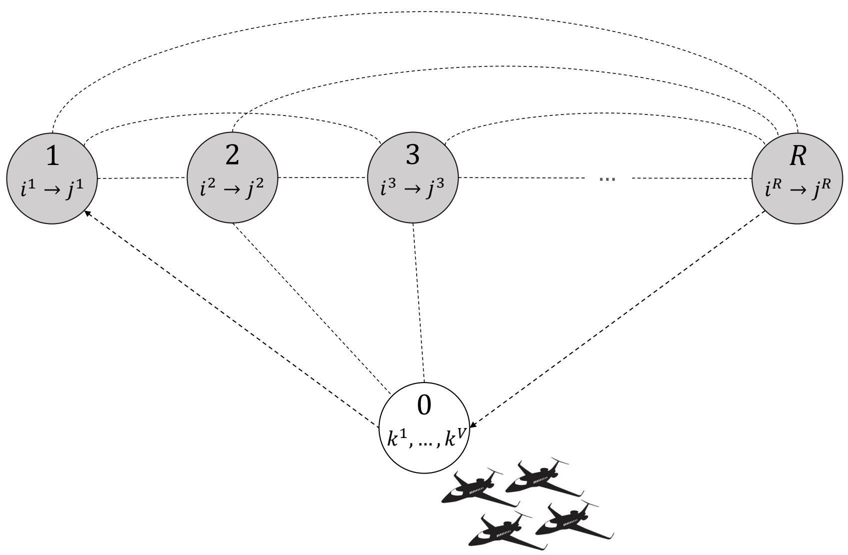
- $\mathcal{K} = \{1, \ldots, K\}$: set of airports in which the company operates;
- $\mathcal{V} = \{1, \ldots, V\}$: set of available aircraft;
- $\mathcal{P} = \{1, \ldots, P\}$: set of aircraft types;

- $\mathcal{L}$: set of customer requests;
- $\mathcal{M}$: set of maintenance requests (hence, $\mathcal{M} \cap \mathcal{L} = \emptyset$);
- $\mathcal{R} = \{0\} \cup \mathcal{L} \cup \mathcal{M}$: set of requests, where 0 is a dummy request used as the first and last request serviced by any aircraft;
- $\mathcal{V}^p$: subset of aircraft corresponding to type p;
- k^v: initial (pre-designated) airport of aircraft v;
- $\check{p}^v$: type of aircraft v;
- $\hat{p}^r$: required type of aircraft in request r;
- i^r, j^r: origin and destination airports, respectively, of the request r;
- c_p: travel cost per time unit of an aircraft of type p, in \$/min;
- TF_{ij}^p: travel time between airports i and j for an aircraft of type p, in minutes;
- av_v: exact time at which aircraft v becomes available to fly for the first time in the planning horizon, in minutes;
- tat_k^r: turnaround time required for an aircraft at airport k before servicing request r, in minutes ($tat_k^r = 0;\ \forall\ r \in \mathcal{M}, k \in \mathcal{K}$);
- st_r: planned starting time of request r, in minutes;
- $\Delta_{\mathcal{L}}$: maximum delay allowed to start servicing any customer request, in minutes;
- v^r: index of the aircraft that must undergo the maintenance request r;
- TL_r: duration of maintenance request $r \in \mathcal{M}$, in minutes;
- $\Delta_{\mathcal{M}}$: maximum tolerance of the anticipation/postponement of a maintenance event, in minutes.

3.2.1 Base model

A peculiarity of the formulation developed in Munari and Alvarez (2019) (which served as the basis for our model extension) is that, instead of using a problem representation based on the traditional network in which airports are nodes and decision variables determine which arcs will be chosen (LETCHFORD; SALAZAR-GONZÁLEZ, 2015; DESAULNIERS; MADSEN; ROPKE, 2014), the authors adopted a type of alternative network where nodes represent requests, as can be seen in Figure 11. The motivation for their choice is that this network typically leads to a more effective mathematical model than other traditional formulations as the number of requests is significantly smaller than the number of available airports. In this representation, the aircraft flow is defined through requests, and each node has to be visited only once by just one plane. Consequently, the visit to a node r indicates that the aircraft serves request r and thus visits the related origin and destination airports (i^r and j^r). If the aircraft goes from a node (request) r to another node s, such that $j^r \neq i^s$, i.e., the destination airport of r is not the same as the origin airport of s, then a ferry leg is implied between these visits. All routes must start and finish at the dummy node (request) 0. This means that each aircraft v departs from its initial airport k^v before serving the first request, and at the end of its route, it stays at the destination airport of the last request that immediately precedes the dummy node 0. The problem is then to find routes in this network, so that all nodes (requests) are visited (served) exactly once, reducing the occurrence of positioning flight displacements.

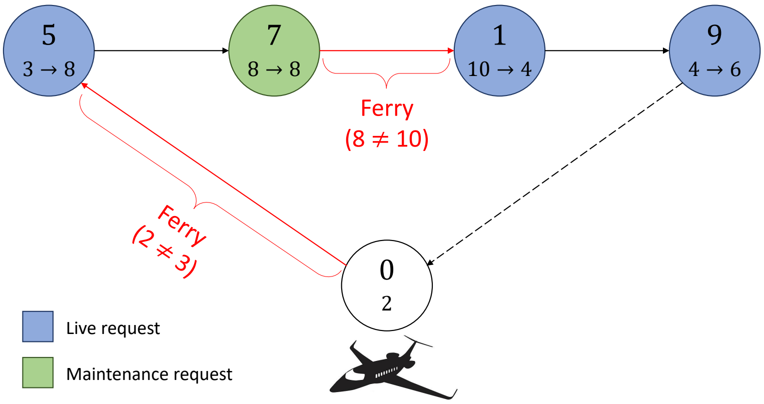
Figure 11 – Problem representation using a network of requests.



Source: Adapted from Munari and Alvarez (2019).

To illustrate the occurrences of ferries on a route, consider Figure 12. In this path, an aircraft departs from airport 2 (see the initial airport within dummy node 0) and has as its first request, the 5, which takes as its origin airport, the 3. As these two airports are different ($2 \neq 3$), there is a necessity to execute a positioning flight, that is, a trip without passengers among these airports, so that the aircraft is available to execute the live request 5. After request 5 (flight from airport 3 to 8, as indicated in the node), the next node to be visited is a maintenance request (7), which has the same airport 8 (to ensure a uniform notation, we assume that for $r \in \mathcal{M}$, the airports i^r and j^r are equal and denote the airport in which the maintenance shall happen). In this way, there is no ferry and the aircraft is just undergoing maintenance at airport 8. However, in the next request (1), the origin airport is the 10, hence we have a ferry leg to then carry out request 1 (flight between airports 10 and 4). Finally, as the subsequent request to 1 (9) has the same airport, as soon as the aircraft fulfills request 1, it is already to serve request 9 (i.e., there is no ferry). At the end of the route, the aircraft will be at airport 6, which implies that request 9 precedes dummy request 0 in the graph.

Figure 12 – Exemplifying the occurrence of ferries.



Source: Own authorship.

The described network requires a proper definition of costs. Let Cf^v_{rs} be the positioning cost, which corresponds to the flying cost when aircraft v flies without customers between requests r and s, defined as follows:

$$Cf^v_{rs} = \begin{cases} c_{\check{p}^v}.TF^{\check{p}^v}_{k^v i^s}, & \text{if } r = 0 \wedge s > 0 \wedge k^v \neq i^s; \\ c_{\check{p}^v}.TF^{\check{p}^v}_{j^r i^s}, & \text{else if } r > 0 \wedge s > 0 \wedge r \neq s \wedge j^r \neq i^s; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The upgrade cost, taken as Cup^v_r, is determined by choosing an aircraft v with type better (superior) than the one contracted in request r, calculated as:

$$Cup^v_r = \begin{cases} c_{\check{p}^v}.TF^{\check{p}^v}_{i^r j^r} - c_{\hat{p}^r}.TF^{\hat{p}^r}_{i^r j^r}, & \text{if } \check{p}^v > \hat{p}^r; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

To simplify the notation, we assume that set $\mathcal{P}$ follows a non-descending order regarding the quality of aircraft types. Therefore, an aircraft of type p is inferior than or similar to an aircraft of type $p + 1$.

Eventually, from the flow-network illustrated in Figure 11, the following decision variables are defined:

- $y^v_{rs} = \begin{cases} 1, & \text{if aircraft } v \text{ services requests } r \text{ and } s\text{, consecutively;} \\ 0, & \text{otherwise.} \end{cases}$
- W_r: exact time at which request r is serviced, in minutes.

Using the parameters and notation just defined, the compact formulation (called base model) is then formalized by:

$$\min \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}}\sum_{\substack{s\in\mathcal{R}:\\ r\neq s}} Cf^v_{rs}.y^v_{rs} + \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{L}}\sum_{\substack{s\in\mathcal{R}:\\ r\neq s}} Cup^v_r.y^v_{rs}; \quad (3)$$

s.t.

$$\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}}\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^v_{rs} = 1;\ \forall\ r\in\mathcal{L}; \quad (4)$$

$$\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^v_{rs} = 0;\ \forall\ v\in\mathcal{V}, r\in\mathcal{L}\mid \check{p}^v<\hat{p}^r; \quad (5)$$

$$\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^{v^r}_{rs} = 1;\ \forall\ r\in\mathcal{M}; \quad (6)$$

$$\sum_{\substack{v\in\mathcal{V}:\\ v\neq v^r}}\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^v_{rs} = 0;\ \forall\ r\in\mathcal{M}; \quad (7)$$

$$\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^v_{sr} = \sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y^v_{rs}; \forall\ v\in\mathcal{V}, r\in\mathcal{L}\cup\mathcal{M}; \quad (8)$$

$$\sum_{s\in\mathcal{R}} y^v_{0s} = \sum_{r\in\mathcal{R}} y^v_{r0} = 1;\ \forall\ v\in\mathcal{V}; \quad (9)$$

$$st_r\leq W_r\leq st_r+\Delta_{\mathcal{L}};\ \forall\ r\in\mathcal{L}; \quad (10)$$

$$W_s\geq W_r+\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}}(TF^{\check{p}^v}_{i^rj^r}+tat^s_{j^r}+TF^{\check{p}^v}_{j^ri^s}+tat^s_{i^s}).y^v_{rs}$$
$$-M^1_r.(1-\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}} y^v_{rs});\ \forall\ r\in\mathcal{L},\ s\in\mathcal{L}\cup\mathcal{M}\mid r\neq s\wedge j^r\neq i^s; \quad (11)$$

$$W_s\geq W_r+\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}}(TF^{\check{p}^v}_{i^rj^r}+tat^s_{i^s}).y^v_{rs}$$
$$-M^1_r.(1-\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}} y^v_{rs});\ \forall\ r\in\mathcal{L},\ s\in\mathcal{L}\cup\mathcal{M}\mid r\neq s\wedge j^r= i^s; \quad (12)$$

$$W_s\geq(av_v+TF^{\check{p}^v}_{k^vi^s}+tat^s_{i^s}).y^v_{0s}\ \forall\ v\in\mathcal{V},\ s\in\mathcal{L}\mid k^v\neq i^s; \quad (13)$$

$$W_s\geq(av_v+tat^s_{i^s}).y^v_{0s};\ \forall\ v\in\mathcal{V}, s\in\mathcal{L}\mid k^v=i^s; \quad (14)$$

$$st_r-\Delta_{\mathcal{M}}\leq W_r\leq st_r+\Delta_{\mathcal{M}};\ \forall\ r\in\mathcal{M}; \quad (15)$$

$$W_s\geq W_r+(TL_r+TF^{\hat{p}^r}_{j^ri^s}+tat^s_{i^s}).y^{v^r}_{rs}$$
$$-M^2_r.(1-y^{v^r}_{rs});\ \forall\ r\in\mathcal{M},\ s\in\mathcal{L}\cup\mathcal{M}\mid r\neq s\wedge j^r\neq i^s; \quad (16)$$

$$W_s\geq W_r+(TL_r+tat^s_{i^s}).y^{v^r}_{rs}$$
$$-M^2_r.(1-y^{v^r}_{rs});\ \forall\ r\in\mathcal{M},\ s\in\mathcal{L}\cup\mathcal{M}\mid r\neq s\wedge j^r= i^s; \quad (17)$$

$$W_s\geq(av_{v^s}+TF^{\hat{p}^s}_{k^{v^s}i^s}).y^{v^s}_{0s};\ \forall\ s\in\mathcal{M}\mid k^{v^s}\neq i^s; \quad (18)$$

$$W_s\geq av_{v^s}.y^{v^s}_{0s};\ \forall\ s\in\mathcal{M}\mid k^{v^s}=i^s; \quad (19)$$

$$y^v_{rs}\in\{0,1\};\ \forall\ v\in\mathcal{V};\ r,s\in\mathcal{R}; \quad (20)$$

$$W_r\geq 0;\ \forall\ r\in\mathcal{L}\cup\mathcal{M}; \quad (21)$$

where $M^1_r = st_r + \Delta_{\mathcal{L}}, \ \forall \ r \in \mathcal{L}$ and $M^2_r = st_r + \Delta_{\mathcal{M}}, \ \forall \ r \in \mathcal{M}$.

The objective function (3) aims at minimizing the operational expenses. They are composed by the costs of aircraft positioning, which arise in trips that aircraft fly without customers, represented by the first term; and upgrade cost, the increase in cost due to the assignment of a better aircraft type to a live leg, represented by the second term.

Routing constraints are given by (4)-(9). Constraints (4) ensure that each request $r \in \mathcal{L}$ is serviced by an aircraft of requested type $\hat{p}^r$ or higher ($\check{p}^v \geq \hat{p}^r$). On the other hand, constraints (5) prevent downgrading ($\check{p}^v < \hat{p}^r$). Fulfillment of maintenance request $r \in \mathcal{M}$ for specified aircraft v^r is guaranteed by constraints (6) and (7)[1]. While constraints (8) ensure the flow conservation of aircraft through the network of requests, constraints (9) enforce the balance in dummy request, where every aircraft must depart of and return from.

The schedule of aircraft is modeled by the family of constraints (10)-(19), which can be separated into two groups: referring to customer requests (10)-(14) and maintenance requests (15)-(19). In the first group, the time windows for customer requests are assured by constraints (10). The minimum time to start a customer request s is obliged by constraints (11) and (12). The first set is activated when positioning is needed to service request s after r ($j^r \neq i^s$) and thereby, it includes this additional travel time ($tat^s_{j^r} + T^{\check{p}^v}_{j^r i^s}$). The second is applied when the destination of request r is the same as the departure airport of s, therefore, being only necessary to compute the travel time of request r ($T^{\check{p}^v}_{i^r j^r}$) and the turnaround time before request s ($tat^s_{i^s}$). Constraints (13) and (14) impose the time that each aircraft will be ready to service the first request s at the beginning of planning horizon, with the first set of constraints being used when a positioning flight is necessitated between the aircraft's starting airport (k^v) and the request's departure airport (i^s), and the second one is employed when these airports are identical. Constraints (10)-(14) are analogous to (15)-(19), but for maintenance requests. The main difference between this type of request and the customer requests is that, instead of requiring a flight from i^r to j^r for executing the request ($T^{\check{p}^v}_{i^r j^r}$), the aircraft must stay on the ground at airport i^r during the whole duration (TL_r) of the maintenance process.

At last, the domain of the decision variables is defined in (20) and (21).

[1] At first glance, constraints (5) and (7) appear redundant, however, they are mandatory. In (4) and (6), the sums do not cover all requests and aircraft. Unlike traditional VRP variants, in which all arcs are costed, in this problem, there is only cost for arcs with ferry leg. If (5) and (7) were omitted, we could have a garbage-out/infeasible solution, since sub-cycles with requests already assigned to other aircraft may occur, and consequently, the objective function value would be lower (incorrect), because such a relaxed formulation permit downgrading or unauthorized request assignment in an aircraft's route (note that (8) consider all requests and aircraft), for having a cheaper positioning cost. An alternative to omitting these constraints would be to only declare the decision variables that satisfy the upgrade and aircraft pre-assignment conditions. Nevertheless, this would aggravate the proposed model's size (shown ahead), as its constraints would have to be further partitioned, otherwise, we would witness a memory segmentation fault in the model's implementation.

3.2.2 Proposed model

As a way to faithfully represent the specifics of our DAFP, that is, in the context of on-demand air transportation, take into account characteristics of distinct aircraft and their different usage costs, as well as comply with aspects of legislation relevant to the inclusion of crews in the operation, we had to elaborate a very detailed formulation with a considerable number of constraints and variables. This makes the modeling presentation non-trivial. For ease of understanding, in addition to grouping the constraints concerning the six operational topics about crew regulation (as portrayed in the problem description by Section 3.1), further partitioned into subsections 3.2.2.1-3.2.2.6, and introducing the parameters and decision variables within the topics (specifically, at the beginning of each one) that the families of constraints are explained, we chose to bring together the required notation again as a complement (also divided into the topics by tables) in Appendix A.1, which can function as a glossary. We believe that this organization may expose the model in a more didactic way, facilitating the notation query.

It is important to highlight that these additional crew constraints do not compromise the model performance as it can be effectively used within general-purpose MIP solvers to obtain optimal or close-to-optimality solutions to real-world instances in reasonable computing times.

3.2.2.1 Crew rest rules

To incorporate the minimum rest rule into the base model, we compute the accumulated duty time considering the difference between time instants of consecutive requests r and s (i.e., $W_s - W_r$) as the aircraft advances on its route. However, to account for the residual time at the end of the current duty and at the beginning of the next one, we need to define binary variables for pointing out the moment when crew members start their rests, and continuous variables for counting the residual time among duties. To model this, we define the following decision variables:

- $U_s \geq 0$: accumulated work time up to request s since the inclusion of the last rest;
- $E1_{rs} \in \{0, 1\}$: 1, if there is a rest before a ferry leg between requests r and s; 0, otherwise;
- $E2_{rs} \in \{0, 1\}$: 1, if there is a rest after a ferry leg between requests r and s; 0, otherwise;
- $E12_{rs} \in \{0, 1\}$: 1, if there is a rest before and after a ferry leg between requests r and s; 0, otherwise;
- $E_{rs} \in \{0, 1\}$: 1, if there is a rest (regardless of a positioning flight) between requests r and s; 0, otherwise;

- $E0_{vr} \in \{0,1\}$: 1, if there is a rest after a ferry leg between the initial airport of aircraft v and the origin airport of request $r \in \mathcal{L}$; 0, otherwise;
- $lengthCurrR1_r \geq 0$: duty time without ferry leg added at the end of the current duty by request r;
- $lengthCurrR2_r \geq 0$: duty time with ferry leg added at the end of the current duty by request r;
- $lengthNextR1_s \geq 0$: duty time with ferry leg to be added at the beginning of the next duty by request s;
- $lengthNextR2_s \geq 0$: duty time without ferry leg to be added at the beginning of the next duty by request s;
- $lengthAmongR12_r \geq 0$: ferry time between the end and start of consecutive duties by request r;
- $RestM_s \geq 0$: rest time to be included between a maintenance request and a live or ferry leg represented by request s. This variable takes into account the use of subsequent maintenance events (which will be explained further in Subsection 3.2.2.3).

Constraints (22)-(35) concern the aircraft scheduling considering crew regulation. They correspond to constraints (10)-(19) of the base model, partitioned in more special cases due to the different ways of including the rest when taking advantage of the maintenance duration. Nevertheless, the total number of constraints remains the same (i.e., for all $r, s \in \mathcal{L} \cup \mathcal{M}$). Constraints (23) ensure that the binary variables $E1_{rs}$ and $E2_{rs}$ enforce the minimum rest time if a duty ends/starts between requests $r, s \in \mathcal{L}$ with different airports. Notably, activating $E1_{rs}$ inserts the full minimum rest ($POS + minRest + PRE$) before the ferry, and activating $E2_{rs}$ inserts the full minimum rest after the ferry. As there is no positioning flight in constraints (24), the inclusion of minimum rest is well-defined using only variable E_{rs}. In constraints (25), for all $r \in \mathcal{L}$ and $s \in \mathcal{M}$, there is the insertion of rest before a ferry only when $E1_{rs} = 1$. This enables taking advantage of ground time from the end of a request with ferry until the start of a maintenance request, which can be used later by also considering TL_s in a subsequent request at s. This way, for any constraint in block (23)-(28) that contains $s \in \mathcal{M}$, there is the possibility of saving its ground time to be used later. Because the airports are the same in constraints (26), the accumulation of ground time is quantified by the time contained between the end of a live leg and the beginning of a maintenance event. We clarify this in the example provided in Figure 13. Constraints (27) and (28) impose the exact time from which the services in a first request $s \in \mathcal{L}$ of an aircraft v can start. Since a positioning flight in constraints (27) requires a crew on board, the minimum rest is activated by the variable $E0_{vs}$. For each request $r \in \mathcal{M}$ with a flight event (whether ferry or live leg to $s \in \mathcal{L}$), $RestM_s$ represents the rest time to be included between a maintenance request and a live or ferry leg, considering the previous maintenance durations and the current TL_r itself. In the circumstance of positioning flight in constraints (30), as request $s \in \mathcal{L}$, there is no opportunity for ground time accumulation if the rest is assigned after the ferry, portraying the placement of $(POS + minRest + PRE).E2_{rs}$. Concerning constraints (32) and (33), because $s \in \mathcal{M}$, this accumulation is possible.

$$st_r.(1 - out_r) \leq W_r \leq (st_r + \Delta_{\mathcal{L}}).(1 - out_r);\;\; \forall\; r \in \mathcal{L}; \tag{22}$$

$$\begin{aligned} W_s \geq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} &(TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}).y^v_{rs} \\ &+ (POS + minRest + PRE).(E1_{rs} + E2_{rs}) \\ &- M^1_r.(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\;\; \forall\; r, s \in \mathcal{L} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{23}$$

$$\begin{aligned} W_s \geq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} &(TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}).y^v_{rs} + (POS + minRest + PRE).E_{rs} \\ &- M^1_r.(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\;\; \forall\; r, s \in \mathcal{L} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{24}$$

$$\begin{aligned} W_s \geq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} &(TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ &+ (POS + minRest + PRE).E1_{rs} \\ &- M^1_r.(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\;\; \forall\; r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{25}$$

$$W_s \geq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} - M^1_r.(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\;\; \forall\; r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \tag{26}$$

$$\begin{aligned} W_s \geq (av_v + PRE + TF^{\check{p}^v}_{k^v i^s} + tat^s_{i^s}).y^v_{0s} \\ + (POS + minRest + PRE).E0_{vs};\;\; \forall\; v \in \mathcal{V}, s \in \mathcal{L} \mid k^v \neq i^s; \end{aligned} \tag{27}$$

$$W_s \geq (av_v + PRE + tat^s_{i^s}).y^v_{0s};\;\; \forall\;\; v \in \mathcal{V}, s \in \mathcal{L} \mid k^v = i^s; \tag{28}$$

$$st_r - \Delta_{\mathcal{M}} \leq W_r \leq st_r + \Delta_{\mathcal{M}};\;\; \forall\;\; r \in \mathcal{M}; \tag{29}$$

$$\begin{aligned} W_s \geq W_r + (TL_r + TF^{\hat{p}^r}_{j^r i^s} + tat^s_{i^s}).y^{v^r}_{rs} + RestM_s \\ + (POS + minRest + PRE).E2_{rs} \\ - (M^2_r + M^3_s).(1 - y^{v^r}_{rs});\;\; \forall\; r \in \mathcal{M}, s \in \mathcal{L} \mid j^r \neq i^s; \end{aligned} \tag{30}$$

$$\begin{aligned} W_s \geq W_r + (TL_r + tat^s_{i^s}).y^{v^r}_{rs} + RestM_s \\ - (M^2_r + M^3_s).(1 - y^{v^r}_{rs});\;\; \forall\; r \in \mathcal{M}, s \in \mathcal{L} \mid j^r = i^s; \end{aligned} \tag{31}$$

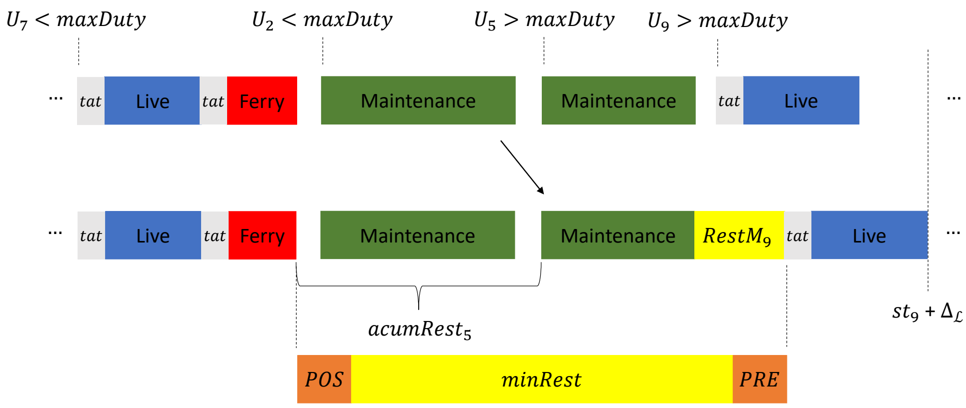
$$\begin{aligned} W_s \geq W_r + (TL_r + TF^{\hat{p}^r}_{j^r i^s}).y^{v^r}_{rs} + RestM_s \\ - (M^2_r + M^3_s).(1 - y^{v^r}_{rs});\;\; \forall\; r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{32}$$

$$W_s \geq W_r + TL_r.y^{v^r}_{rs} - M^2_r.(1 - y^{v^r}_{rs});\;\; \forall\;\; r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{33}$$

$$W_s \geq (av_{v^s} + PRE + TF^{\hat{p}^s}_{k^{v^s} i^s}).y^{v^s}_{0s};\;\; \forall\;\; s \in \mathcal{M} \mid k^{v^s} \neq i^s; \tag{34}$$

$$W_s \geq av_{v^s}.y^{v^s}_{0s};\;\; \forall\;\; s \in \mathcal{M} \mid k^{v^s} = i^s. \tag{35}$$

Figure 13 – An example of using maintenance requests to include the required rest time.



Source: Own authorship.

Constraints (36)-(72) determine the accumulated work time used to enforce rest events in the schedules. As previously defined, the continuous variable U_s represents the accumulated time on duty until request s. The value of U_s is reset every time there is a change of duty shifts. To do this, first, we define how the variables related to rest are activated. Constraints (36) and (37) allow only one of binary variables E_{rs} (rest condition), $\sum_{f=1}^{4} B_{rs}^{f}$ (break condition) or DA_{rs} (for $s \in \mathcal{M}$, expressing a ground time accumulation) to have value of 1 when there is a route between the requests r and s (variables B_{rs}^{f} and DA_{rs} will be better explained in Subsections 3.2.2.2 and 3.2.2.3). Constraints (38) and (39) ensure that only a single request $r \in \mathcal{L}$ is associated with exactly one aircraft $v \in \mathcal{V}$, when $E0_{vs} = 1$ and $y_{0s}^{v} = 1$ (i.e., there is a rest between dummy request 0 and request s). Constraints (40) impose $E_{rs} = 1$, if $E1_{rs} = 1$ and/or $E2_{rs} = 1$, while (41) and (42) enforce $E12_{rs} = 1$ when both variables $E1_{rs}$ and $E2_{rs}$ has a value of 1, depicting the inclusion of rest before and after on positioning flight between requests r and s.

$$E_{rs} + \sum_{f=1}^{4} B_{rs}^{f} \leq \sum_{v\in\mathcal{V}} y_{rs}^{v};\ \ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{L} \mid r \neq s; \tag{36}$$

$$E_{rs} + DA_{rs} + \sum_{f=1}^{4} B_{rs}^{f} \leq \sum_{v\in\mathcal{V}} y_{rs}^{v};\ \ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s; \tag{37}$$

$$\sum_{v\in\mathcal{V}} E0_{vs} \leq \sum_{v\in\mathcal{V}} y_{0s}^{v};\ \ \forall\ s \in \mathcal{L}; \tag{38}$$

$$\sum_{s\in\mathcal{L}} E0_{vs} \leq \sum_{s\in\mathcal{L}} y_{0s}^{v};\ \ \forall\ v \in \mathcal{V}; \tag{39}$$

$$E_{rs} \leq E1_{rs} + E2_{rs} \leq 2.E_{rs};\ \ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \tag{40}$$

$$E1_{rs} + E2_{rs} \geq 2.E12_{rs};\ \ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \tag{41}$$

$$E12_{rs} \geq E1_{rs} + E2_{rs} - 1;\ \ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s. \tag{42}$$

The calculation of the accumulated work time is guaranteed by constraints (43)-(53). For preceding requests $r \in \mathcal{L} \cup \mathcal{M}$ and $s \in \mathcal{L}$, in constraints (47), the accumulated work time is obtained by the difference between the instants $W_s - W_r$, which in addition to the travel times, also consider the idleness corresponding to the waiting time of crew in airports. If $\sum_{v \in \mathcal{V}} y^v_{rs} = 1$, and $E_{rs} = 1$ or $\sum_{f=2}^{3} B^f_{rs} = 1$ (time ranges in which there is a change in ground time, in order to reduce the duty), these constraints remain redundant, admitting that U_s can be reset by another constraint (as discussed later). Constraints (48) are similar to (47) except for the use of terms $s \in \mathcal{M}$ and $Duty_{rs}$ instead of $s \in \mathcal{L}$ and $W_s - W_r$. In the situation where we take advantage of the maintenance durations for crew breaks (viable when request $s \in \mathcal{M}$, and $\sum_{v \in \mathcal{V}} y^v_{rs} = 1$, $E_{rs} = 0$ or $\sum_{f=2}^{3} B^f_{rs} = 0$), making variable $Duty_{rs} = 0$, admitting that $U_s = U_r$. This behavior guarantees that the value of U_r is maintained throughout the accumulation of ground time, so that it can be used later on, when some $B^2_{rs} = 1$ or $B^3_{rs} = 1$ (this will be better detailed in the split duty constraints, in Subsection 3.2.2.2). When a request s is executed right after a dummy ($y^v_{0s} = 1$), constraints (43)-(46) are applied to determine the initial value of time in U_s, where constraints (43) employ the same reasoning used for calculating the rest in (47) and (48).

Given the redundancy condition of constraints (43), (47) and (48) by the activations of E_{rs}, B^2_{rs} or B^3_{rs}, and $E0_{vs}$, respectively, U_s can be reset by one of constraints (49) or (50), where, specifically, determine the work time that variable U_s must contain when starting a new duty from request s. For the dummy request, constraints (49) ensure that U_s is simply the crew presentation time. However, between requests $r, s \in \mathcal{L} \cup \mathcal{M}$, this time depends on whether the inclusion of rest is made in the absence of a positioning event, or even before and/or after the occurrence of a positioning flight. For this reason, we put the variables $lengthNextR1_r$ and $lengthNextR2_r$ in (50), where the first variable takes value as a function of $E1_{rs} = 1$, and the second by $E2_{rs} = 1$.

Another determination is how much work time precisely should be left at the end of a current duty. Thereby, variables $lengthCurrR1_r$ and $lengthCurrR2_r$ are responsible for quantifying this residual work and so they are in constraints (51), aiming to verify whether the accumulated work time at the end of a duty by request r ($U_r + lengthCurrR1_r + lengthCurrR2_r$) exceeds $maxDuty$. If the maximum allowed time without rest is extrapolated, variable $overR_r$ quantifies this work excess, thus generating a part of the overtime costs in (189). There is also a determination for the current duty among requests, which is aimed at activating variable $E12_{rs}$. When the rest time is entered before and after a positioning flight, it means that the travel time value related to the positioning itself must also be checked against $maxDuty$. Variable $lengthAmongR12_r$ assigns the value of ferry leg and the constraints (52) confer whether $lengthAmongR12_r$ surpasses $maxDuty$, valuing variable $overF_r$ in the overtime situation. Finally, the one last determination for the current duty period is about a ferry in the middle of dummy request and first request s executed by aircraft v. This is a specific situation, but relevant to the crew. Consider that $PRE + TF^{\check{p}^v}_{k^v i^s} > maxDuty$ for a route between 0 and s. Even if $E0_{vs} = 1$, which causes U_s to reset (making it small), this ferry could exceed $maxDuty$ without paying overtime. Then, we put constraints (53) to measure this surplus time ($over0_v$), which is penalized in the objective function (189).

$$\begin{aligned} U_s \geq\ & (PRE + TF^{\check{p}^v}_{k^v i^s} + tat^s_{i^s}).y^v_{0s} \\ & - M^3_s.(1 + E0_{vs} - y^v_{0s});\ \forall\ v \in \mathcal{V}, s \in \mathcal{L} \mid k^v \neq i^s; \end{aligned} \tag{43}$$

$$U_s \geq (PRE + tat^s_{i^s}).y^v_{0s};\ \forall\ v \in \mathcal{V}, s \in \mathcal{L} \mid k^v = i^s; \tag{44}$$

$$U_s \geq (PRE + TF^{\hat{p}^s}_{k^{v^s} i^s}).y^{v^s}_{0s};\ \forall\ s \in \mathcal{M} \mid k^{v^s} \neq i^s; \tag{45}$$

$$U_s \geq PRE.y^{v^s}_{0s};\ \forall\ s \in \mathcal{M} \mid k^{v^s} = i^s; \tag{46}$$

$$\begin{aligned} U_s \geq\ & U_r + (W_s - W_r) \\ & - (M^3_r + M^3_s).(1 + E_{rs} + \sum_{f=2}^{3} B^f_{rs} - \sum_{v \in \mathcal{V}} y^v_{rs});\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{L} \mid r \neq s; \end{aligned} \tag{47}$$

$$\begin{aligned} U_s \geq\ & U_r + Duty_{rs} \\ & - (M^3_r + M^3_s).(1 + E_{rs} + \sum_{f=2}^{3} B^f_{rs} - \sum_{v \in \mathcal{V}} y^v_{rs});\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s; \end{aligned} \tag{48}$$

$$U_s \geq (PRE + tat^s_{i^s}).\sum_{v \in \mathcal{V}} E0_{vs};\ \forall\ s \in \mathcal{L}; \tag{49}$$

$$\begin{aligned} U_s \geq\ & lengthNextR1_r + lengthNextR2_r \\ & - M^4_r.(1 - E_{rs});\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \end{aligned} \tag{50}$$

$$U_r + lengthCurrR1_r + lengthCurrR2_r \leq maxDuty + overR_r;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{51}$$

$$lengthAmongR12_r \leq maxDuty + overF_r;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{52}$$

$$over0_v \geq \sum_{s \in \mathcal{L}} (PRE + TF^{\check{p}^v}_{k^v i^s}).E0_{vs} - maxDuty;\ \forall\ v \in \mathcal{V}. \tag{53}$$

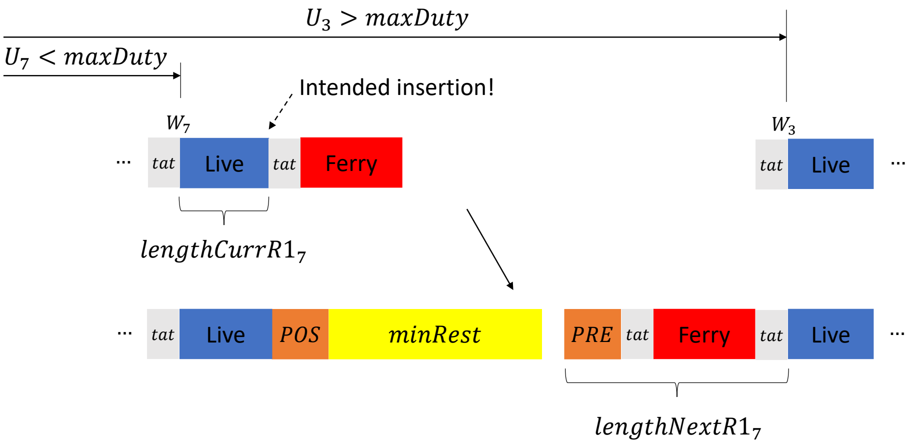
The three ways to insert *minRest* are illustrated in Figures 14, considering an example involving the inclusion of a rest time between requests 7 and 3. The first two illustrations (Figures 14a and 14b) show how the values of $lengthCurrR1_r$, $lengthCurrR2_r$, $lengthNextR1_r$ and $lengthNextR2_r$ would be as a function of the activations of $E1_{rs}$ and $E2_{rs}$, while the last one (Figure 14c) indicates how the values of $lengthCurrR1_r$, $lengthAmongR12_r$ and $lengthNextR2_r$ would be by $E12_{rs} = 1$. Based on Figure 14, constraints (54)-(72) determine the value of each length type. It is worth mentioning that placing variable $E12_{rs}$ in constraints (56)-(59) and (63)-(66) enforce $lengthCurrR2_r = 0$ and $lengthNextR1_r = 0$ in the circumstance of having rest before and after a ferry leg, consistent with which is shown in Figure 14c.

$$\begin{aligned} lengthCurrR1_r \geq\ & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{R}: \\ s \neq r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} \\ & - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E1_{rs} - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{r0});\ \forall\ r \in \mathcal{L}; \end{aligned} \tag{54}$$

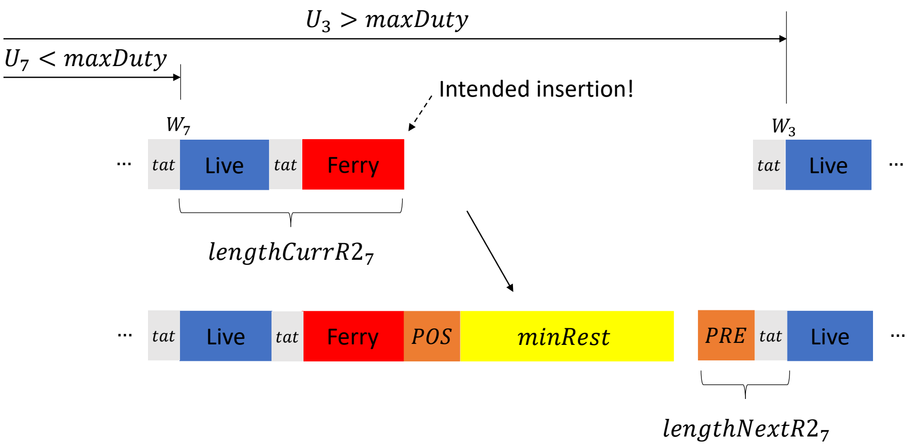
$$lengthCurrR1_r \leq M^4_r.(\sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E1_{rs} + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{r0});\ \forall\ r \in \mathcal{L}; \tag{55}$$

$$\begin{aligned} lengthCurrR2_r \geq\ & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ & - M^4_r.[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E12_{rs} - E2_{rs})];\ \forall\ r \in \mathcal{L}; \end{aligned} \tag{56}$$

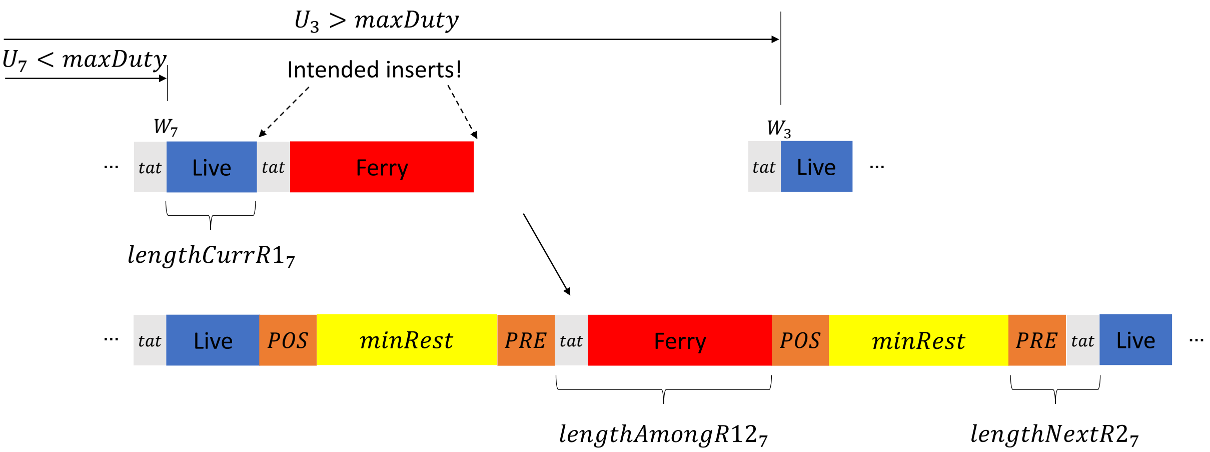
Figure 14 – Illustration of the three ways of including minimum rest.



(a)



(b)



(c)

Source: Own authorship.

$$lengthCurrR2_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E2_{rs} - E12_{rs}); \;\; \forall \; r \in \mathcal{L}; \tag{57}$$

$$\begin{aligned} lengthNextR1_r \geq\; & PRE + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ & + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} tat^s_{i^s}.y^v_{rs} \\ & - M^4_r.[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E12_{rs} - E1_{rs})]; \;\; \forall \; r \in \mathcal{L}; \end{aligned} \tag{58}$$

$$lengthNextR1_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E1_{rs} - E12_{rs}); \;\; \forall \; r \in \mathcal{L}; \tag{59}$$

$$\begin{aligned} lengthNextR2_r \geq\; & PRE + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} tat^s_{i^s}.y^v_{rs} \\ & - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E2_{rs}); \;\; \forall \; r \in \mathcal{L}; \end{aligned} \tag{60}$$

$$lengthNextR2_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E2_{rs}; \;\; \forall \; r \in \mathcal{L}; \tag{61}$$

$$lengthCurrR1_r = 0; \;\; \forall \; r \in \mathcal{M}; \tag{62}$$

$$\begin{aligned} lengthCurrR2_r \geq\; & \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} TL_r.y^{v^r}_{rs} + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} \\ & - M^4_r.[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E12_{rs} - E2_{rs})]; \;\; \forall \; r \in \mathcal{M}; \end{aligned} \tag{63}$$

$$lengthCurrR2_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E2_{rs} - E12_{rs}); \;\; \forall \; r \in \mathcal{M}; \tag{64}$$

$$\begin{aligned} lengthNextR1_r \geq\; & PRE + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} tat^s_{i^s}.y^{v^r}_{rs} \\ & - M^4_r.[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E12_{rs} - E1_{rs})]; \;\; \forall \; r \in \mathcal{M}; \end{aligned} \tag{65}$$

$$lengthNextR1_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E1_{rs} - E12_{rs}); \;\; \forall \; r \in \mathcal{M}; \tag{66}$$

$$lengthNextR2_r \geq PRE + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} tat^s_{i^s}.y^{v^r}_{rs} - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E2_{rs}); \;\; \forall \; r \in \mathcal{M}; \tag{67}$$

$$lengthNextR2_r \leq M^4_r . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E2_{rs}; \;\; \forall \; r \in \mathcal{M}; \tag{68}$$

$$\begin{aligned} lengthAmongR12_r \geq\; & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ & - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} E12_{rs}); \;\; \forall \; r \in \mathcal{L}; \end{aligned} \tag{69}$$

$$lengthAmongR12_r \leq M_r^4 . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} E12_{rs}; \ \ \forall \ r \in \mathcal{L}; \tag{70}$$

$$lengthAmongR12_r \geq \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF_{j^r i^s}^{\hat{p}^r} . y_{rs}^{v^r} - M_r^4 . (1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} E12_{rs}); \ \ \forall \ r \in \mathcal{M}; \tag{71}$$

$$lengthAmongR12_r \leq M_r^4 . \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} E12_{rs}; \ \ \forall \ r \in \mathcal{M}. \tag{72}$$

3.2.2.2 Crew break constraints

In modeling terms, split duty originates a nonlinear behavior. For a complete linearization, we divide the ground time into four time ranges, as shown in Table 3. The first column refers to each time range f, the next two, DL and DU, are the limits of ground time, and the last column shows how much the ground time will be (the new GT). From Table 3, $f = 4$ portrays an artificial situation, defined only to meet the logical purposes of the optimization model. Note that the break time is the difference between GT and new GT (the crew time free of duty).

Table 3 – Modification of the ground time due to split duty.

f	DL (min)	DU (min)	new GT (min)
1	0	90	GT
2	91	360	$(GT - 90)/2 + 90$
3	361	$minRest$	60
4	$minRest + 1$	∞	-

Source: Own authorship.

Based on these time ranges, we define the following parameters and variables:

- DL_f: ground time's lower bound belonging to range f;
- DU_f: ground time's upper bound belonging to range f;
- $B_{rs}^f \in \{0, 1\}$: binary variable that takes 1 if and only if the ground time between requests r and s is classified in the time range f;
- $GT_{rs} \in \mathbb{R}$: continuous variable that quantifies the ground time value between requests r and s;
- $acumGT_r \geq 0$: continuous variable that accumulates the potential ground time until request r, to be used as a break (split duty proceeding);
- $Duty_{rs} \in \mathbb{R}$: continuous variable that quantifies the duty time between requests r and s, if there is no ground time accumulation among them;

- $DA_{rs} \in \{0, 1\}$: binary variable that assumes value 1 if and only if there is ground time accumulation between requests r and s, given the use of maintenance events.

Constraints (73)-(74) make $Duty_{rs} = W_s - W_r$ for $DA_{rs} = 0$, while constraints (75)-(76) ensure that $Duty_{rs} = 0$ for $DA_{rs} = 1$ when maintenance durations are converted on ground time to have split duty. On the other hand, in constraints (77)-(82), where there are flight events among requests r and s, $Duty_{rs}$ receives the residual work time from the current journey for $DA_{rs} = 1$, so that it is transferred to U_s at constraints (48). The idea is that the value of U_s remains constant along an aircraft route that contains sequential maintenance requests (without ferry or live legs). Constraints (83) assure that if there is ground time accumulation between requests r and s, in the following request h, there may be a ground time accumulation again ($DA_{sh} = 1$, if $h \in \mathcal{M}$ and no ferry), or a break time may be applied ($B^2_{sh} = 1$ or $B^3_{sh} = 1$). The ground time accumulated up to request s, by variable $acumGT_s$, is warranted by constraints (84)-(92). They are analogous to constraints (118)-(128), employed to quantify $acumRest_s$. For the ground time between requests r and s, we elaborate the group of constraints (93)-(100). The presence of $acumGT_r$ in the subgroup (97)-(99) is justify by aggregating the ground time accumulation until request r to variable GT_{rs}, for $r \in \mathcal{M}$. Conversely, absence of TL_r in the subtractions is because maintenance times must already be in GT_{rs} itself.

$$Duty_{rs} \geq (W_s - W_r) - (M^3_r + M^3_s).DA_{rs};\ \ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s; \tag{73}$$

$$Duty_{rs} \leq (W_s - W_r) + (M^3_r + M^3_s).DA_{rs};\ \ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s; \tag{74}$$

$$Duty_{rs} \geq -(M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{75}$$

$$Duty_{rs} \leq (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{76}$$

$$\begin{aligned} Duty_{rs} \geq\ & TL_r + TF^{\hat{p}^r}_{j^r i^s} \\ & - (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{77}$$

$$\begin{aligned} Duty_{rs} \leq\ & TL_r + TF^{\hat{p}^r}_{j^r i^s} \\ & + (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{78}$$

$$\begin{aligned} Duty_{rs} \geq\ & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \check{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ & - (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{79}$$

$$\begin{aligned} Duty_{rs} \leq\ & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \check{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ & + (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{80}$$

$$\begin{aligned} Duty_{rs} \geq\ & \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \check{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} \\ & - (M^3_r + M^3_s).(1 - DA_{rs});\ \ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \end{aligned} \tag{81}$$

$$Duty_{rs} \leq \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} + (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \tag{82}$$

$$DA_{rs} \leq \sum_{\substack{h \in \mathcal{M}: \\ h \neq s}} DA_{sh} + \sum_{f=2}^{3} \sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}: \\ h \neq s}} B^f_{sh};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s; \tag{83}$$

$$acumGT_r \leq (M^3_r + M^3_s). \sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}: \\ h \neq r}} DA_{hr};\ \forall\ r \in \mathcal{M}; \tag{84}$$

$$acumGT_s \geq acumGT_r + (W_s - W_r) - (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{85}$$

$$acumGT_s \leq acumGT_r + (W_s - W_r) + (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{86}$$

$$acumGT_s \geq (W_s - W_r) - (TL_r + TF^{\hat{p}^r}_{j^r i^s}) - (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \tag{87}$$

$$acumGT_s \leq (W_s - W_r) - (TL_r + TF^{\hat{p}^r}_{j^r i^s}) + (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \tag{88}$$

$$acumGT_s \geq (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} - (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \tag{89}$$

$$acumGT_s \leq (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} + (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \tag{90}$$

$$acumGT_s \geq (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} - (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \tag{91}$$

$$acumGT_s \leq (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} + (M^3_r + M^3_s).(1 - DA_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \tag{92}$$

$$GT_{rs} = W_s - W_r - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}).y^v_{rs} - (POS + minRest + PRE).(E1_{rs} + E2_{rs});\ \forall\ r, s \in \mathcal{L} \mid r \neq s \wedge j^r \neq i^s; \tag{93}$$

$$GT_{rs} = W_s - W_r - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}).y^v_{rs} - (POS + minRest + PRE).E_{rs};\ \forall\ r, s \in \mathcal{L} \mid r \neq s \wedge j^r = i^s; \tag{94}$$

$$\begin{aligned} GT_{rs} = W_s - W_r - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} \\ - (POS + minRest + PRE).E1_{rs};\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{95}$$

$$GT_{rs} = W_s - W_r - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs};\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \tag{96}$$

$$\begin{aligned} GT_{rs} = W_s - W_r + acumGT_r - (TF^{\hat{p}^r}_{j^r i^s} + tat^s_{i^s}) \\ - RestM_s \\ - (POS + minRest + PRE).E2_{rs};\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r \neq i^s; \end{aligned} \tag{97}$$

$$\begin{aligned} GT_{rs} = W_s - W_r + acumGT_r - tat^s_{i^s} \\ - RestM_s;\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r = i^s; \end{aligned} \tag{98}$$

$$\begin{aligned} GT_{rs} = W_s - W_r + acumGT_r - TF^{\hat{p}^r}_{j^r i^s} \\ - RestM_s;\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{99}$$

$$GT_{rs} = W_s - W_r;\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s. \tag{100}$$

Given the ground time values, constraints (101) are used to classify each GT_{rs} between the ranges $f = 1, \ldots, 4$, if there is a route in requests r and s ($\sum_{v \in \mathcal{V}} y^v_{rs} = 1$), without the presence of the rest insertion ($E_{rs} = 0$). Lastly, the constraints (102)-(109) affect the reset of variable U_s by reducing the duty, given by the new ground time value (which triggers the break), being $(GT_{rs} - 90)/2 + 90$ when GT_{rs} is in range 2, and 60 for range 3. Note that $Duty_{rs}$ in (48) implies U_s to receive a U_r that has been held since the last request preceding r, which started the ground time accumulation.

$$\begin{aligned} \sum_{f=1}^{4} DL_f.B^f_{rs} - M^6.[1 + E_{rs} - \sum_{v \in \mathcal{V}} y^v_{rs}] \leq GT_{rs} \leq \\ \sum_{f=1}^{4} DU_f.B^f_{rs} + M^6.[1 + E_{rs} - \sum_{v \in \mathcal{V}} y^v_{rs}];\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \end{aligned} \tag{101}$$

$$\begin{aligned} U_s \geq U_r + (GT_{rs} - 90)/2 + 90 + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}).y^v_{rs} \\ - (M^3_r + M^3_s).(1 - B^2_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{102}$$

$$\begin{aligned} U_s \geq U_r + (GT_{rs} - 90)/2 + 90 + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}).y^v_{rs} \\ - (M^3_r + M^3_s).(1 - B^2_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{103}$$

$$\begin{aligned} U_s \geq U_r + (GT_{rs} - 90)/2 + 90 + (TF^{\hat{p}^r}_{j^r i^s} + tat^s_{i^s}).y^{v^r}_{rs} \\ - (M^3_r + M^3_s).(1 - B^2_{rs});\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{104}$$

$$\begin{aligned} U_s \geq U_r + (GT_{rs} - 90)/2 + 90 + tat^s_{i^s}.y^{v^r}_{rs} \\ - (M^3_r + M^3_s).(1 - B^2_{rs});\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{105}$$

$$U_s \geq U_r + 60 + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}).y^v_{rs} \\ - (M^3_r + M^3_s).(1 - B^3_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \quad (106)$$

$$U_s \geq U_r + 60 + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}).y^v_{rs} \\ - (M^3_r + M^3_s).(1 - B^3_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r = i^s; \quad (107)$$

$$U_s \geq U_r + 60 + (TF^{\hat{p}^r}_{j^r i^s} + tat^s_{i^s}).y^{v^r}_{rs} \\ - (M^3_r + M^3_s).(1 - B^3_{rs});\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \quad (108)$$

$$U_s \geq U_r + 60 + tat^s_{i^s}.y^{v^r}_{rs} \\ - (M^3_r + M^3_s).(1 - B^3_{rs});\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \wedge j^r = i^s. \quad (109)$$

3.2.2.3 Maintenance utilization constraints

Although uncommon, there can be more than one maintenance request in sequence, without the presence of a ferry among them. This is because $\Delta_{\mathcal{M}}$ is usually large (24 hours), enabling two immediately preceding requests r and s with the values $st_r + \Delta_{\mathcal{M}} + TL_r$ and $st_s - \Delta_{\mathcal{M}}$ close together, hindering a pre-processing that makes the grouping of these maintenance events. Consequently, one may lose feasible and optimal solutions by overlooking such circumstances, since maintenance events would be underused as the rest/break considerations. To get around this inconvenience, we need to define these decision variables:

- $firstM_s \in [0, 1]$: 1, if a maintenance (or dummy) request precedes other maintenance request s without ferry leg among them; 0, otherwise. The variable $firstM_s$ does not have to be binary, because constraints (110)-(115), shown further in the text, preserve its domain in a feasible solution; and

- $acumRest_s \geq 0$: potential accumulated ground time up to request s, to be used as rest.

To quantify the benefits of maintenance durations, the set of constraints (110)-(132) is applied. It is based on the first flight event on the route of each used aircraft, taking into account the presence of any maintenance request that may be at the beginning of the aircraft schedules. This mechanism is expressed by the group (110)-(117). The idea is to propagate the value 1 between variables $firstM_r$ and $firstM_s$, correlated to the immediately preceding maintenance requests with airports in common. As an example, suppose that the sequence of requests served by aircraft v is represented by the set $\mathcal{O}_v = \{0, 3, 5, 2, 4, 0\}$. Let $subL_v$ and $subM_v$ be subsets of requests, where $subL_v \subset \mathcal{L} = \{2, 4\}$, and $subM_v \subset \mathcal{M} = \{3, 5\}$. If $k^v = j^3 = i^5$, constraints (110)-(115) enforce

$firstM_3 = firstM_5 = 1$. If $k^v = j^3$ and $j^3 \neq i^5$, $firstM_3 = 1$ and $firstM_5 = 0$. Hence, a variable $firstM_s = 1$ means that from dummy request 0 to $s \in \mathcal{M}$, there are previous maintenance requests with the same airport.

Whenever we take advantage of maintenance in sequence among any requests with immediate precedence, the binary variable E_{rs} and consequently, some of the related variables ($E1_{rs}$, $E2_{rs}$ and $E12_{rs}$), must be activated. This behavior means that the crew is not working between requests r and s (even if there is no rest inclusion or break consideration, but instead, a ground time accumulation). Constraints (116) and (117) serve this purpose. If there is a ferry leg, request $s \in \mathcal{M}$ and $E2_{rs} = 1$, (116) forces $E1_{sh} = 1$, where request h follows s. In general terms, $E1_{sh} = 1$ represents an insertion of rest, if $h \in \mathcal{L}$ or $j^s \neq i^h$ (there is a flight event), or yet, a ground time accumulation, if $h \in \mathcal{M}$ and $j^s = i^h$. The second condition is propagated by constraints (117).

$$firstM_s \geq y_{0s}^{v^s};\ \forall\ s \in \mathcal{M} \mid k^{v^s} = i^s; \tag{110}$$
$$firstM_s \leq 1 - y_{0s}^{v^s};\ \forall\ s \in \mathcal{M} \mid k^{v^s} \neq i^s; \tag{111}$$
$$firstM_s \geq firstM_r - (1 - E_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{112}$$
$$firstM_s \leq firstM_r + (1 - E_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \tag{113}$$
$$firstM_s \leq 1 - y_{rs}^{v^r};\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \tag{114}$$
$$firstM_r \leq \sum_{\substack{h \in \mathcal{M}:\\ h \neq r}} E_{hr} + y_{0r}^{v^r};\ \forall\ r \in \mathcal{M}; \tag{115}$$
$$E2_{rs} \leq \sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}:\\ h \neq s}} E1_{sh} + y_{s0}^{v^s};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \tag{116}$$
$$E_{rs} \leq \sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}:\\ h \neq s}} E1_{sh} + y_{s0}^{v^s};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s. \tag{117}$$

The block of constraints (118)-(128) determines the value of variable $acumRest_r$, so that the minimum rest is met when making use of sequential maintenance durations. To better explain block (118)-(128), it is useful to partition these constraints into (118), (119)-(122) and (123)-(128). Constraints (118) enforce $acumRest_s = 0$ when $\sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}:\\ h \neq r}} E_{hr} = 0$ for each $r \in \mathcal{L} \cup \mathcal{M}$. Constraints (119)-(122) perform the ground time accumulation for rest purposes, whereas (119)-(120) guarantee that $acumRest_s = acumRest_r + (W_s - W_r)$, if $E_{rs} = 1$ and $\sum_{\substack{h \in \mathcal{M}:\\ h \neq r\\ j^h \neq i^r}} E1_{hr} = 0$. Constraints (121)-(122) assure $acumRest_s = (W_s - W_r)$, if $E_{rs} = 1$ and $\sum_{\substack{h \in \mathcal{M}:\\ h \neq r\\ j^h \neq i^r}} E1_{hr} = 1$. This sum activation portrays that there is a ferry between requests $h, r \in \mathcal{M}$, and because it is a flight event, $E1_{hr} = 1$ actually includes a rest before the ferry (not a ground time accumulation). The last constraints, (123)-(128), aim to quantify the first ground time in the accumulation. For example, suppose that in the sequence of maintenance request $r \in \{3, 2, 5\}$, it is possible to use their duration as ground time, so we obtain $acumRest_3 > 0$ (the first ground time used as rest to be included when the accumulation arrives on request 5). The identification of the first ground time in the accumulation for a rest is based on the premise that request $s \in \mathcal{M}$

(i.e., we can take advantage of the subsequent maintenance request), and the existence of a flight event, when $E_{rs} = 1$ for $r \in \mathcal{L} \wedge j^r = i^s$ or $E2_{rs} = 1$ for $j^r \neq i^s$ (it means that the time interval between the beginning of request s and the end of flight event that starts at request r can be used in favor of rest time).

Consolidated the measurement of $acumRest_r$, constraints (129)-(132) are in charge of quantifying $RestM_s$ (i.e., the resulting value obtained from the ground time accumulation by previous maintenance requests) given a request $r \in \mathcal{M}$, and the presence of a flight event at request s, when $E_{rs} = 1$ for $s \in \mathcal{L} \wedge j^r = i^s$ or $E2_{rs} = 1$ for $j^r \neq i^s$. Observe that $firstM_r = 1$ implies $RestM_r = 0$, avoiding unnecessary rest insertion for a route that has maintenance requests at its beginning.

$$acumRest_r \leq M_r^3. \sum_{\substack{h \in \mathcal{L} \cup \mathcal{M}: \\ h \neq r}} E_{hr}; \;\; \forall \; r \in \mathcal{L} \cup \mathcal{M}; \tag{118}$$

$$\begin{aligned} acumRest_s \geq \; & acumRest_r + (W_s - W_r) \\ & - (M_r^3 + M_s^3).(1 + \sum_{\substack{h \in \mathcal{M}: \\ h \neq r \\ j^h \neq i^r}} E1_{hr} - E_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{119}$$

$$\begin{aligned} acumRest_s \leq \; & acumRest_r + (W_s - W_r) \\ & + (M_r^3 + M_s^3).(1 + \sum_{\substack{h \in \mathcal{M}: \\ h \neq r \\ j^h \neq i^r}} E1_{hr} - E_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{120}$$

$$\begin{aligned} acumRest_s \geq \; & (W_s - W_r) \\ & - (M_r^3 + M_s^3).(2 - \sum_{\substack{h \in \mathcal{M}: \\ h \neq r \\ j^h \neq i^r}} E1_{hr} - E_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{121}$$

$$\begin{aligned} acumRest_s \leq \; & (W_s - W_r) \\ & + (M_r^3 + M_s^3).(2 - \sum_{\substack{h \in \mathcal{M}: \\ h \neq r \\ j^h \neq i^r}} E1_{hr} - E_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \end{aligned} \tag{122}$$

$$\begin{aligned} acumRest_s \geq \; & (W_s - W_r) - (TL_r + TF_{j^r i^s}^{\hat{p}^r}) \\ & - (M_r^3 + M_s^3).(1 - E2_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{123}$$

$$\begin{aligned} acumRest_s \leq \; & (W_s - W_r) - (TL_r + TF_{j^r i^s}^{\hat{p}^r}) \\ & + (M_r^3 + M_s^3).(1 - E2_{rs}); \;\; \forall \; r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{124}$$

$$\begin{aligned} acumRest_s \geq \; & (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF_{i^r j^r}^{\check{p}^v} + tat_{j^r}^s + TF_{j^r i^s}^{\check{p}^v}).y_{rs}^v \\ & - (M_r^3 + M_s^3).(1 - E2_{rs}); \;\; \forall \; r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{125}$$

$$\begin{aligned} acumRest_s \leq \; & (W_s - W_r) - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} (TF_{i^r j^r}^{\check{p}^v} + tat_{j^r}^s + TF_{j^r i^s}^{\check{p}^v}).y_{rs}^v \\ & + (M_r^3 + M_s^3).(1 - E2_{rs}); \;\; \forall \; r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{126}$$

$$
\begin{aligned}
acumRest_s \geq (W_s - W_r) - &\sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF_{i^r j^r}^{\check{p}^v}.y_{rs}^v \\
&- (M_r^3 + M_s^3).(1 - E_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s;
\end{aligned}
\tag{127}
$$

$$
\begin{aligned}
acumRest_s \leq (W_s - W_r) - &\sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} TF_{i^r j^r}^{\check{p}^v}.y_{rs}^v \\
&+ (M_r^3 + M_s^3).(1 - E_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s;
\end{aligned}
\tag{128}
$$

$$
\begin{aligned}
RestM_s \geq (POS + minRest + PRE) &- (acumRest_r + TL_r) \\
- M_r^5.(1 + firstM_r - E1_{rs});\ &\forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s;
\end{aligned}
\tag{129}
$$

$$
\begin{aligned}
RestM_s \geq (POS + minRest + PRE) &- (acumRest_r + TL_r) \\
- M_r^5.(1 + firstM_r - E1_{rs});\ &\forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r \neq i^s;
\end{aligned}
\tag{130}
$$

$$
\begin{aligned}
RestM_s \geq (POS + minRest + PRE) &- (acumRest_r + TL_r) \\
- M_r^5.(1 + firstM_r - E_{rs});\ &\forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r = i^s;
\end{aligned}
\tag{131}
$$

$$
RestM_r \leq M_r^3.(1 - firstM_r);\ \forall\ r \in \mathcal{M}. \tag{132}
$$

3.2.2.4 Crew flying time constraints

Analogous to the rest rules, we define continuous variables to calculate the accumulated flight time in a duty, and ensure the residual time is consistent with the duration of live and/or ferry leg at the end of each current duty. Thus, consider the additional definitions:

- $Q_s \geq 0$: accumulated flight time up to request s in a duty;
- $lengthCurrFD1_r \geq 0$: duration of a live leg at the end of a current duty by request r;
- $lengthCurrFD2_r \geq 0$: duration of a ferry leg at the end of a current duty by request r;
- $lengthCurrFD12_r \geq 0$: ferry time between the end and start of consecutive duties by request r.

We set the family of constraints (133)-(161) to comply with the maximum flying time rule. In it, Q_s measures the total flight time (live and ferry legs) up to request s. With the presence of E_{rs}, variable Q_s could be reset for each duty by constraints (133)-(144), assuming new values in (145)-(147) for the next duty, being the accumulated flight time checked in (148)-(150) for the current duty, which corresponds to follow the same principle of the rest constraints (43)-(53). Inspired on (DESROCHERS; LAPORTE, 1991), we develop stronger MTZ-based constraints than the traditional forms (MILLER; TUCKER; ZEMLIN, 1960; ÖNCAN; ALTINEL; LAPORTE, 2009). Note that apart from variables y_{rs}^v and E_{rs}, there are also the variables y_{sr}^v and E_{sr}. By focusing on y_{rs}^v, we have three distinct cases: (i) $y^v_{rs} = 1$ when $y^v_{sr} = 0$; (ii) $y^v_{sr} = 1$ for $y^v_{rs} = 0$; or yet (iii) $y^v_{rs} = 0$ and $y^v_{sr} = 0$. These are valid for E_{rs} too. For example, in constraints (135), suppose that request $r = 4$ and $s = 3$, performed by aircraft v, where $y^v_{43} = 1$. Therefore, the first case implies $Q_3 \geq Q_4 + TF^{\check{p}^v}_{i^4 j^4} + TF^{\check{p}^v}_{j^4 i^3}$, and the second case corresponds to do $Q_4 \geq Q_3 - TF^{\check{p}^v}_{i^4 j^4} - TF^{\check{p}^v}_{j^4 i^3}$, which means $Q_3 = Q_4 + TF^{\check{p}^v}_{i^4 j^4} + TF^{\check{p}^v}_{j^4 i^3}$, in addition to improving the linear relaxation.

Through the variables $lengthCurrFD1_r$ and $lengthCurrFD2_r$, which quantify the live and ferry durations, respectively, and by variable $lengthAmongFD12_r$, which gives the ferry time where there is rest time before/after a positioning flight, the flight times are determined on the end of a current duty and the start of the next one by constraints (151)-(161).

$$Q_s \geq TF^{\check{p}^v}_{k^v i^s}.y^v_{0s} - M^1_s.(1 + E0_{vs} - y^v_{0s});\ \forall\ v \in \mathcal{V}, s \in \mathcal{L} \mid k^v \neq i^s; \tag{133}$$

$$Q_s \leq M^1_s.(1 - y^v_{0s});\ \forall\ v \in \mathcal{V}, s \in \mathcal{L} \mid k^v = i^s; \tag{134}$$

$$\begin{aligned} Q_s \geq\ & Q_r + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} (TF^{\check{p}^v}_{i^s j^s} + TF^{\check{p}^v}_{j^s i^r}).y^v_{sr} \\ & - M^1_r.\left[1 + (E_{rs} + E_{sr}) - (\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{rs} + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} y^v_{sr})\right];\ \forall\ r, s \in \mathcal{L} \mid r \neq s \land j^r \neq i^s; \end{aligned} \tag{135}$$

$$\begin{aligned} Q_s \geq\ & Q_r + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} (TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs} - TF^{\hat{p}^s}_{j^s i^r}.y^{v^s}_{sr} \\ & - M^1_r.\left[1 + (E_{rs} + E_{sr}) - (\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{rs} + y^{v^s}_{sr})\right];\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{136}$$

$$\begin{aligned} Q_s \geq\ & Q_r + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} TF^{\check{p}^v}_{i^s j^s}.y^v_{sr} \\ & - M^1_r.\left[1 + (E_{rs} + E_{sr}) - (\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{rs} + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} y^v_{sr})\right];\ \forall\ r, s \in \mathcal{L} \mid r \neq s \land j^r = i^s; \end{aligned} \tag{137}$$

$$\begin{aligned} Q_s \geq\ & Q_r + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} \\ & - M^1_r.(1 + E_{rs} - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r = i^s; \end{aligned} \tag{138}$$

$$Q_s \geq TF^{\hat{p}^s}_{k^{v^s} i^s}.y^{v^s}_{0s};\ \forall\ s \in \mathcal{M} \mid k^{v^s} \neq i^s; \tag{139}$$

$$Q_s \leq M^2_s.(1 - y^{v^s}_{0s});\ \forall\ s \in \mathcal{M} \mid k^{v^s} = i^s; \tag{140}$$

$$\begin{aligned} Q_s \geq\ & Q_r + TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} - TF^{\hat{p}^s}_{j^s i^r}.y^{v^s}_{sr} \\ & - M^2_r.[1 + (E_{rs} + E_{sr}) - (y^{v^r}_{rs} + y^{v^s}_{sr})];\ \forall\ r, s \in \mathcal{M} \mid r \neq s \land j^r \neq i^s; \end{aligned} \tag{141}$$

$$Q_s \geq Q_r + TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} (TF^{\check{p}^v}_{i^s j^s} + TF^{\check{p}^v}_{j^s i^r}).y^v_{sr}$$
$$- M^2_r.[1 + (E_{rs} + E_{sr}) - (y^{v^r}_{rs} + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} y^v_{sr})];\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r \neq i^s; \quad (142)$$

$$Q_s \geq Q_r - M^2_r.(1 + E_{rs} - y^{v^r}_{rs});\ \forall\ r, s \in \mathcal{M} \mid r \neq s \wedge j^r = i^s; \quad (143)$$

$$Q_s \geq Q_r - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} TF^{\check{p}^v}_{i^s j^s}.y^v_{sr}$$
$$- M^2_r.[1 + (E_{rs} + E_{sr}) - (y^{v^r}_{rs} + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^s}} y^v_{sr})];\ \forall\ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r = i^s; \quad (144)$$

$$Q_s \leq M^1_s.(1 - \sum_{v\in\mathcal{V}} E0_{vs});\ \forall\ s \in \mathcal{L}; \quad (145)$$

$$Q_s \geq \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{r\in\mathcal{L}:\\ s\neq r\\ j^r\neq i^s}} TF^{\check{p}^v}_{j^r i^s}.y^v_{rs} + \sum_{\substack{r\in\mathcal{M}:\\ s\neq r\\ j^r\neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs}$$
$$- M^3_s.[1 + \sum_{\substack{r\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r}} (E12_{rs} - E1_{rs})];\ \forall\ s \in \mathcal{L} \cup \mathcal{M}; \quad (146)$$

$$Q_s \leq M^3_s.(1 - \sum_{\substack{r\in\mathcal{L}\cup\mathcal{M}:\\ r\neq s}} E2_{rs});\ \forall\ s \in \mathcal{L} \cup \mathcal{M}; \quad (147)$$

$$Q_r + lengthCurrFD1_r + lengthCurrFD2_r \leq maxflying + overR_r;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \quad (148)$$

$$lengthAmongFD12_r \leq maxflying + overF_r;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \quad (149)$$

$$over0_v \geq \sum_{s\in\mathcal{L}} TF^{\check{p}^v}_{k^v i^s}.E0_{vs} - maxflying;\ \forall\ v \in \mathcal{V}; \quad (150)$$

$$lengthCurrFD1_r \geq \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs}$$
$$- M^4_r.(1 - \sum_{\substack{s\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r}} E_{rs} - \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{r0});\ \forall\ r \in \mathcal{L}; \quad (151)$$

$$lengthCurrFD1_r \leq M^4_r.(\sum_{\substack{s\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r}} E_{rs} + \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} y^v_{r0});\ \forall\ r \in \mathcal{L}; \quad (152)$$

$$lengthCurrFD2_r \geq \sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r\\ j^r\neq i^s}} TF^{\check{p}^v}_{j^r i^s}.y^v_{rs}$$
$$- M^4_r.[1 + \sum_{\substack{s\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r}} (E12_{rs} - E2_{rs})];\ \forall\ r \in \mathcal{L}; \quad (153)$$

$$lengthCurrFD2_r \leq M^4_r. \sum_{\substack{s\in\mathcal{L}\cup\mathcal{M}:\\ s\neq r}} (E2_{rs} - E12_{rs});\ \forall\ r \in \mathcal{L}; \quad (154)$$

$$lengthCurrFD1_r = 0;\ \forall\ r \in \mathcal{M}; \quad (155)$$

$$lengthCurrFD2_r \geq \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} - M^4_r.[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E12_{rs} - E2_{rs})];\ \forall\ r \in \mathcal{M}; \tag{156}$$

$$lengthCurrFD2_r \leq M^4_r. \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} (E2_{rs} - E12_{rs});\ \forall\ r \in \mathcal{M}; \tag{157}$$

$$lengthAmongFD12_r \geq \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\check{p}^v}_{j^r i^s}.y^{v}_{rs} - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E12_{rs});\ \forall\ r \in \mathcal{L}; \tag{158}$$

$$lengthAmongFD12_r \leq M^4_r. \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E12_{rs};\ \forall\ r \in \mathcal{L}; \tag{159}$$

$$lengthAmongFD12_r \geq \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs} - M^4_r.(1 - \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E12_{rs});\ \forall\ r \in \mathcal{M}; \tag{160}$$

$$lengthAmongFD12_r \leq M^4_r. \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r}} E12_{rs};\ \forall\ r \in \mathcal{M}. \tag{161}$$

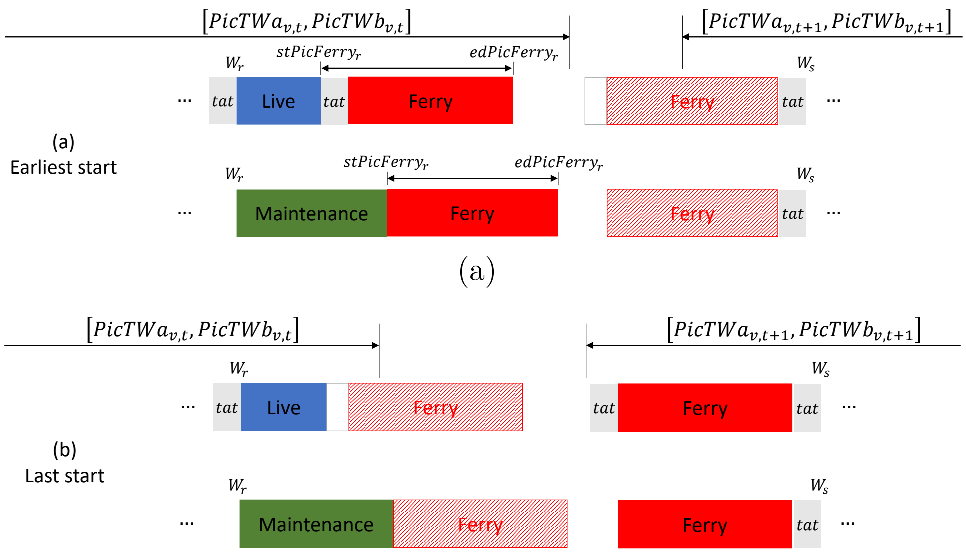
3.2.2.5 Pilot rostering constraints

To determine whether the pilots' time windows cover the entire flight schedules, that is, under the condition that there is always a pilot available to perform a flight assigned to a certain aircraft, we define:

- nTW_v: number of pilot time windows of aircraft v;
- $[PicTWa_{vt}, PicTWb_{vt}]$: time window t of a pilot assigned to aircraft v;
- $zl^{vt}_r \in \{0, 1\}$: binary variable that assumes 1 if and only if customer request r is served inside the time window t of a pilot allocated to aircraft v;
- $zf^{vt}_r \in \{0, 1\}$: binary variable that assumes 1 if and only if request r with ferry leg is within the time window t of a pilot assigned to aircraft v;
- $stPicFerry_r \geq 0$ and $edPicFerry_r \geq 0$: continuous variables related to the exact time to start/end of a ferry leg from request r on the time horizon;
- $stPicFerry0_v \geq 0$ and $edPicFerry0_v \geq 0$: continuous variables that correspond to the exact time to start/end of the first ferry leg executed by aircraft v.

Constraints (162)-(187) concern pilots' time windows. We separate this family into three subgroups: (162)-(167) stipulate in which time window the starting/ending time of a flight event is located, (168)-(183) determine exactly the starting/ending time of a ferry contained between requests r and s, as well as a ferry found between the dummy request and the first request served by an aircraft, and (184)-(187) enforce the fulfillment of pilot's time window, linking the index t with the indices v and r. Differently from a live leg, where we know its start time and trivially its end time by variable W_r, we declare the continuous variables $stPicFerry_r$, $edPicFerry_r$, $stPicFerry0_v$ and $edPicFerry0_v$ to have knowledge of these times, since the location of a ferry in the planning horizon can be from the end of a live or maintenance event until the beginning of the next live or maintenance event, as illustrated in Figure 15. When a rest is included, the company policy establishes that if the rest is placed before a ferry, that same ferry must earliest start, and if rest is after, that ferry must last start. This conduct makes all ground time existing among flight events be converted as rest, in order to improve the crew's well-being. In this way, constraints (169) and (173) are activated for the first situation, and (175) and (179) for the second one.

Figure 15 – The earliest and last start of a ferry into pilot's time windows.



(a)

(b)

Source: Own authorship.

$$\sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} PicTWa_{vt}.zl_r^{vt} - (M_r^3 + M_r^4).(1 - \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} zl_r^{vt}) \leq W_r - tat_{ir}^r;\ \forall\ r \in \mathcal{L}; \quad (162)$$

$$W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{R}: \\ s \neq r}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} \leq \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} PicTWb_{vt}.zl^{vt}_r + (M^3_r + M^4_r).(1 - \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} zl^{vt}_r);\ \forall\ r \in \mathcal{L}; \tag{163}$$

$$\sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} PicTWa_{vt}.zf^{vt}_r - (M^3_r + M^4_r).(1 - \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} zf^{vt}_r) \leq stPicFerry_r;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{164}$$

$$edPicFerry_r \leq \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} PicTWb_{vt}.zf^{vt}_r + (M^3_r + M^4_r).(1 - \sum_{v \in \mathcal{V}} \sum_{t=1}^{nTW_v} zf^{vt}_r);\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{165}$$

$$\sum_{t=1}^{nTW_v} PicTWa_{vt}.zf^{vt}_0 - M^6.(1 - \sum_{t=1}^{nTW_v} zf^{vt}_0) \leq stPicFerry0_v;\ \forall\ v \in \mathcal{V}; \tag{166}$$

$$edPicFerry0_v \leq \sum_{t=1}^{nTW_v} PicTWb_{vt}.zf^{vt}_0 + M^6.(1 - \sum_{t=1}^{nTW_v} zf^{vt}_0);\ \forall\ v \in \mathcal{V}; \tag{167}$$

$$stPicFerry_r \geq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (POS + minRest + PRE).E1_{rs};\ \forall\ r \in \mathcal{L}; \tag{168}$$

$$stPicFerry_r \leq W_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\check{p}^v}_{i^r j^r}.y^v_{rs} + (M^3_r + M^4_r).[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (E12_{rs} - E2_{rs})];\ \forall\ r \in \mathcal{L}; \tag{169}$$

$$edPicFerry_r = stPicFerry_r + \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}).y^v_{rs};\ \forall\ r \in \mathcal{L}; \tag{170}$$

$$edPicFerry_r \leq W_s - tat^s_{i^s} - (POS + minRest + PRE).E2_{rs} + (M^3_r + M^4_s).(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\ \forall\ r, s \in \mathcal{L} \mid r \neq s \wedge j^r \neq i^s; \tag{171}$$

$$edPicFerry_r \leq W_s - tat^s_{i^s} + (M^3_r + M^4_s).(1 - \sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} y^v_{rs});\ \forall\ r \in \mathcal{L}, s \in \mathcal{M} \mid j^r \neq i^s; \tag{172}$$

$$edPicFerry_r \geq W_s - tat^s_{i^s} - (M^3_s + M^4_s).[1 + (E12_{rs} - E1_{rs})];\ \forall\ r \in \mathcal{L}, s \in \mathcal{L} \cup \mathcal{M} \mid j^r \neq i^s; \tag{173}$$

$$stPicFerry_r \geq W_r + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TL_r.y^{v^r}_{rs} + RestM_r; \ \forall \ r \in \mathcal{M}; \tag{174}$$

$$\begin{aligned} stPicFerry_r \leq W_r + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TL_r.y^{v^r}_{rs} + RestM_r \\ + (M^3_r + M^4_r).[1 + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} (E12_{rs} - E2_{rs})]; \ \forall \ r \in \mathcal{M}; \end{aligned} \tag{175}$$

$$edPicFerry_r = stPicFerry_r + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} TF^{\hat{p}^r}_{j^r i^s}.y^{v^r}_{rs}; \ \forall \ r \in \mathcal{M}; \tag{176}$$

$$\begin{aligned} edPicFerry_r \leq W_s - tat^s_{i^s} - (POS + minRest + PRE).E2_{rs} \\ + (M^3_r + M^4_s).(1 - y^{v^r}_{rs}); \ \forall \ r \in \mathcal{M}, s \in \mathcal{L} \mid j^r \neq i^s; \end{aligned} \tag{177}$$

$$\begin{aligned} edPicFerry_r \leq W_s - tat^s_{i^s} \\ + (M^3_r + M^4_s).(1 - y^{v^r}_{rs}); \ \forall \ r, s \in \mathcal{M} \mid r \neq s \wedge j^r \neq i^s; \end{aligned} \tag{178}$$

$$\begin{aligned} edPicFerry_r \geq W_s - tat^s_{i^s} \\ - (M^3_s + M^4_s).[1 + (E12_{rs} - E1_{rs})]; \\ \forall \ r \in \mathcal{M}, s \in \mathcal{L} \cup \mathcal{M} \mid j^r \neq i^s; \end{aligned} \tag{179}$$

$$stPicFerry0_v \geq av_v + PRE; \ \forall \ v \in \mathcal{V}; \tag{180}$$

$$edPicFerry0_v = stPicFerry0_v + \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ k^v \neq i^s}} TF^{\check{p}^v}_{k^v i^s}.y^v_{0s}; \ \forall \ v \in \mathcal{V}; \tag{181}$$

$$\begin{aligned} edPicFerry0_v \leq W_s - tat^s_{i^s} - (POS + minRest + PRE).E0_{vs} \\ + (M^3_s + M^4_s).(1 - y^v_{0s}); \ \forall \ v \in \mathcal{V}, s \in \mathcal{L} \mid k^v \neq i^s; \end{aligned} \tag{182}$$

$$edPicFerry0_{v^s} \leq W_s + (M^3_s + M^4_s).(1 - y^{v^s}_{0s}); \ \forall \ s \in \mathcal{M} \mid k^{v^s} \neq i^s; \tag{183}$$

$$\sum_{t=1}^{nTW_v} zl^{vt}_r = \sum_{\substack{s \in \mathcal{R}: \\ s \neq r}} y^v_{rs}; \ \forall \ v \in \mathcal{V}, r \in \mathcal{L}; \tag{184}$$

$$\sum_{t=1}^{nTW_v} zf^{vt}_r = \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} y^v_{rs}; \ \forall \ v \in \mathcal{V}, r \in \mathcal{L}; \tag{185}$$

$$\sum_{t=1}^{nTW_{v^r}} zf^{v^r,t}_r = \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ s \neq r \\ j^r \neq i^s}} y^{v^r}_{rs}; \ \forall \ r \in \mathcal{M}; \tag{186}$$

$$\sum_{t=1}^{nTW_v} zf^{vt}_0 = \sum_{\substack{s \in \mathcal{L} \cup \mathcal{M}: \\ k^v \neq i^s}} y^v_{0s}; \ \forall \ v \in \mathcal{V}. \tag{187}$$

3.2.2.6 Objective function including outsourcing and overtime costs and domain of decision variables

As the demand of requests is mandatory, we created slack variables to make the operation more flexible, allowing subterfuges that are penalized in the objective function, such as outsourcing and overtime costs. Below, there are the related parameters and variables:

- $Cout_r$: outsourcing cost to carry out live request r;
- $overtPerc$: percentage used on the travel cost to pay crew members' overtime;
- $out_r \in \{0,1\}$: binary variable that indicates with 1 whether live request r should be serviced by another company (outsourcing event);
- $overR_r \geq 0$: continuous variable that quantifies the overtime performed by the crew in live request r;
- $overF_r \geq 0$: continuous variable that quantifies the overtime performed by the crew on a ferry leg (if any) in request r;
- $over0_v \geq 0$: continuous variable that quantifies the overtime obtained by the first ferry leg (if any) of aircraft v.

In the proposed model, we also use the family of constraints (4)-(9) (Subsection 3.2.1), just adding variable out_r in (4), i.e.:

$$\sum_{\substack{v \in \mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\substack{s \in \mathcal{R}: \\ s \neq r}} y_{rs}^v + out_r = 1; \ \forall \ r \in \mathcal{L}; \tag{188}$$

which make it possible to outsource a customer request r, if $out_r = 1$.

The objective function (189) is composed of four terms. The first and second are related to repositioning and upgrade costs, while the other two represent the outsourcing and overtime costs, respectively.

$$\begin{aligned} \min \ & \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} \sum_{\substack{s \in \mathcal{R}: \\ r \neq s}} Cf_{rs}^v . y_{rs}^v + \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{L}} \sum_{\substack{s \in \mathcal{R}: \\ r \neq s}} Cup_r^v . y_{rs}^v + \sum_{r \in \mathcal{L}} Cout_r . out_r \\ & + \left[\sum_{r \in \mathcal{L} \cup \mathcal{M}} c_{\hat{p}^r} . overPerc . (overR_r + overF_r) + \sum_{v \in \mathcal{V}} c_{\check{p}^v} . overPerc . over0_v \right]. \end{aligned} \tag{189}$$

Our formulation is completed by (190)-(208), which defines the domain of decision variables.

$$y_{rs}^v \in \{0,1\}; \ \forall \ v \in \mathcal{V}, r, s \in \mathcal{R}; \tag{190}$$
$$out_r \geq 0; \ \forall \ r \in \mathcal{L}; \tag{191}$$
$$W_r \geq 0, U_r \geq 0, Q_r \geq 0; \ \forall \ r \in \mathcal{L} \cup \mathcal{M}; \tag{192}$$
$$E0_{vr} \in \{0,1\}; \ \forall \ v \in \mathcal{V}, r \in \mathcal{L}; \tag{193}$$
$$E1_{rs} \in \{0,1\}, E2_{rs} \in \{0,1\}, E12_{rs} \in \{0,1\}, E_{rs} \in \{0,1\}; \ \forall \ r, s \in \mathcal{L} \cup \mathcal{M}; \tag{194}$$
$$overR_r \geq 0, overF_r \geq 0; \ \forall \ r \in \mathcal{L} \cup \mathcal{M}; \tag{195}$$
$$over0_v \geq 0; \ \forall \ v \in \mathcal{V}; \tag{196}$$

$$lengthCurrR1_r \geq 0, lengthCurrR2_r \geq 0, lengthNextR1_r \geq 0, \\ lengthNextR2_r \geq 0, lengthAmongR12_r \geq 0;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{197}$$

$$lengthCurrFD1_r \geq 0, lengthCurrFD2_r \geq 0, \\ lengthAmongFD12_r \geq 0;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{198}$$

$$firstM_r \in [0,1];\ \forall\ r \in \mathcal{M}; \tag{199}$$

$$acumRest_r \geq 0, RestM_r \geq 0;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{200}$$

$$B^f_{rs} \in \{0,1\};\ \forall\ r,s \in \mathcal{L} \cup \mathcal{M}, f = 1,\ldots,4; \tag{201}$$

$$GT_{rs} \in \mathbb{R};\ \forall\ r,s \in \mathcal{L} \cup \mathcal{M}; \tag{202}$$

$$acumGT_r \geq 0;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; \tag{203}$$

$$Duty_{rs} \in \mathbb{R}, DA_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s \in \mathcal{M}; \tag{204}$$

$$zl^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{L}; v \in \mathcal{V}; t = 1,\ldots,nTW_v; \tag{205}$$

$$zf^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{R}; v \in \mathcal{V}; t = 1,\ldots,nTW_v; \tag{206}$$

$$stPicFerry0_v \geq 0, edPicFerry0_v \geq 0;\ \forall\ v \in \mathcal{V}; \tag{207}$$

$$stPicFerry_r \geq 0, edPicFerry_r \geq 0;\ \forall\ r \in \mathcal{L} \cup \mathcal{M}. \tag{208}$$

3.2.2.7 Variable pre-fixing

Finally, this subsection concludes the formulation by showing how the variables are pre-fixed. Constraints (209)-(212) are added to the proposed model to reduce the number of variables, after all, pre-solve does not treat variable pre-fixing as constraints inserted into the coefficient matrix, but rather as variables to be removed from the model, permitting them to speed up the approach's execution times through the use of general-purpose optimization software.

$$y^v_{rs} = 0;\ \forall\ v \in \mathcal{V}, r \in \mathcal{L}, s \in \mathcal{M} \mid st_r + TF^P_{i^r j^r} \geq st_s + \Delta_{\mathcal{M}}; \tag{209}$$

$$y^v_{rs} = 0;\ \forall\ v \in \mathcal{V}, r \in \mathcal{M}, s \in \mathcal{L} \mid st_r - \Delta_{\mathcal{M}} + TL_r \geq st_s + \Delta_{\mathcal{L}}; \tag{210}$$

$$y^v_{rs} = 0;\ \forall\ v \in \mathcal{V}; r,s \in \mathcal{L} \mid s \neq r \wedge st_r + TF^P_{i^r j^r} \geq st_s + \Delta_{\mathcal{L}}; \tag{211}$$

$$y^v_{rs} = 0;\ \forall\ v \in \mathcal{V}; r,s \in \mathcal{M} \mid s \neq r \wedge st_r - \Delta_{\mathcal{M}} + TL_r \geq st_s + \Delta_{\mathcal{M}}. \tag{212}$$

As each decision variable W_r is bounded by the request time window, if we consider that request r precedes request s, given the values of st_r and st_s along with their anticipation/postponement tolerances, it is impossible that the time window opening plus the duration of request r will be greater than the time window closing for request s, that is, a situation where there is a reversal of precedence among these requests. Hence, the idea is to pre-set at zero each variable y^v_{rs} that portrays such a circumstance, contemplating the four possible combinations of requisition types (customer with maintenance, maintenance with customer, customer with customer and maintenance with maintenance).

3.3 Heuristic approach

Exactly solving the studied DAFP, when in view of medium and larger instances, may not be viable in practice given the computational effort and time demanded by our optimization model (22)-(208). Hence, in this section, we propose to apply relax-and-fix and fix-and-optimize heuristic methods in this formulation to try reaching superior quality solutions. These two approaches are, by definition, matheuristics – hybridizations that combine heuristic or metaheuristic methods with exact mathematical programming techniques. Specifically, they are classified as type I (MANIEZZO; STÜTZLE; VOSS, 2021), insofar as we have these heuristics acting at a higher level and controlling the call of exact method for solving the mathematical model (not the other way around, which are type II). Next, we present our relax-and-fix (constructive) and fix-and-optimize (improvement) heuristics in Subsections 3.3.1 and 3.3.2, respectively.

3.3.1 Relax-and-fix heuristic

The relax-and-fix (R&F) heuristic was initially presented by Wolsey (1998) and consists of a decomposition method destined to a mixed-integer programming model in order to dismember it into smaller and disjoint submodels, which can be resolved more quickly, but without the guarantee of optimality, and depending on the case, also of feasibility. This decomposition is characterized by partitioning the set of integer variables from original problem Γ into I subsets, given by Γ^i, $\forall\ i = 1, \dots, I$. The number of subsets I determines the total of iterations. For each iteration i in the derived model, only the variables in subset Γ^i are considered as integer variables. Subsets $\Gamma^1, \dots, \Gamma^{i-1}$ are those formed by variables fixed on values obtained from solutions of integer variables solved in previous iterations, that is, they assume values that will compose the final feasible solution of problem Γ, if found at the end of procedure. The subsets $\Gamma^{i+1}, \dots, \Gamma^I$ are composed of variables subject to continuous domain, i.e., linearly-relaxed (POCHET; WOLSEY, 2006). This procedure is done subsequently, incrementing i until it reaches the value of I. Note that the main benefit of this method is to solve smaller subproblems at each step, possibly easier to solve than the initial problem, which reduces the computational effort required at first (OLIVEIRA et al., 2014). If all subproblems i are feasible, then the R&F solution is valid for the original model (FERREIRA; MORABITO; RANGEL, 2008); otherwise, the heuristic failed and it must be interrupted in the next iteration. This does not mean that the problem is originally infeasible, just that there is no solution for the current subproblem, considering the previously fixed variables.

In our problem, we initially tried to partition integer variables by aircraft, however, we ascertained this approach would require many iterations (see the size of V in Table 4 at Section 3.4) with non-trivial subproblems to be solved. Another alternative was to partition the variables in relation to the requests, which would lead to a much smaller number of iterations than the previous idea. Let $\mathcal{I} = \{1, \ldots, I\}$ be the set of partitions conducted by R&F, $\mathcal{A}$ be the list of requests $r \in \mathcal{L} \cup \mathcal{M}$ sorted in non-descending order (for a forward temporal strategy) or in non-ascending order (for a backward temporal strategy) by request's starting time with anticipation rl_r ($rl_r = st_r$, if $r \in \mathcal{L}$; and $rl_r = st_r - \Delta_{\mathcal{M}}$, if $r \in \mathcal{M}$), and $\mathcal{D}_i \subset \mathcal{A}$ be the subset of requests from partition i ($|\mathcal{D}_i| \cong \frac{|\mathcal{A}|}{I}$). In favor of obtaining a higher quality solution at each iteration $i' \in \mathcal{I}$, we propose to not completely fix the schedule (referring to y^v_{rs}), i.e., keep simultaneously the allocation and sequencing of flights, but rather, preserve just one of them. For this, we resort to two different mathematical artifices, both shown below.

Artifice 1 – fix the assignment among flight and aircraft, allowing the flight precedence:

$$\sum_{\substack{s \in \mathcal{R}: \\ s \neq r}} y^v_{rs} = X^1_{vr};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; i = i'; \tag{213}$$

where

$$X^1_{vr} = \begin{cases} 1, & \text{if aircraft } v \text{ is assigned to request } r; \\ 0, & \text{otherwise.} \end{cases}$$

Artifice 2 – fix the flight precedence, enabling the assignment:

$$\sum_{v \in \mathcal{V}} y^v_{rs} = X^2_{rs};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; i = i'; \tag{214}$$

where

$$X^2_{rs} = \begin{cases} 1, & \text{if request } r \text{ immediately precedes request } s; \\ 0, & \text{otherwise.} \end{cases}$$

When opting *Artifice* 1, we include variables X^1_{vr} and constraints (213) in the optimization model so that assignments between flights and aircraft are fixed according to their previous solutions, defined by $\overline{X}^1_{vr}$. On the other hand, *Artifice* 2 results in adding variables X^2_{rs} and constraints (214), bringing about fixing the flight precedence in agreement to their previously obtained values, $\overline{X}^2_{rs}$.

Knowing the possibility of infeasibility by fixing binary variables during iterations, we realized that these artifices should undergo a reformulation for partitions $i = 1, \ldots, i' - 1$ (related to the fixed variables), in such a manner that subproblems always generate feasible solutions. Therefore, we incorporate variables $rgX^{1-}_{vr} \in [0,1]$, $rgX^{1+}_{vr} \in [0,1]$, $rgX^{2-}_{rs} \in [0,1]$ and $rgX^{2+}_{rs} \in [0,1]$ to replace the constraints (213) and (214) by (215) and (216), and the objective function (189) by (217), respectively, being the total cost $penTrat$ used to penalize the activation of these variables when one of the artifices is chosen, which eliminates the chance of failure due to infeasibility of the heuristic method.

$$\sum_{\substack{s\in\mathcal{R}:\\ s\neq r}} y_{rs}^{v} = X_{vr}^{1} - rgX_{vr}^{1-} + rgX_{vr}^{1+}; \quad \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; i = 1,\dots,i^{'}-1 \mid i^{'} > 1; \tag{215}$$

$$\sum_{v\in\mathcal{V}} y_{rs}^{v} = X_{rs}^{2} - rgX_{rs}^{2-} + rgX_{rs}^{2+};\ \ \forall\ r \in \mathcal{D}_i; \quad s \in \mathcal{L}\cup\mathcal{M} \mid r \neq s; i = 1,\dots,i^{'}-1 \mid i^{'} > 1; \tag{216}$$

$$\begin{aligned}\min \quad & \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}}\sum_{\substack{s\in\mathcal{R}:\\ r\neq s}} Cf_{rs}^{v}.y_{rs}^{v} + \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{L}}\sum_{\substack{s\in\mathcal{R}:\\ r\neq s}} Cup_{r}^{v}.y_{rs}^{v} + \sum_{r\in\mathcal{L}} Cout_r.out_r \\ & + \left[\sum_{r\in\mathcal{L}\cup\mathcal{M}} c_{\hat{p}^r}.overPerc.(overR_r + overF_r) + \sum_{v\in\mathcal{V}} c_{\check{p}^v}.overPerc.over0_v\right] \\ & + penTrat.\end{aligned} \tag{217}$$

where

$$penTrat = \begin{cases} \displaystyle\sum_{i=1}^{i^{'}-1:i^{'}>1}\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{D}_i\cap\mathcal{L}} BigM.(rgX_{vr}^{1-} + rgX_{vr}^{1+}), & \text{if } \textit{Artifice } 1 \text{ is chosen;} \\ \displaystyle\sum_{i=1}^{i^{'}-1:i^{'}>1}\sum_{r\in\mathcal{D}_i}\sum_{\substack{s\in\mathcal{R}:\\ r\neq s}} BigM.(rgX_{rs}^{2-} + rgX_{rs}^{2+}), & \text{else if } \textit{Artifice } 2 \text{ is chosen.} \end{cases}$$

Bear in mind that the model is made feasible by regretting fixing X_{vr}^{1} or X_{rs}^{2}. At *Artifice* 1, if $\overline{X}_{vr}^{1} = 1$, variable rgX_{vr}^{1-} assumes value of 1 on constraint (215) when the solver verifies that it is not possible to allocate request r in aircraft v. If $\overline{X}_{vr}^{1} = 0$, variable rgX_{vr}^{1+} can have value 1 in an infeasibility circumstance. As rgX_{vr}^{1-} and rgX_{vr}^{1+} are penalized in objective function (217), they are not activated unduly, forcing (215) to comply with the established fixations. These logics are analogous to *Artifice* 2, in which rgX_{rs}^{2-} and rgX_{rs}^{2+} are able to activate or not due to the viability of $\overline{X}_{rs}^{2}$ on (216). It is important to note that these regret variables do not need to be declared as binary since constraints (215) and (216) guarantee that their values will always be 0 or 1.

Another adaptation to the traditional R&F was how many partitions of linearly-relaxed variables to consider (defined by *qtR*). This option is useful in situations where we have optimization models that are heavily loaded with variables, which could make it difficult to solve even in the first iterations of R&F. If $qtR = 0$, the proposed R&F approach reduces to a typical rolling-horizon heuristic.

The general pseudocode that represents our relax-and-fix heuristic is described in Algorithm 1.

3.3.2 Fix-and-optimize heuristic

A MIP-based heuristic, kind of like fix-and-optimize (F&O), consists of starting from an initial feasible solution, and then looking for new feasible solutions with better quality.

Algorithm 1: *Application of the relax-and-fix heuristic*

Input: problem instance.

Let M-RF$^{i'}$ be the model derived from formulation (22)-(208), which will be solved by R&F at each iteration i', and $F^{i'}$, $S^{i'}$, be their respective objective function (217) and solution $(\overline{y})$;

Let Ω^i, $\forall\ i \in \mathcal{I}$ and Φ be auxiliary sets (initializing $\Omega^i \leftarrow \emptyset$, $\forall\ i \in \mathcal{I}$ and $\Phi \leftarrow \emptyset$);

Define how the list of requests $\mathcal{A}$ will be sorted (forward or backward) and partition it into subsets $\mathcal{D}_i$, $\forall\ i \in \mathcal{I}$;

Choose between *Artifices* 1 and 2, and then determine qtR;

foreach $i' \in \mathcal{I}$ **do**

- **if** $Artifice = 1$ **then**
  - Include decision variables X^1_{vr}, rgX^{1-}_{vr}, rgX^{1+}_{vr} and constraints (213), (215) in model M-RF$^{i'}$;
  - **for** $i = 1$ **to** $i' - 1$, step $+1$, if $i' > 1$ **do** `// auxiliary and fixed variables`
    - $$\Omega^i \leftarrow \left\{ \begin{array}{l} X^1_{vr} \leftarrow \overline{X}^1_{vr} \in S^{i'-1};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; \\ rgX^{1-}_{vr} \in [0,1];\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; \\ rgX^{1+}_{vr} \in [0,1];\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}. \end{array} \right\};$$
  - $\Phi \leftarrow \{X^1_{vr} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_{i'} \cap \mathcal{L}\}$; `// part of integrality variables`
- **else** `// Artifice = 2`
  - Include decision variables X^2_{rs}, rgX^{2-}_{rs}, rgX^{2+}_{rs} and constraints (214), (216) in model M-RF$^{i'}$;
  - **for** $i = 1$ **to** $i' - 1$, step $+1$, if $i' > 1$ **do** `// auxiliary and fixed variables`
    - $$\Omega^i \leftarrow \left\{ \begin{array}{l} X^2_{rs} \leftarrow \overline{X}^2_{rs} \in S^{i'-1};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ rgX^{2-}_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ rgX^{2+}_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s. \end{array} \right\};$$
  - $\Phi \leftarrow \{X^2_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s\}$; `// part of integrality variables`
- `// Except for artifice variables, after integralization, the other (originals) binary variables remain integers in partitions` $i < i'$`, thus guaranteeing feasibility`
- **for** $i = 1$ **to** $i' - 1$, step $+1$, if $i' > 1$ **do**
  - $$\Gamma^i \leftarrow \left\{ \begin{array}{l} y^v_{rs} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i; s \in \mathcal{R} \mid r \neq s; \\ E0_{vr} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; \\ E1_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E2_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E12_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ B^f_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; f = 1, \ldots, 4; \\ DA_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{M} \mid r \neq s; \\ zl^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{D}_i \cap \mathcal{L}; v \in \mathcal{V}; t = 1, \ldots, nTW_v; \\ zf^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{D}_i; v \in \mathcal{V}; t = 1, \ldots, nTW_v. \end{array} \right\} \cup \Omega^i;$$
- $$\Gamma^{i'} \leftarrow \left\{ \begin{array}{l} y^v_{rs} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_{i'} \cup \{0\}; s \in \mathcal{R} \mid r \neq s; \\ E0_{vr} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_{i'} \cap \mathcal{L}; \\ E1_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E2_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E12_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ B^f_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; f = 1, \ldots, 4; \\ DA_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; s \in \mathcal{M} \mid r \neq s; \\ zl^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'} \cap \mathcal{L}; v \in \mathcal{V}; t = 1, \ldots, nTW_v; \\ zf^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{D}_{i'}; v \in \mathcal{V}; t = 1, \ldots, nTW_v. \end{array} \right\} \cup \Phi;$$ `// integrality variables`
- **for** $i = i' + 1$ **to** $i' + qtR$, step $+1$, if $i < I \wedge qtR > 0$ **do** `// linearly-relaxed variables`
  - $$\Gamma^i \leftarrow \left\{ \begin{array}{l} y^v_{rs} \in [0,1];\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i; s \in \mathcal{R} \mid r \neq s; \\ E0_{vr} \in [0,1];\ \forall\ v \in \mathcal{V}; r \in \mathcal{D}_i \cap \mathcal{L}; \\ E1_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E2_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E12_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ E_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s \\ B^f_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; f = 1, \ldots, 4; \\ DA_{rs} \in [0,1];\ \forall\ r \in \mathcal{D}_i; s \in \mathcal{M} \mid r \neq s; \\ zl^{vt}_r \in [0,1];\ \forall\ r \in \mathcal{D}_i \cap \mathcal{L}; v \in \mathcal{V}; t = 1, \ldots, nTW_v; \\ zf^{vt}_r \in [0,1];\ \forall\ r \in \mathcal{D}_i; v \in \mathcal{V}; t = 1, \ldots, nTW_v. \end{array} \right\};$$
- `// Variables disregarded in linear relaxation depending on the choice of` qtR
- $\Gamma^i \leftarrow \emptyset;\ \forall\ i = i' + qtR + 1, \ldots, I$;
- Build and solve M-RF$^{i'}$;
- From $S^{i'}$, save the solution obtained from integer partition i' $(\overline{X}^1_{vr} \leftarrow X^1_{vr}$ or $\overline{X}^2_{rs} \leftarrow X^2_{rs})$;

return F^I, S^I;

This approach was initially proposed by Pochet and Wolsey (2006) under the name *Exchange*, later referred to as fix-and-optimize by Sahling et al. (2009). As in R&F, the set of integer variables present in the original problem Ψ is partitioned into J subsets, being them, Ψ^j, $\forall\, j = 1, \ldots, J$. For each iteration j, the provided feasible solution is used to fix variables contained in $\Omega = \Psi \backslash \Psi^j$, while the remaining subset Ψ^j is put again as integer variables to perform reoptimization. The current solution is then changed by F&O, if the solution found by heuristic is better than the incumbent solution. In this method, observe that there is no risk of infeasibility during the iterations, since the provided solution is already feasible and makes up its pool of solutions. We also find it relevant to point out that the number of partitions into F&O (J) does not necessarily have to be of the same size as R&F, and which the larger $|\Psi^j|$, the better the solution quality tends to be.

In our F&O heuristic, we define the following notation: G, the total R&F partitions to be reoptimized at each F&O iteration; $\mathcal{J} = \{1, \ldots, J\}$, the set of partitions considered on F&O, (note that, the way we structure it, a partition j can be composed of one or more partitions i of R&F; with this in mind, $J = I - G + 1 \mid 0 < G < I$); and $\mathcal{O}_j = \bigcup_{i=j}^{j+G-1} \mathcal{D}_i$, subset of requests from partition j.

In accordance with artifices established in R&F, only variables X^1_{vr} or X^2_{rs} are reoptimized and fixed throughout iterations $j^{'} \in \mathcal{J}$, maintaining the integrity of remaining binary variables (original from the problem). Obviously, all variables could be fixed in the respective partitions, however, we found that the gain in solution quality prevailed over the reduction in runtime at computational tests. As there is no risk of infeasibility, both variables rgX^{1-}_{vr}, rgX^{1+}_{vr}, rgX^{2-}_{rs}, rgX^{2+}_{rs}, and constraints (215), (216), are no longer needed. The objective function used is again (189).

The representative pseudocode for applying fix-and-optimize to our problem is given in Algorithm 2.

We conclude this subsection with Figure 16. It illustrates how the R&F-F&O approach would behave by choosing $I = 5$, $qtR = 2$ and $G = 2$.

3.4 Computational experiments

The company provided six different month intervals of journey logs, where the first four correspond to the same data used by Munari and Alvarez (2019), not containing the pilots' time windows, and the remaining two months, encompassing all content pertinent to the crew rules covered here. Specifically, the first month comprises 10 days of operation and a total of 112 requests (including customer requests and maintenance events); the second involves 10 days and 129 requests; the third consists of 8 days and 107 requests; the fourth has 16 days and 578 requests; the fifth gives, a higher demand period, 17 days and 730 requests; and the sixth with 16 days and 275 requests.

As proposed in Munari and Alvarez (2019), we cluster the flights of each month in

Algorithm 2: *Application of the fix-and-optimize heuristic*

Input: problem instance, parameters $(I, \mathcal{D}_i)$, initial solution (S^0) of R&F.

Let M-FO$^{j'}$ be the model derived from formulation (22)-(208), which will be solved by F&O at each iteration j', and $F^{j'}$, $S^{j'}$, be their respective objective function (189) and solution $(\overline{y})$;

Let Ω and Φ be auxiliary sets (initializing $\Omega \leftarrow \emptyset$ and $\Phi \leftarrow \emptyset$);

Determine G $(0 < G < I)$, and from that make $J \leftarrow I - G + 1$ and $\mathcal{O}_j \leftarrow \bigcup_{i=j}^{j+G-1} \mathcal{D}_i$;

Choose between *Artifices* 1 and 2;

foreach $j' \in \mathcal{J}$ **do**

  if *Artifice* $= 1$ **then**

    Include decision variables X^1_{vr} and constraints (213) in model M-FO$^{j'}$;

    `// Part of the variables integralited again`

    $\Phi \leftarrow \left\{X^1_{vr} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{O}_{j'} \cap \mathcal{L}\right\}$;

    `// Fixed variables`

    $\Omega \leftarrow \left\{X^1_{vr} \leftarrow \overline{X}^1_{vr} \in S^{j'-1};\ \forall\ v \in \mathcal{V}; r \in \mathcal{L} \backslash (\mathcal{O}_{j'} \cap \mathcal{L})\right\}$;

  else `// Artifice` $= 2$

    Include decision variables X^2_{rs} and constraints (214) in model M-FO$^{j'}$;

    `// Part of the variables to be integralited again`

    $\Phi \leftarrow \left\{X^2_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{O}_{j'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s\right\}$;

    `// Fixed variables`

    $\Omega \leftarrow \left\{X^2_{rs} \leftarrow \overline{X}^2_{rs} \in S^{j'-1};\ \forall\ r \in (\mathcal{L} \cup \mathcal{M}) \backslash \mathcal{O}_{j'}; s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s\right\}$;

  `// Determining the set that contains all variables (`Ψ`)`

$$\Psi \leftarrow \left\{\begin{array}{l} y^v_{rs} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r, s \in \mathcal{R} \mid r \neq s; \\ E0_{vr} \in \{0,1\};\ \forall\ v \in \mathcal{V}; r \in \mathcal{L}; \\ E1_{rs} \in \{0,1\};\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ E2_{rs} \in \{0,1\};\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ E12_{rs} \in \{0,1\};\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ E_{rs} \in \{0,1\};\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; \\ B^f_{rs} \in \{0,1\};\ \forall\ r, s \in \mathcal{L} \cup \mathcal{M} \mid r \neq s; f = 1, \ldots, 4; \\ DA_{rs} \in \{0,1\};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}; s \in \mathcal{M} \mid r \neq s; \\ zl^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{L}; v \in \mathcal{V}; t = 1, \ldots, nTW_v; \\ zf^{vt}_r \in \{0,1\};\ \forall\ r \in \mathcal{R}; v \in \mathcal{V}; t = 1, \ldots, nTW_v. \end{array}\right\} \cup \Phi \cup \Omega;$$

  Build and solve M-FO$^{j'}$;

  From $S^{j'}$, save the solution obtained from integer partition j' $(\overline{X}^1_{vr} \leftarrow X^1_{vr}$ or $\overline{X}^2_{rs} \leftarrow X^2_{rs})$;

return F^J, S^J;

instances covering three days of operation each, which is compatible with the company's planning horizon (up to three days). Also, this is the usual choice in the planning process of flights for fractional ownership programs (ZWAN; WILS; GHIJS, 2011).

Table 4 presents each created instance, named in the format Mx_$y$to$z$ in the first column, where x is an identifier for the month, and y and z portray the first and last day considered into the month in question. For example, M2_3to5 is the instance built with data from the 3rd to the 5th day provided in the second month. Columns 2 to 8 provide the number of airports (K), the number of aircraft (V), the number of aircraft types (P), the maximum number of time windows for an aircraft ($mTW = max_{v \in \mathcal{V}}\{nTW_v\}$), the number of live requests ($L = |\mathcal{L}|$), the number of maintenance requests ($M = |\mathcal{M}|$), and the total number of requests ($R = |\mathcal{L} \cup \mathcal{M}|$). Finally, the last two columns show the percentage of customer ($\%L$) and maintenance ($\%M$) requests out of R. At the end of each month, we calculate the arithmetic mean of each column ("Avg Mx").

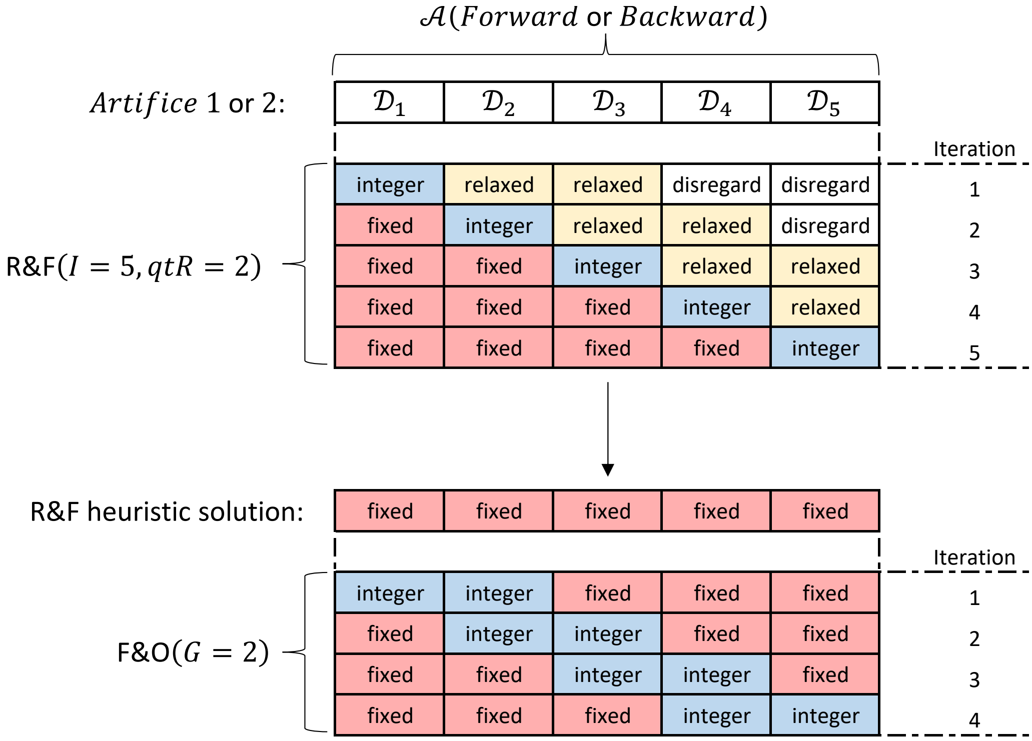
This section is divided into Subsections 3.4.1, 3.4.2 and 3.4.3. In the first, we show a toy

Table 4 – Main information of the 65 real-life-based instances provided by the company.

Instance	K	V	P	mTW	R	L	M	$\%L$	$\%M$
M1_1to3	26	21	6	-	36	14	22	38.9%	61.1%
M1_2to4	31	16	5	-	34	17	17	50.0%	50.0%
M1_3to5	32	15	5	-	40	24	16	60.0%	40.0%
M1_4to6	33	22	6	-	42	24	18	57.1%	42.9%
M1_5to7	31	23	6	-	39	22	17	56.4%	43.6%
M1_6to8	31	19	5	-	31	15	16	48.4%	51.6%
M1_7to9	29	15	5	-	28	14	14	50.0%	50.0%
M1_8to10	27	16	5	-	23	11	12	47.8%	52.2%
Avg M1	**30.0**	**18.4**	**5.4**	**-**	**34.1**	**17.6**	**16.5**	**51.6%**	**48.4%**
M2_1to3	25	19	5	-	27	8	19	29.6%	70.4%
M2_2to4	30	21	6	-	33	11	22	33.3%	66.7%
M2_3to5	30	19	6	-	37	13	24	35.1%	64.9%
M2_4to6	35	25	6	-	50	17	33	34.0%	66.0%
M2_5to7	34	26	6	-	49	18	31	36.7%	63.3%
M2_6to8	30	23	6	-	43	14	29	32.6%	67.4%
M2_7to9	25	21	6	-	38	10	28	26.3%	73.7%
M2_8to10	30	22	6	-	41	12	29	29.3%	70.7%
Avg M2	**29.9**	**22.0**	**5.9**	**-**	**39.8**	**12.9**	**26.9**	**32.4%**	**67.6%**
M3_1to3	64	27	7	-	60	49	11	81.7%	18.3%
M3_2to4	58	28	7	-	51	44	7	86.3%	13.7%
M3_3to5	51	26	6	-	41	36	5	87.8%	12.2%
M3_4to6	38	21	5	-	28	26	2	92.9%	7.1%
M3_5to7	36	16	6	-	30	28	2	93.3%	6.7%
M3_6to8	29	13	6	-	25	24	1	96.0%	4.0%
Avg M3	**46.0**	**21.8**	**6.2**	**-**	**39.2**	**34.5**	**4.7**	**88.1%**	**11.9%**
M4_1to3	77	48	4	-	108	67	41	62.0%	38.0%
M4_2to4	67	48	5	-	97	58	39	59.8%	40.2%
M4_3to5	69	50	5	-	98	62	36	63.3%	36.7%
M4_4to6	77	51	5	-	103	68	35	66.0%	34.0%
M4_5to7	83	51	5	-	109	70	39	64.2%	35.8%
M4_6to8	81	50	4	-	103	63	40	61.2%	38.8%
M4_7to9	88	49	4	-	113	72	41	63.7%	36.3%
M4_8to10	83	44	5	-	101	68	33	67.3%	32.7%
M4_9to11	90	49	5	-	110	76	34	69.1%	30.9%
M4_10to12	93	52	5	-	117	80	37	68.4%	31.6%
M4_11to13	93	51	5	-	124	82	42	66.1%	33.9%
M4_12to14	88	52	5	-	120	71	49	59.2%	40.8%
M4_13to15	87	53	5	-	112	62	50	55.4%	44.6%
M4_14to16	83	48	5	-	104	56	48	53.8%	46.2%
Avg M4	**82.8**	**49.7**	**4.8**	**-**	**108.5**	**68.2**	**40.3**	**62.9%**	**37.1%**
M5_1to3	112	54	5	1	130	87	43	66.9%	33.1%
M5_2to4	113	51	5	1	138	93	45	67.4%	32.6%
M5_3to5	123	54	5	1	137	99	38	72.3%	27.7%
M5_4to6	117	56	4	1	131	94	37	71.8%	28.2%
M5_5to7	117	55	5	1	146	96	50	65.8%	34.2%
M5_6to8	107	55	5	1	143	88	55	61.5%	38.5%
M5_7to9	113	56	5	1	149	92	57	61.7%	38.3%
M5_8to10	113	56	5	1	122	85	37	69.7%	30.3%
M5_9to11	124	54	5	1	132	99	33	75.0%	25.0%
M5_10to12	129	56	5	1	140	106	34	75.7%	24.3%
M5_11to13	128	55	5	1	142	102	40	71.8%	28.2%
M5_12to14	110	51	5	1	136	86	50	63.2%	36.8%
M5_13to15	99	50	5	1	115	67	48	58.3%	41.7%
M5_14to16	95	51	5	1	112	70	42	62.5%	37.5%
M5_15to17	89	45	5	1	103	69	34	67.0%	33.0%
Avg M5	**112.6**	**53.3**	**4.9**	**1.0**	**131.7**	**88.9**	**42.9**	**67.5%**	**32.5%**
M6_1to3	87	47	4	1	98	79	19	80.6%	19.4%
M6_2to4	78	42	5	1	77	64	13	83.1%	16.9%
M6_3to5	69	42	5	1	74	65	9	87.8%	12.2%
M6_4to6	62	36	5	1	58	53	5	91.4%	8.6%
M6_5to7	66	34	4	1	60	57	3	95.0%	5.0%
M6_6to8	59	29	4	1	47	47	0	100.0%	0.0%
M6_7to9	50	24	4	1	44	44	0	100.0%	0.0%
M6_8to10	46	23	4	1	35	35	0	100.0%	0.0%
M6_9to11	50	23	4	1	33	33	0	100.0%	0.0%
M6_10to12	50	20	4	1	32	32	0	100.0%	0.0%
M6_11to13	43	18	4	1	31	31	0	100.0%	0.0%
M6_12to14	51	23	4	1	37	37	0	100.0%	0.0%
M6_13to15	52	24	3	1	36	36	0	100.0%	0.0%
M6_14to16	52	25	4	1	34	34	0	100.0%	0.0%
Avg M6	**58.2**	**29.3**	**4.1**	**1.0**	**49.7**	**46.2**	**3.5**	**93.0%**	**7.0%**

Source: Own authorship.

Figure 16 – Application of the R&F-F&O approach when choosing $I = 5$, $qtR = 2$ and $G = 2$.



Source: Own authorship.

problem based on part of a real instance with the intention of bringing a numerical example for a better understanding of the crew rules contained in the problem. In the second, we present and discuss the results of computational experiments with the optimization models defined in Subsections 3.2.1 and 3.2.2 using a general-purpose MIP solver and real-life data provided by the company involved in our study. Likewise, we expose the results of relax-and-fix and fix-and-optimize heuristics from Section 3.3 and compare these approaches with each other, and then, with the proposed model, as finding out which is the most suitable for the operation.

All experiments were executed on a PC with processor Intel Core i7-4790 3.6 GHz CPU and 16 GB RAM. The optimization models and MIP-based heuristics were solved by the commercial software IBM CPLEX Optimization Studio version 12.10, and coded using the Optimization Programming Language (OPL) interface. We imposed a time limit of one hour and kept the default tolerance of CPLEX for the relative optimality gap (i.e., 0.01%).

3.4.1 Toy problem

To illustrate the impact of including the crew regulation, we present two Gantt charts in Figures 17, depicting optimal solutions for a toy problem from part of a real instance. Specifically, Figure 17a demonstrates an optimal solution found without considering the crew rules, therefore referring to the use of base model, whereas Figure 17b shows an optimal solution satisfying crew rules, consistent with the use of the proposed model. In both figures, the charts' vertical axis portrays the available fleet (dummy tail numbers: "OE-AAA", "OE-BBB", "OE-CCC" and "OE-DDD"), while the horizontal axis represents the time horizon (from 0 to 2800 minutes). For each aircraft, there are flight sequences composed by rectangles with different colors. In this design, customer requests (live legs) are blue, maintenance requirements are green, positioning flights (ferry legs) are red, turnaround times (*tat*) are gray, crew presentation times (PRE and POS) are orange and minimum rests (*minRest*) are yellow rectangles (ground times are the gaps among one request and another, for this reason, they are omitted). With the exception of events *tat*, PRE and POS, which have same lengths ($tat = 20$ min, $PRE = 40$ min and $POS =$ 30 min), we plot data labels to indicate the event durations (in minutes). About rectangles standing for customer and maintenance requests, the number inside round brackets (before semicolon mark) denotes their respective identification indices.

Comparing Figures 17a and 17b, we note that the sequences and assignments modified slightly by including crew requirements. In addition to changing the request start times (many of them were delayed), there was an increase in a new positioning flight due to customer request 14 having been allocated on aircraft "OE-DDD" (formerly belonging to "OE-AAA"). In Figure 17b, we have interesting situations to exemplify some of crew rules. In aircraft "OE-AAA", PRE is placed after the maintenance event (request 1), as a workday only starts with a travel event (the ferry before customer request 13). In "OE-BBB", PRE shows the starting of crew work (originated by request 3). Between requests 3 and 8 there is a ground time of 276 min. By the split duty rule, this value would be in range 2 (see Table 3), ergo, the counted time becomes $(276-90)/2+90 = 183$ min, resulting in a worked total time up to request 8 (U_8) of 398 min ($40 + 20 + 87 + 183 + 20 + 28 + 20$). After fulfilling request 8, the schedule must close the crew workday, including $POS + minRest + PRE$ (remembering that $minRest = 600$ min). Note that if this aircraft continued from request 8 without rest and went to request 11, we would have $U_{11} = U_8 + 250$ (live leg duration) + 741 (ground time existing among requests 8 and 11, as shown in Figure 17a) + 20 + 24 (ferry leg duration) + 20 = 983 min, which exceeds $maxDuty = 780$ min $= 13$ h. At "OE-CCC", after carrying out the ferry of 242 min, there is a maintenance lasting 930 min. As this time is much longer than the inclusion of minimum rest ($30 + 600 + 40 = 670$ min), this event allows the crew to rest without need to add more time to it. Starting from request 9 and going up to request 10, we have a split duty again, but now, belonging to range 3, ($499 > 360$), thus the new ground time is 60 min, computing $U_{10} = 236$ min ($40 + 20 + 96 + 60 + 20$). At long last, in "OE-DDD", between requests 4 and 2, a rest of only 200 min was included because the maintenance's duration of 400 min with ground time of 269 min already equals 669 min, consequently needing to delay request 2 in 1 minute. From request 2 until the workday ending with request 14, the total time required is $40+251+88+20+60+20+291 = 770$ min, which does not go beyond $maxDuty$. On the other hand, if aircraft "OE-AAA" had made request 14 in its place (by Figure 17a, $247+20+210+70+20+261 = 858$ min), the time exceeded (overtime) would be $858 - 780 = 78$ min. Thus, the advantage of making an extra ferry leg lasting 60 min is that it costs less than paying overtime for 78 min.

3.4.2 Results of the base and proposed models

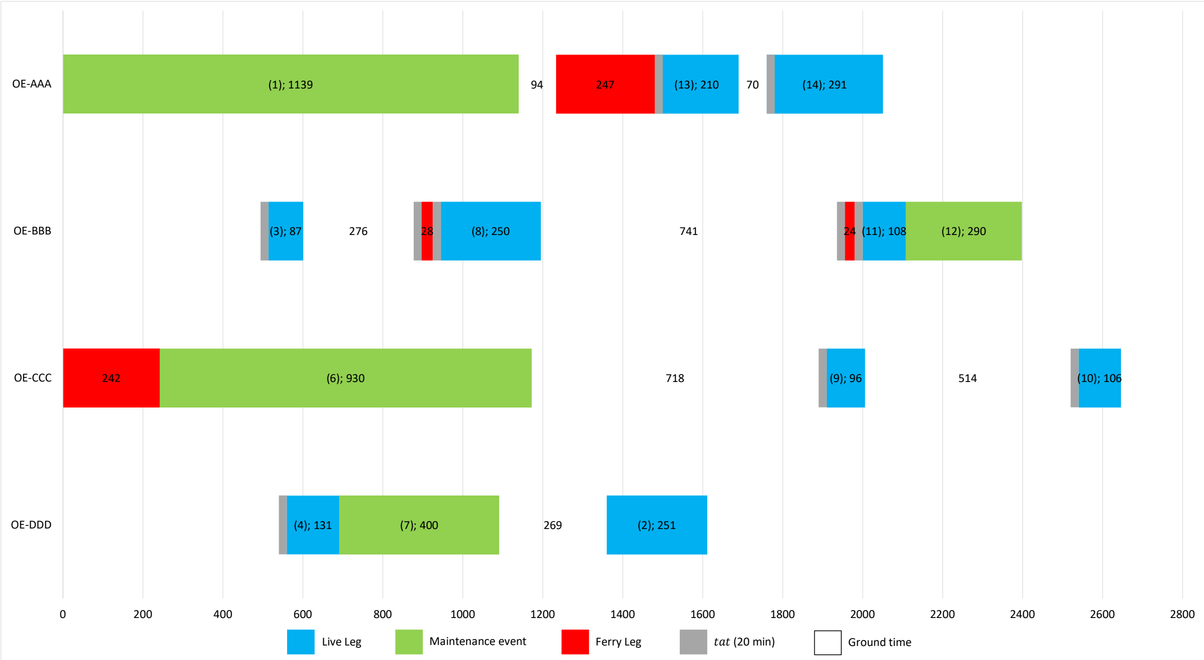
First, for comparison purposes, we present the results of solving instances in months M1 to M4 using the base model in Table 5, including only the possibility of outsourcing, in order to have a greater feasibility condition in the instances (i.e., we have $out_r = 1$ when it is not possible to allocate all customer requests). In addition to the first column that identifies the instance, under the header "Operational Costs", the three next columns show the costs of positioning (Cf), upgrade (Cup) and outsourcing ($Cout$). In the header "Counters", $nOut$ and nF give the number of outsources and the number of ferry legs. Lastly, under the header "CPLEX B&C", we give the main information of the solutions obtained by the general-purpose MIP solver of CPLEX: the lower (OF_{lb}) and upper (OF_{ub}) bounds, the relative gap computed as $100\%.(OF_{ub} - OF_{lb})/(OF_{ub} + 10^{-10})$, the number of nodes explored on the B&C tree ($nNode$), the total iterations of the simplex method used to solve the linear relaxations ($nSimplex$), the number of constraints ($nConst$), the number of decision variables ($nVar$), the number of binary variables ($nBinVar$) and the computational time in seconds ($CPUt$). Along with the results for each instance, Table 5 shows at the end of each month the average values for the instances in that month.

As discussed in Munari and Alvarez (2019), this optimization model is capable of solving all instances in M1 to M4 optimally in short computational times (0.82 to 26.14 seconds on average), showing the efficiency of a formulation that uses a network of requests.

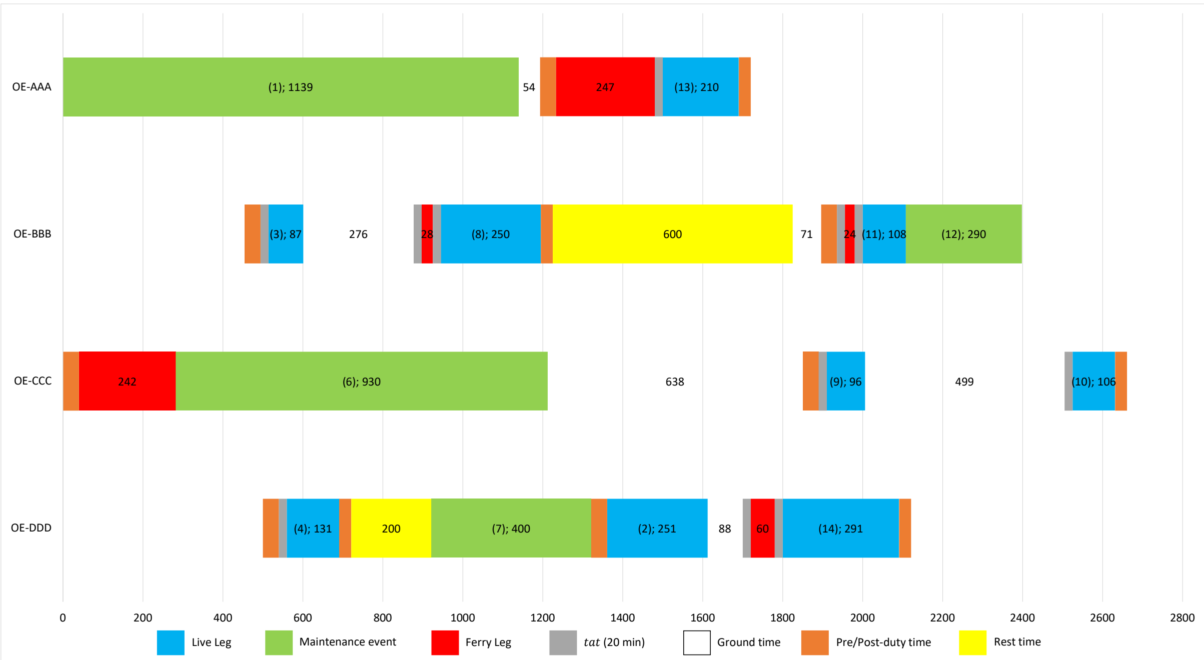
Table 6 summarizes the results obtained for months M1-M4, using the proposed model. Comparing to the previous table, we add the columns $Cover$, $nOver$ and $nRest$ to Table 6. They are, in this order, overtime cost, total overtime in minutes and number of rests.

To help us confront base model with proposed model, we built Table 7. It shows the percentage difference of columns OF_{ub} and nF from Tables 5 and 6, denoted by the columns $\%OF_{ub}$ and $\%nF$. For example, if we assume that OF^1_{ub} is a column of base model, and OF^2_{ub} belongs to proposed model, we have $\%OF_{ub} = (OF^2_{ub} - OF^1_{ub})/OF^1_{ub}$. Besides, Table 7 calculates the absolute difference of columns $nOut$, $nNode$, $nSimplex$, $nConst$, $nVar$, $nBinVar$ and $CPUt$, described by $\Delta nOut$, $\Delta nNode$, $\Delta nSimplex$, $\Delta nConst$, $\Delta nVar$, $\Delta nBinVar$ and $\Delta CPUt$. Analogously, we have $\Delta nOut = nOut^2 - nOut^1$, for example.

Figure 17 – Analysis of a toy problem.



(a)



(b)

Source: Own authorship.

Table 5 – Computational results obtained with the base model for months M1-M4.

	Operational Costs			Counters		CPLEX B&C								
Instance	Cf	Cup	$Cout$	$nOut$	nF	OF_{lb}	OF_{ub}	gap	$nNode$	$nSimplex$	$nConst$	$nVar$	$nBinVar$	$CPUt$
M1_1to3	64,413.69	3,860.80	0.00	0	10	68,274.49	68,274.49	0.000%	0	174	2,660	28,044	27,993	1.10
M1_2to4	101,415.89	1,820.29	0.00	0	11	103,236.18	103,236.18	0.000%	0	242	2,256	19,108	19,056	0.85
M1_3to5	81,364.02	2,893.93	0.00	0	13	84,257.95	84,257.95	0.000%	0	433	2,863	24,680	24,615	1.01
M1_4to6	61,739.88	1,937.22	0.00	0	9	63,677.10	63,677.10	0.000%	0	394	3,628	39,821	39,754	1.32
M1_5to7	66,465.02	5,167.71	0.00	0	10	71,632.73	71,632.73	0.000%	0	317	3,340	35,965	35,903	1.03
M1_6to8	70,756.46	2,170.40	0.00	0	8	72,926.86	72,926.86	0.000%	0	154	2,124	18,914	18,867	0.51
M1_7to9	58,959.51	2,912.28	0.00	0	8	61,871.79	61,871.79	0.000%	0	119	1,637	12,238	12,195	0.42
M1_8to10	69,947.47	1,990.35	0.00	0	6	71,937.82	71,937.82	0.000%	0	107	1,255	8,883	8,848	0.33
Avg M1	**71,882.74**	**2,844.12**	**0.00**	**0**	**9**	**74,726.87**	**74,726.87**	**0.000%**	**0**	**243**	**2,470**	**23,457**	**23,404**	**0.82**
M2_1to3	99,768.30	3,112.39	0.00	0	9	102,880.69	102,880.69	0.000%	0	60	1,624	14,419	14,383	0.57
M2_2to4	119,755.98	6,575.92	0.00	0	12	126,331.90	126,331.90	0.000%	0	93	2,320	23,628	23,583	0.62
M2_3to5	119,703.68	6,654.25	0.00	0	11	126,357.93	126,357.93	0.000%	0	219	2,654	26,784	26,733	0.88
M2_4to6	228,120.40	4,021.96	0.00	0	19	232,142.36	232,142.36	0.000%	0	355	4,617	63,843	63,775	1.31
M2_5to7	158,376.20	0.00	0.00	0	17	158,376.20	158,376.20	0.000%	0	466	4,580	63,794	63,726	1.44
M2_6to8	241,022.33	0.00	0.00	0	16	241,022.33	241,022.33	0.000%	0	197	3,497	43,597	43,539	1.13
M2_7to9	127,410.97	2,802.51	0.00	0	11	130,213.48	130,213.48	0.000%	0	130	2,778	31,192	31,143	0.79
M2_8to10	187,461.89	13,517.26	0.00	0	13	200,979.15	200,979.15	0.000%	0	233	3,215	37,960	37,906	0.94
Avg M2	**160,202.47**	**4,585.54**	**0.00**	**0**	**14**	**164,788.01**	**164,788.01**	**0.000%**	**0**	**219**	**3,161**	**38,152**	**38,099**	**0.96**
M3_1to3	219,390.97	9,381.96	0.00	0	22	228,772.93	228,772.93	0.000%	0	899	7,275	98,957	98,847	3.73
M3_2to4	182,774.91	0.00	279,726.34	1	21	462,501.25	462,501.25	0.000%	0	668	5,973	74,380	74,284	2.53
M3_3to5	118,716.58	2,232.76	0.00	0	13	120,949.34	120,949.34	0.000%	0	437	4,214	44,876	44,798	1.49
M3_4to6	155,934.44	7,005.75	290,126.08	1	10	453,066.27	453,066.27	0.000%	0	300	2,274	17,128	17,073	0.80
M3_5to7	175,763.32	11,005.47	0.00	0	9	186,768.79	186,768.79	0.000%	0	306	2,135	14,955	14,896	0.79
M3_6to8	15,160.16	0.00	0.00	0	6	15,160.16	15,160.16	0.000%	0	165	1,493	8,513	8,463	0.65
Avg M3	**144,623.40**	**4,937.66**	**94,975.40**	**0**	**14**	**244,536.46**	**244,536.46**	**0.000%**	**0**	**463**	**3,894**	**43,135**	**43,060**	**1.67**
M4_1to3	252,724.87	3,792.75	204,528.22	1	30	461,045.84	461,045.84	0.000%	0	3,570	21,552	565,280	565,104	19.70
M4_2to4	205,585.46	15,705.56	0.00	0	24	221,276.68	221,291.02	0.006%	178	7,920	18,129	456,492	456,336	21.91
M4_3to5	202,816.86	9,256.82	0.00	0	26	212,073.68	212,073.68	0.000%	0	2,359	18,911	485,311	485,150	22.16
M4_4to6	215,138.24	5,854.93	0.00	0	32	220,974.68	220,993.17	0.008%	0	2,593	20,866	546,535	546,363	24.69
M4_5to7	253,584.96	3,909.45	0.00	0	34	257,494.41	257,494.41	0.000%	0	3,175	22,640	611,721	611,541	23.25
M4_6to8	194,775.15	6,690.24	0.00	0	25	201,465.39	201,465.39	0.000%	0	2,350	20,559	535,817	535,650	17.92
M4_7to9	243,357.56	3,664.01	0.00	0	29	247,021.57	247,021.57	0.000%	0	2,268	23,613	631,453	631,267	20.15
M4_8to10	260,620.44	2,959.86	0.00	0	27	263,580.30	263,580.30	0.000%	0	2,314	19,316	453,502	453,332	15.87
M4_9to11	312,402.20	6,838.90	0.00	0	34	319,241.10	319,241.10	0.000%	0	3,816	22,927	598,526	598,339	26.85
M4_10to12	376,997.71	21,350.54	0.00	0	42	398,311.25	398,348.25	0.009%	1,921	45,855	25,734	718,162	717,964	54.14
M4_11to13	350,013.98	11,733.17	0.00	0	46	361,747.15	361,747.15	0.000%	1	5,790	27,702	790,758	790,551	46.04
M4_12to14	307,678.68	5,862.01	0.00	0	35	313,540.69	313,540.69	0.000%	0	5,164	26,017	755,284	755,092	33.57
M4_13to15	326,447.29	19,518.61	208,926.22	1	32	554,892.12	554,892.12	0.000%	0	3,617	23,332	670,996	670,821	25.38
M4_14to16	265,439.80	10,862.69	0.00	0	29	276,302.49	276,302.49	0.000%	0	2,230	19,908	524,369	524,208	14.31
Avg M4	**269,113.09**	**9,142.82**	**29,532.46**	**0**	**32**	**307,783.38**	**307,788.37**	**0.002%**	**150**	**6,644**	**22,229**	**596,015**	**595,837**	**26.14**

Source: Own authorship.

Table 6 – Computational results obtained with the proposed model for months M1-M4.

	Operational Costs				Counters				CPLEX B&C								
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{lb}	OF_{ub}	*gap*	*nNode*	*nSimplex*	*nConst*	*nVar*	*nBinVar*	*CPUt*
M1_1to3	64,413.69	3,860.80	0.00	0.00	0	0	10	8	68,274.49	68,274.49	0.000%	0	575	28,478	41,787	39,137	4.59
M1_2to4	101,415.89	1,820.29	0.00	0.00	0	0	11	13	103,236.18	103,236.18	0.000%	0	2,370	24,032	31,126	28,865	3.53
M1_3to5	81,364.02	2,893.93	0.00	0.00	0	0	13	12	84,257.95	84,257.95	0.000%	0	4,985	31,312	40,935	38,079	8.92
M1_4to6	61,739.88	1,937.22	0.00	0.00	0	0	9	13	63,677.10	63,677.10	0.000%	0	3,098	35,736	57,969	54,796	6.36
M1_5to7	66,465.02	5,167.71	0.00	0.00	0	0	10	13	71,632.73	71,632.73	0.000%	0	2,557	31,279	51,704	48,911	5.23
M1_6to8	70,756.46	2,170.40	0.00	0.00	0	0	8	8	72,926.86	72,926.86	0.000%	0	1,280	20,547	29,014	27,072	2.60
M1_7to9	58,959.51	2,912.28	0.00	0.00	0	0	8	13	61,871.79	61,871.79	0.000%	0	1,021	16,515	20,443	18,831	2.01
M1_8to10	69,947.47	1,990.35	0.00	0.00	0	0	6	6	71,937.82	71,937.82	0.000%	0	714	11,514	14,503	13,336	1.56
Avg M1	**71,882.74**	**2,844.12**	**0.00**	**0.00**	**0**	**0**	**9**	**11**	**74,726.87**	**74,726.87**	**0.000%**	**0**	**2,075**	**24,927**	**35,935**	**33,628**	**4.35**
M2_1to3	99,768.30	3,112.39	0.00	0.00	0	0	9	7	102,880.69	102,880.69	0.000%	0	144	16,834	22,312	20,645	1.48
M2_2to4	119,755.98	6,575.92	0.00	0.00	0	0	12	11	126,331.90	126,331.90	0.000%	0	257	24,545	35,298	32,966	2.26
M2_3to5	119,703.68	6,654.25	0.00	0.00	0	0	11	10	126,357.93	126,357.93	0.000%	0	2,665	30,267	41,332	38,500	3.69
M2_4to6	228,120.40	4,021.96	0.00	0.00	0	0	19	22	232,142.36	232,142.36	0.000%	0	1,811	55,140	90,343	85,417	9.88
M2_5to7	158,376.20	0.00	0.00	0.00	0	0	17	20	158,376.20	158,376.20	0.000%	0	4,367	52,593	89,180	84,498	9.66
M2_6to8	241,022.33	0.00	0.00	0.00	0	0	16	10	241,022.33	241,022.33	0.000%	0	2,878	41,148	63,292	59,527	9.42
M2_7to9	127,410.97	2,802.51	0.00	0.00	0	0	11	9	130,213.48	130,213.48	0.000%	0	515	33,031	46,737	43,637	3.43
M2_8to10	187,461.89	13,517.26	0.00	0.00	0	0	13	15	200,979.15	200,979.15	0.000%	0	3,918	37,939	55,958	52,450	5.95
Avg M2	**160,202.47**	**4,585.54**	**0.00**	**0.00**	**0**	**0**	**14**	**13**	**164,788.00**	**164,788.00**	**0.000%**	**0**	**2,069**	**36,437**	**55,557**	**52,205**	**5.72**
M3_1to3	234,230.12	5,502.76	0.00	4,199.90	0	21	23	25	243,932.78	243,932.78	0.000%	0	7,773	64,093	134,327	129,139	29.13
M3_2to4	182,774.91	0.00	279,726.34	57,998.55	1	290	21	17	520,499.80	520,499.80	0.000%	0	4,286	46,511	100,018	96,266	13.94
M3_3to5	127,382.09	4,333.36	0.00	73,049.35	0	409	12	13	204,764.80	204,764.80	0.000%	0	4,609	30,563	61,582	59,054	10.23
M3_4to6	155,934.44	7,005.75	290,126.08	73,049.35	1	409	10	9	526,115.62	526,115.62	0.000%	0	2,090	14,529	25,003	23,721	4.25
M3_5to7	176,296.64	11,005.47	0.00	30,970.47	0	207	9	13	218,272.58	218,272.58	0.000%	0	3,028	15,751	23,789	22,362	5.29
M3_6to8	30,047.34	0.00	0.00	9,199.77	0	46	7	9	39,247.11	39,247.11	0.000%	0	1,422	10,847	14,638	13,599	3.34
Avg M3	**151,110.92**	**4,641.22**	**94,975.40**	**41,411.23**	**0**	**230**	**14**	**14**	**292,138.78**	**292,138.78**	**0.000%**	**0**	**3,868**	**30,382**	**59,893**	**57,357**	**11.03**
M4_1to3	252,051.53	5,553.53	204,528.22	0.00	1	0	30	48	462,133.28	462,133.28	0.000%	1,391	236,588	225,517	682,916	665,155	934.58
M4_2to4	209,718.69	15,705.56	0.00	38,999.03	0	195	24	37	264,423.28	264,423.28	0.000%	2,751	449,870	185,374	552,056	537,360	1,467.36
M4_3to5	228,016.23	9,256.82	0.00	0.00	0	0	27	34	237,273.05	237,273.05	0.000%	1	61,916	186,677	582,443	567,790	296.94
M4_4to6	231,804.49	5,854.93	0.00	9,199.77	0	46	33	40	246,851.83	246,859.19	0.003%	0	38,335	203,235	653,260	637,449	271.80
M4_5to7	339,504.34	5,807.25	0.00	42,998.93	0	215	38	53	388,361.91	388,373.52	0.003%	1,355	1,702,944	228,223	731,318	713,499	3,585.86
M4_6to8	266,506.57	10,769.96	0.00	42,998.93	0	215	28	45	320,275.46	320,275.46	0.000%	1,359	382,587	207,374	643,253	626,928	1,066.78
M4_7to9	265,223.68	3,664.01	0.00	33,799.16	0	169	30	41	302,686.85	302,686.85	0.000%	0	43,081	244,670	759,782	740,635	289.11
M4_8to10	329,818.71	2,959.86	0.00	0.00	0	0	29	43	332,778.57	332,778.57	0.000%	2,435	383,614	192,671	555,518	540,424	412.89
M4_9to11	322,535.28	6,838.90	297,325.90	76,798.08	1	384	34	51	703,498.16	703,498.16	0.000%	2,492	338,275	189,379	706,579	689,699	3,190.48
M4_10to12	432,191.14	19,458.92	0.00	33,199.17	0	166	46	68	484,849.23	484,849.23	0.000%	1,392	878660	258411	854818	834992	3,581.54
M4_11to13	427,212.07	11,733.17	0.00	52,198.70	0	261	52	65	491,143.94	491,143.94	0.000%	2,522	659718	289898	944411	921915	3,549.52
M4_12to14	358,477.41	5,862.01	284,259.56	18,399.54	1	92	37	57	666,998.52	666,998.52	0.000%	1,339	439,780	281,531	900,988	878,855	2,415.60
M4_13to15	365,512.98	19,518.61	208,926.22	0.00	1	0	33	42	593,957.81	593,957.81	0.000%	1,170	354,160	250,917	798,991	779,113	455.27
M4_14to16	304,318.77	14,089.14	0.00	0.00	0	0	30	39	318,407.91	318,407.91	0.000%	1,634	180,605	217,457	634,953	617,536	251.58
Avg M4	**309,492.28**	**9,795.41**	**71,074.28**	**24,899.38**	**0**	**124**	**34**	**47**	**415,259.98**	**415,261.34**	**0.000%**	**1,417**	**439,295**	**225,810**	**714,378**	**696,525**	**1,554.95**

Source: Own authorship.

Table 7 – Comparison between the main results of base and proposed model for months M1-M4.

Instance	$\%OF_{ub}$	$\Delta nOut$	$\%nF$	$\Delta nNode$	$\Delta nSimplex$	$\Delta nConst$	$\Delta nVar$	$\Delta nBinVar$	$\Delta CPUt$
M1_1to3	0.00%	0	0.00%	0	401	25,818	13,743	11,144	3.49
M1_2to4	0.00%	0	0.00%	0	2,128	21,776	12,018	9,809	2.68
M1_3to5	0.00%	0	0.00%	0	4,552	28,449	16,255	13,464	7.91
M1_4to6	0.00%	0	0.00%	0	2,704	32,108	18,148	15,042	5.04
M1_5to7	0.00%	0	0.00%	0	2,240	27,939	15,739	13,008	4.20
M1_6to8	0.00%	0	0.00%	0	1,126	18,423	10,100	8,205	2.09
M1_7to9	0.00%	0	0.00%	0	902	14,878	8,205	6,636	1.59
M1_8to10	0.00%	0	0.00%	0	607	10,259	5,620	4,488	1.23
Avg M1	**0.00%**	**0**	**0.00%**	**0**	**1,833**	**22,456**	**12,479**	**10,225**	**3.53**
M2_1to3	0.00%	0	0.00%	0	84	15,210	7,893	6,262	0.91
M2_2to4	0.00%	0	0.00%	0	164	22,225	11,670	9,383	1.64
M2_3to5	0.00%	0	0.00%	0	2,446	27,613	14,548	11,767	2.81
M2_4to6	0.00%	0	0.00%	0	1,456	50,523	26,500	21,642	8.57
M2_5to7	0.00%	0	0.00%	0	3,901	48,013	25,386	20,772	8.22
M2_6to8	0.00%	0	0.00%	0	2,681	37,651	19,695	15,988	8.29
M2_7to9	0.00%	0	0.00%	0	385	30,253	15,545	12,494	2.64
M2_8to10	0.00%	0	0.00%	0	3,685	34,724	17,998	14,544	5.01
Avg M2	**0.00%**	**0**	**0.00%**	**0**	**1,850**	**33,277**	**17,404**	**14,107**	**4.76**
M3_1to3	6.63%	0	4.55%	0	6,874	56,818	35,370	30,292	25.40
M3_2to4	12.54%	0	0.00%	0	3,618	40,538	25,638	21,982	11.41
M3_3to5	69.30%	0	-7.69%	0	4,172	26,349	16,706	14,256	8.74
M3_4to6	16.12%	0	0.00%	0	1,790	12,255	7,875	6,648	3.45
M3_5to7	16.87%	0	0.00%	0	2,722	13,616	8,834	7,466	4.50
M3_6to8	158.88%	0	16.67%	0	1,257	9,354	6,125	5,136	2.69
Avg M3	**19.47%**	**0**	**1.23%**	**0**	**3,406**	**26,488**	**16,758**	**14,297**	**9.37**
M4_1to3	0.24%	0	0.00%	1,391	233,018	203,965	117,636	100,051	914.88
M4_2to4	19.49%	0	0.00%	2,573	441,950	167,245	95,564	81,024	1,445.45
M4_3to5	11.88%	0	3.85%	1	59,557	167,766	97,132	82,640	274.78
M4_4to6	11.70%	0	3.12%	0	35,742	182,369	106,725	91,086	247.11
M4_5to7	50.83%	0	5.88%	1,355	1,699,769	205,583	119,597	101,958	3,562.61
M4_6to8	58.97%	0	12.00%	1,359	380,237	186,815	107,436	91,278	1,048.86
M4_7to9	22.53%	0	3.45%	0	40,813	221,057	128,329	109,368	268.96
M4_8to10	26.25%	0	7.41%	2,435	381,300	173,355	102,016	87,092	397.02
M4_9to11	120.37%	1	0.00%	2,492	334,459	166,452	108,053	91,360	3,163.63
M4_10to12	21.71%	0	9.52%	-529	832,805	232,677	136,656	117,028	3,527.40
M4_11to13	35.77%	0	13.04%	2,521	653,928	262,196	153,653	131,364	3,503.48
M4_12to14	112.73%	1	5.71%	1,339	434,616	255,514	145,704	123,763	2,382.03
M4_13to15	7.04%	0	3.12%	1,170	350,543	227,585	127,995	108,292	429.89
M4_14to16	15.24%	0	3.45%	1,634	178,375	197,549	110,584	93,328	237.27
Avg M4	**34.92%**	**0.14**	**5.39%**	**1,267**	**432,651**	**203,581**	**118,363**	**100,688**	**1,528.81**

Source: Own authorship.

In the first month, we observe that in all instances, it was possible to insert the breaks or rests required by the crew rules (proposed model), without modifying the optimal value obtained by disregarding such rules (base model). Therefore, there was no overtime or outsourcing in any instance. On average, in relation to base model, the proposed model resulted in an increase of computational time in 3.53 s, in the number of constraints in 22,456, and in the number of decision variables in 12,479. Again, in the second month, all instances had an optimal value equal to the base model. The average values of computational time, constraints and variables revealed an increase of 4.76 s, 33,277 and 17,404, in that order, compared to the case without considering the crew rules. Even though the average number of requests for month 3 is practically the same as for month 2, the inclusion of breaks and rests caused a change in the optimal values of all instances. This resulted in an average increase of the objective function in 19.47%, of computational time in 9.37 s, and of constraints and variables in about 26,488 and 16,758. In the third month, the solutions of two instances presented outsourcing costs when using the proposed model, but, in the same way, this is noticed in the base model. Furthermore, all instance solutions presented overtime for the proposed model.

For instances in the fourth month, there is a considerable increase in the computational difficulty, given that the average execution time was from 1.67 s to 26.14 s for base model, and from 11.03 s to 1,554.95 s for proposed model. This can be explained by the leap in size of problem between months M3 and M4. For example, in the proposed model, it was 30,382 constraints and 59,893 variables to 225,810 constraints and 714,378 variables (about 7 times more constraints and 12 times more variables). We can also see that, unlike in instances of previous months, CPLEX was not able to solve all instances of month M4 at the root node, just using cutting planes (on average, we had 1,417 explored nodes). Regarding outsourcing costs, we notice that the proposed model obtained two instances with non-zero values more than base model, and on the fourth type of cost, five instances were left without overtime in the proposed model. The average increase of the objective function with the inclusion of breaks and rests was 34.92%. In summary, we note that the rest inserts, apart from being required by the crew regulations, not significantly increase the costs of solutions in relation to the base model. In particular, the proposed model achieved solutions with gaps smaller than 0.01% for all instances from months M1-M4 at a reasonable computational time: a maximum of 3,585.86 seconds ("M4_5to7") and a global average of 608.78 seconds.

Table 8 presents the results obtained for instances of months M5 and M6. The layout of Table 8 is similar to Table 6.

The computational effort reaches its peak in the fifth month. We can observe that only 2 out of 15 instances were solved optimally. According to the central tendency, the gap, constraints and variables were in the order of 42.842%, 341,938 and 1,117,890. However, in the sixth month, the solution time drops dramatically, resembling the computational effort of months 1, 2 and 3.

Based on these results, we can conclude that the proposed optimization model can be used in general-purpose MIP solvers to relatively well solve instances with up to 200,000 constraints and 700,000 variables. Once most instances that came from real-data are below this order of magnitude, we can say that our model is adequate to the problem, having great application potential, since the planning horizon practiced by the company is in general of three days.

3.4.3 Results of the R&F and F&O heuristics

Ahead of running R&F and F&O heuristic methods in sequence to resolve the instances, we necessitate to determine the parameters: I (number of partitions for R&F), ordering of request list $\mathcal{A}$ (forward or backward), qtR (number of linearly-relaxed parti-

Table 8 – Computational results obtained with the proposed model for months M5 and M6.

	Operational Costs				Counters				Cplex B&C								
Instance	Cf	Cup	$Cout$	$Cover$	$nOut$	$nOver$	nF	$nRest$	OF_{lb}	OF_{ub}	gap	$nNode$	$nSimplex$	$nConst$	$nVar$	$nBinVar$	$CPUt$
M5_1to3	542,869.39	13,131.50	0.00	178,291.85	0	933	44	61	539,151.44	734,292.74	26.575%	71	617,980	332,434	1,082,946	1,042,093	3,631.67
M5_2to4	501,957.95	8,410.86	299,459.18	209,216.80	1	1,111	42	72	949,545.49	1,019,044.79	6.820%	1,246	347,659	368,550	1,160,109	1,114,277	3,633.37
M5_3to5	589,185.78	9,109.35	619,304.96	125,996.85	2	630	49	62	554,108.16	1,343,596.94	58.759%	838	353,137	358,347	1,199,648	1,155,371	3,635.84
M5_4to6	474,043.37	29,666.94	320,791.98	254,113.71	1	1,279	50	63	841,281.18	1,078,616.00	22.004%	2,606	470,718	331,473	1,133,062	1,092,414	3,633.11
M5_5to7	1,054,843.42	55,160.56	1,544,920.76	2,145,601.58	6	12,756	64	69	774,465.94	4,800,526.32	83.867%	109	322,623	394,939	1,384,452	1,354,064	3,639.34
M5_6to8	800,609.51	31,747.64	253,327.00	388,849.78	1	2,465	62	81	762,888.21	1,474,533.93	48.262%	215	515,029	386,882	1,329,756	1,299,703	3,637.26
M5_7to9	624,769.81	39,304.39	0.00	215,494.61	0	1,078	51	67	663,358.52	879,568.81	24.581%	1,232	256,410	441,058	1,464,210	1,409,656	3,641.40
M5_8to10	652,356.03	21,004.81	0.00	89,797.76	0	449	50	56	621,733.37	763,158.60	18.532%	2,387	479,484	278,350	985,104	964,195	3,628.80
M5_9to11	564,510.37	16,746.65	0.00	267,152.35	0	1,387	52	64	613,963.14	848,409.37	27.634%	51	410,354	367,126	1,132,566	1,108,379	3,634.28
M5_10to12	653,614.09	35,508.06	917,043.74	1,038,914.82	3	5,220	60	59	1,072,231.95	2,645,080.71	59.463%	741	416,558	369,860	1,292,032	1,246,503	3,641.11
M5_11to13	531,863.98	14,702.13	1,156,078.38	287,292.82	4	1,437	55	61	911,549.74	1,989,937.31	54.192%	1,235	374,434	385,073	1,309,091	1,261,497	3,640.53
M5_12to14	579,076.35	20,313.76	573,429.60	141,248.73	2	790	51	60	1,229,403.96	1,314,068.44	6.443%	2,361	472,923	403,447	1,147,211	1,119,449	3,631.46
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	41	1,694,978.15	1,694,978.15	0.000%	0	25,788	270,639	795,855	762,584	172.97
M5_14to16	423,780.53	9,665.52	276,889.68	101,797.45	1	509	35	47	812,133.18	812,133.18	0.000%	0	29,997	241,447	767,632	748,998	176.81
M5_15to17	415,050.12	27,253.00	0.00	120,322.38	0	614	32	50	511,488.02	562,625.50	9.089%	3,231	934,197	199,443	584,673	569,293	3,617.26
Avg M5	**588,285.31**	**22,777.39**	**481,395.94**	**371,579.42**	**2**	**2,047**	**49**	**61**	**836,818.70**	**1,464,038.05**	**42.842%**	**1,088**	**401,819**	**341,938**	**1,117,890**	**1,083,232**	**3,173.01**
M6_1to3	342,596.91	6,532.87	0.00	41,198.97	0	206	31	49	390,328.75	390,328.75	0.000%	0	20,062	172,867	548,657	535,971	201.73
M6_2to4	239,526.77	10,509.65	0.00	2,199.94	0	11	26	27	252,236.36	252,236.36	0.000%	0	11,259	108,449	310,518	302,614	83.81
M6_3to5	322,575.51	31,931.25	570,065.40	2,199.95	2	11	28	30	926,772.11	926,772.11	0.000%	24	16,646	99,298	287,166	280,083	100.39
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	17	328,431.01	328,431.01	0.000%	0	7,109	61,726	156,884	152,483	37.56
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	18	449,164.08	449,164.08	0.000%	0	4,815	63,788	159,806	155,263	24.10
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	13	134,864.25	134,864.25	0.000%	0	3,119	39,416	87,472	84,705	14.09
M6_7to9	197,684.64	3,973.06	0.00	33,399.17	0	167	12	13	235,056.87	235,056.87	0.000%	0	2,855	33,637	66,429	63,980	9.81
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	80,989.52	80,989.52	0.000%	0	1,370	22,451	41,416	39,771	4.66
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,041,799.55	1,041,799.55	0.000%	0	1,174	20,362	37,010	35,521	3.95
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	11	561,693.60	561,693.60	0.000%	0	2,466	18,592	31,429	30,024	5.87
M6_11to13	30,026.16	0.00	219,727.84	34,199.15	1	171	3	9	283,953.15	283,953.15	0.000%	0	1,250	17,119	27,402	26,076	4.31
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	9	405,466.47	405,466.47	0.000%	0	2,726	24,681	46,070	44,261	6.99
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	9	76,158.33	76,158.33	0.000%	0	1,902	23,805	45,157	43,428	5.73
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	43,598.91	43,598.91	0.000%	0	1,688	21,864	41,776	40,204	5.32
Avg M6	**150,203.40**	**6,458.92**	**188,305.18**	**27,211.99**	**1**	**142**	**13**	**16**	**372,179.50**	**372,179.50**	**0.000%**	**2**	**5,603**	**52,004**	**134,799**	**131,027**	**36.31**

Source: Own authorship.

tions to be considered), J (number of partitions for F&O) and choose one of artifices 1 and 2 (which consists of opting amongst variables X^1_{vr} and X^2_{rs}).

Through preliminary computational tests, we decided how to contemplate the following eight R&F-F&O variants in our experimentation: [$Forward$, $X1$, 3, 1, 2], [$Backward$, $X1$, 3, 1, 2], [$Forward$, $X2$, 3, 1, 2], [$Backward$, $X2$, 3, 1, 2], [$Forward$, $X1$, 4, 1, 3], [$Backward$, $X1$, 4, 1, 3], [$Forward$, $X2$, 4, 1, 3] and [$Backward$, $X2$, 4, 1, 3]. They are named in the format [order of $\mathcal{A}$, *Artifice* 1 or 2, I, qtR, J]. As the instances of months M1-M4 were well-resolved by the proposed model/CPLEX and do not have the pilot windows, we concentrated on solving the largest instances, those belonging to months M5 and M6. Since all instances of M6 were solved optimally in the proposed model, we have a good sample to evaluate the quality of heuristic approaches, and because M5 does not have good results, we can analyze whether these heuristic methods can handle instances with a higher congestion level.

To better report the superiority of one approach over the other, we created the Figure 18, 19 and 20.

Figure 18 reveals the ability of approaches to find feasible solutions within the time limit of one hour. In all, three different kinds of solution status were identified:

- "Ending": involves the situation in which the R&F constructive heuristic is capable of finishing all its iterations (finding a feasible solution) and the F&O improvement heuristic ends its routine with a B&C's gap of 0% (in this way, the total runtime is less than the established limit of 1 h);

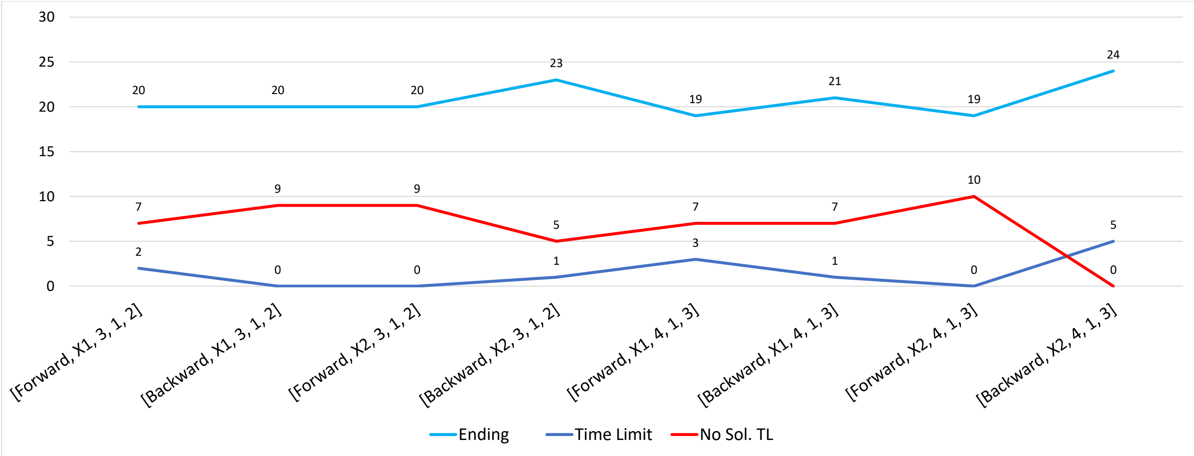
- "Time Limit": implies completing R&F, but F&O ends with a B&C's gap greater than 0% (i.e., the heuristic procedure as a whole reaches the time limit of 1 h, might have the chance to improve the results further, if more time was offered, since F&O's B&C was not fully explored);

- "No Sol. TL": indicates when the solver seeks no feasible solution within the given time limit interval to the R&F (therefore, F&O is not executed, as R&F cannot complete all its iterations).

By counting these statuses for each heuristic variant, we plot the line chart contained in Figure 18. From it, we realize that R&F-F&O[$Backward$, $X2$, 4, 1, 3] approach stands out from the others, being the only one capable to encounter feasible solutions in all instances within the time limit.

Figures 19 are responsible for comparing the heuristic variants by runtime (vertical axes). We organize the comparisons into four charts. The first two focus on the choices of $I = 3$, $qtR = 1$ and $J = 2$, where Figure 19a is dedicated to instances of M5 and Figure 19b to M6. Similarly, the last two charts aim attention at $I = 4$, $qtR = 1$ and $J = 3$, being Figure 19c directed to M5 and Figure 19d to M6. Instances with solution status

Figure 18 – Comparison of R&F-F&O approaches in relation to solution status.



Source: Own authorship.

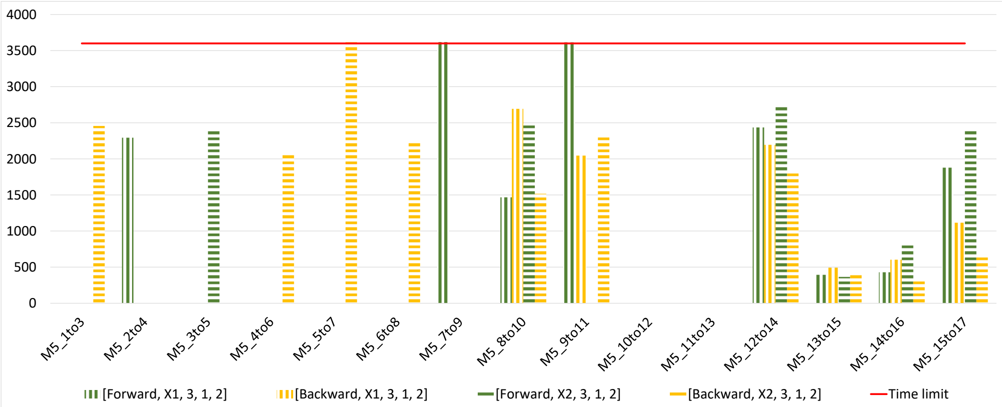
"No Sol. TL" have no vertical bars. Concerning M5, the approaches were competitive, which makes it impossible to determine a best variant. However, in M6, we witnessed that the forward temporal strategy tends to demand less computational time than the backward one.

We utilize Figures 20 to confront the upper bounds obtained from heuristic approaches. This figure set holds four graphs, organized in the same way as Figures 19. In addition to the points that situate the objective function values of heuristic variants, we add as lines the lower bounds of proposed model (blue color) and the best upper bounds found in each instance (red color) when taking into account all approaches (including the proposed model). For optimal values, the red line (UB^*) ended up taking the front of blue line (LB^*), thus we see more the upper bounds. As in Figure 19, instances with solution status "No Sol. TL" do not have points on the chart area. Solving M5 with $I = 3$, $qtR = 1$ and $J = 2$ (Figure 20a) revealed that the forward strategy generates better results than the backward one and also that *Artifice* 2 dominates 1. For M6 (Figure 20b), backward strategy is better, and this time, *Artifice* 1 and 2 are competitive. Now, choosing $I = 4$, $qtR = 1$ and $J = 3$ and solving M5 (Figure 20c), we notice a much greater predominance of backward strategy and *Artifice* 2. This is mainly due to the performance of R&F-F&O[$Backward$, $X2$, 4, 1, 3] variant. For M6 (Figure 20d), over again, backward strategy and *Artifice* 2 are slightly better.

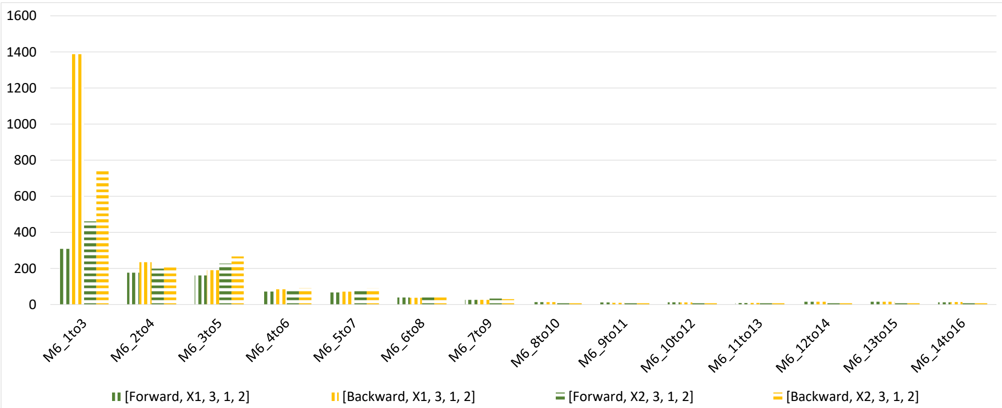
From the previous comparative analyses, the superiority of R&F-F&O[$Backward$, $X2$, 4, 1, 3] approach in relation to the others variants was apparent, proving to be the best for finding quality solutions at reasonable computational times (limited to one hour). Table 9 brings more detailed results of this heuristic variant. Headers "Operational Costs" and "Counters" have the same columns as those described in Tables 6 and 8. The next two headers, "R&F Part" and "F&O Part", show the upper bound (column OF_{ub}) reached in the last iteration (or partition), average B&C's relative gap of the iterations ($Avg\ gap$), number of completed iterations (*iter*) and elapsed time ($CPUt$). In such manner, we can observe how the results were when building a solution with R&F and improving it by applying F&O. Header "R&F-F&O" comprises the results *Avg gap*, *iter* and $CPUt$ by considering the MIP-based heuristic method as a whole. In the last column (St. Sol.), we inform the solution status. The computational results of the remaining seven heuristic variants are presented in the tables of Appendix A.2.

Ultimately, to verify whether the present heuristic variant has potential for application in practice, we proceeded our analysis by confronting its upper bound values with those of the proposed model. For that, we provide Figure 21. It has the similar layout as Figures 20. For the larger instances with the higher level of congestion, belonging to month M5 (Figure 21a), R&F-F&O[*Backward*, $X2$, 4, 1, 3] approach outperforms the proposed model, managing in most cases to obtain the best upper bounds (relatively close to the lower bounds). As regards month M6 (Figure 21b), our heuristic approach was able to achieve optimal results in the vast majority of instances (only in M6_2to4 and M6_5to7 we had relative gaps of 1.36% and 0.39%).

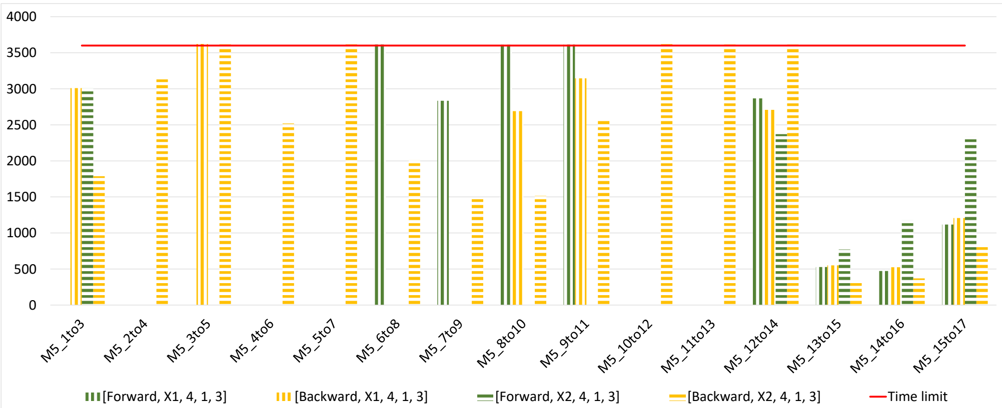
Figure 19 – Comparison of R&F-F&O approaches in relation to runtime.



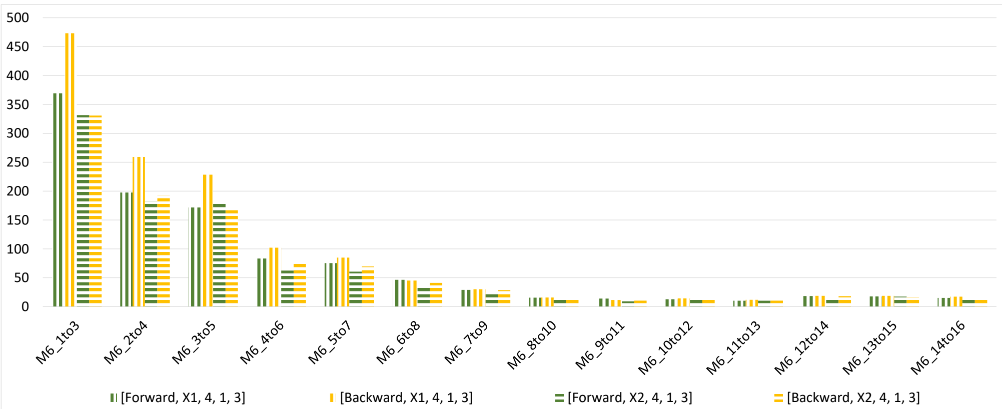
(a)



(b)



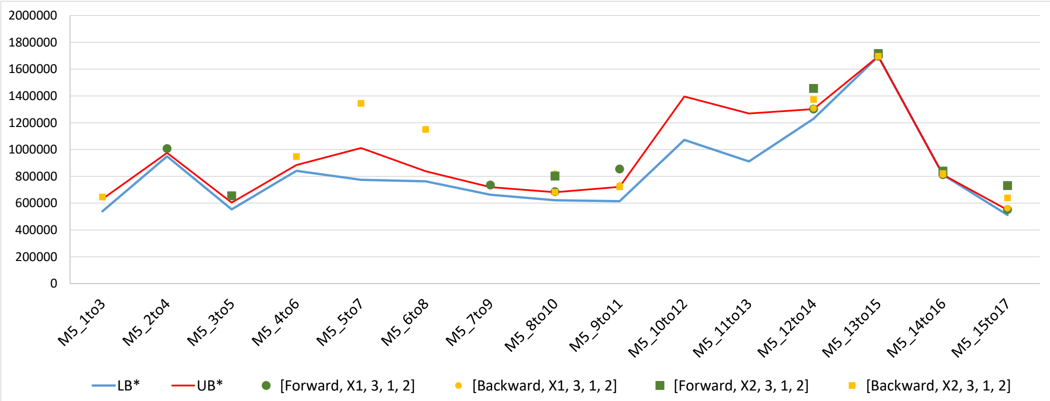
(c)



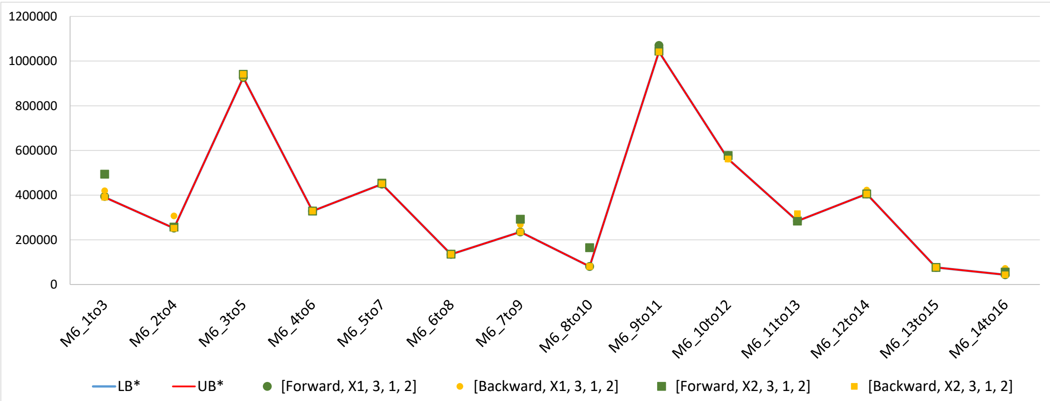
(d)

Source: Own authorship.

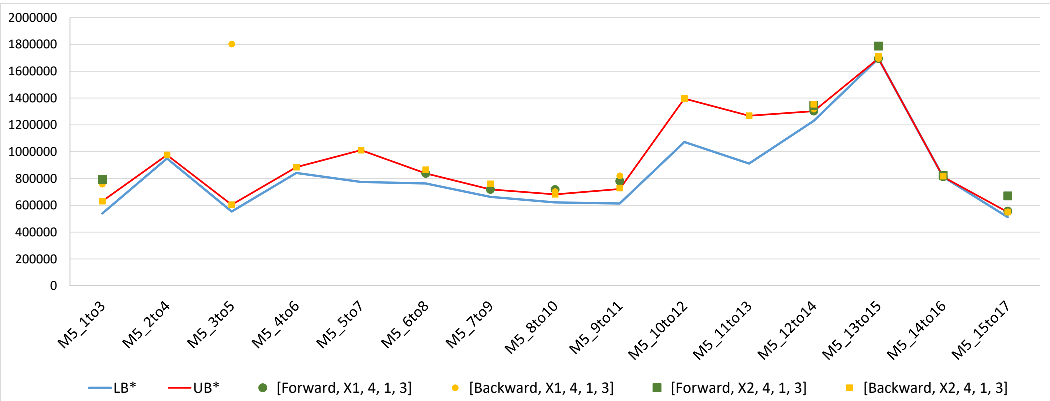
Figure 20 – Comparison of R&F-F&O approaches in relation to upper bound.



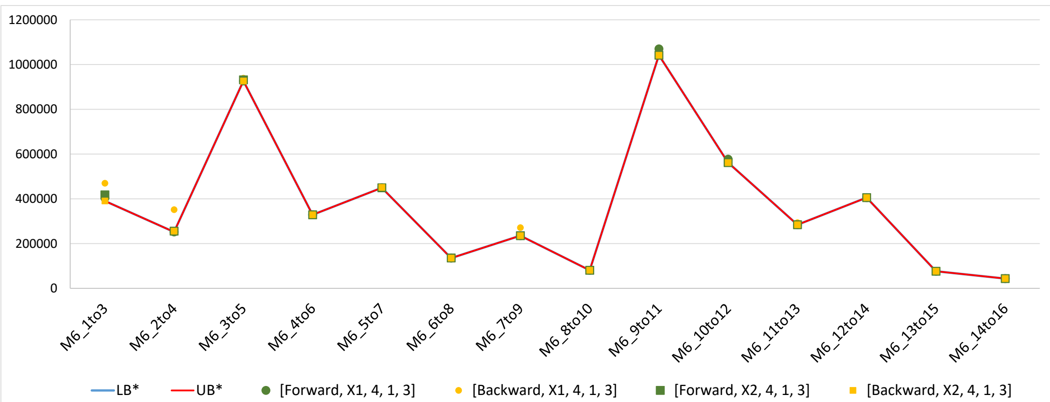
(a)



(b)



(c)



(d)

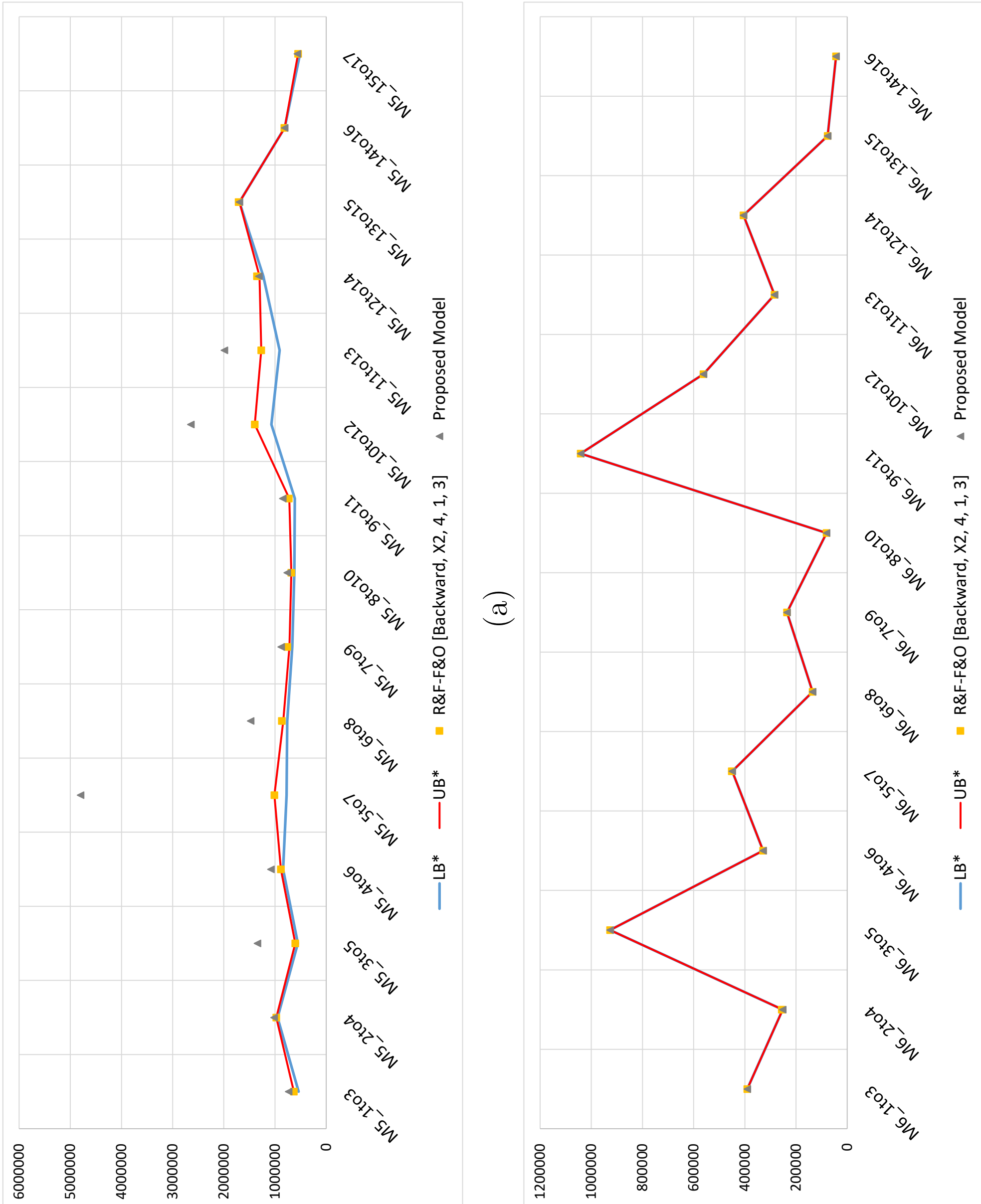
Source: Own authorship.

Table 9 – Computational results obtained with R&F-F&O[$Backward$, $X2$, 4, 1, 3].

Instance	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
	Cf	Cup	$Cout$	$Cover$	$nOut$	$nOver$	nF	$nRest$	OF_{ub}	$Avg\ gap$	$iter$	$CPUt$	OF_{ub}	$Avg\ gap$	$iter$	$CPUt$	$Avg\ gap$	$iter$	$CPUt$	St. Sol.
M5_1to3	465,575.80	15,975.44	0.00	149,196.27	0	746	40	54	1,620,522.41	0.948%	4	1,127.07	630,747.51	0.005%	2	659.74	0.476%	6	1,786.81	Ending
M5_2to4	527,008.22	11,497.76	299,459.18	137,396.56	1	687	46	75	7,101,838.43	32.203%	4	1,478.02	975,361.72	0.007%	2	1,676.08	16.105%	6	3,154.10	Ending
M5_3to5	546,370.41	28,046.46	0.00	29,399.26	0	147	47	69	4,162,992.33	19.148%	4	1,337.54	603,816.13	0.157%	2	2,279.41	9.652%	6	3,616.95	Time Limit
M5_4to6	470,843.61	32,288.61	320,791.98	60,398.49	1	302	47	58	2,071,196.71	6.316%	4	1,476.26	884,322.69	0.005%	2	1,043.31	3.161%	6	2,519.57	Ending
M5_5to7	764,453.57	50,279.52	0.00	196,195.09	0	981	54	60	2,199,559.90	1.406%	4	1,377.24	1,010,928.18	5.804%	2	2,239.70	3.605%	6	3,616.94	Time Limit
M5_6to8	750,585.60	32,070.54	0.00	82,797.93	0	414	58	79	4,166,239.93	1.362%	4	1,064.50	865,454.07	0.008%	2	943.89	0.685%	6	2,008.39	Ending
M5_7to9	631,496.62	24,880.15	0.00	102,997.42	0	515	49	66	2,175,586.23	0.082%	4	1,004.78	759,374.19	0.000%	2	473.44	0.041%	6	1,478.22	Ending
M5_8to10	652,304.08	15,355.93	0.00	14,999.63	0	75	54	56	759,936.99	1.973%	4	1,107.25	682,659.63	0.000%	2	407.77	0.987%	6	1,515.02	Ending
M5_9to11	592,496.42	21,566.96	0.00	115,294.84	0	583.5	53	64	954,579.98	3.988%	4	1,283.77	729,358.22	0.009%	2	1,279.64	1.998%	6	2,563.41	Ending
M5_10to12	864,451.46	15,809.50	354,124.48	161,026.78	1	831	64	70	1,693,500.81	8.699%	4	1,345.16	1,395,412.22	10.015%	2	2,273.98	9.357%	6	3,619.14	Time Limit
M5_11to13	697,589.19	12,420.24	311,992.20	246,324.65	1	1257.5	61	70	1,645,694.89	20.553%	4	1,306.16	1,268,326.28	10.298%	2	2,309.71	15.425%	6	3,615.87	Time Limit
M5_12to14	627,174.89	11,682.74	573,429.60	141,248.73	2	790	55	61	2,661,748.23	0.002%	4	888.49	1,353,535.96	0.628%	2	2,725.15	0.315%	6	3,613.64	Time Limit
M5_13to15	430,628.93	9,935.70	1,259,693.76	9,599.76	4	48	36	39	2,775,032.49	0.000%	4	218.08	1,709,858.15	0.000%	2	124.47	0.000%	6	342.55	Ending
M5_14to16	427,073.97	11,760.56	276,889.68	101,797.45	1	509	36	37	903,814.24	0.000%	4	245.79	817,521.66	0.000%	2	124.41	0.000%	6	370.20	Ending
M5_15to17	408,590.99	25,343.41	0.00	115,447.22	0	581	30	42	649,688.88	1.084%	3	649.94	549,381.62	0.001%	2	157.29	0.542%	5	807.23	Ending
Avg M5	**590,442.92**	**21,260.90**	**226,425.39**	**110,941.34**	**1**	**564**	**49**	**60**	**2,369,462.16**	**6.518%**	**3.93**	**1,060.67**	**949,070.55**	**1.796%**	**2**	**1,247.87**	**4.157%**	**5.93**	**2,308.54**	**-**
M6_1to3	342,596.91	6,532.87	0.00	41,198.97	0	206	31	44	508,925.79	0.000%	4	166.19	390,328.75	0.000%	2	164.68	0.000%	6	330.87	Ending
M6_2to4	242,993.35	10,509.65	0.00	2,199.95	0	11	27	23	324,904.13	0.000%	4	107.99	255,702.95	0.000%	2	85.15	0.000%	6	193.14	Ending
M6_3to5	322,575.51	31,931.25	570,065.40	2,199.95	2	11	28	28	966,168.98	0.002%	4	93.41	926,772.11	0.000%	2	75.64	0.001%	6	169.05	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.70	0	259	23	16	328,431.00	0.000%	4	38.17	328,431.00	0.000%	2	36.58	0.000%	6	74.75	Ending
M6_5to7	204,242.47	7,506.50	227,194.32	11,999.70	1	60	22	18	461,076.07	0.000%	4	36.59	450,942.99	0.000%	2	33.13	0.000%	6	69.72	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	13	137,257.46	0.000%	4	22.06	134,864.25	0.000%	2	19.01	0.000%	6	41.07	Ending
M6_7to9	197,684.64	3,973.06	0.00	33,399.17	0	167	12	15	235,776.86	0.000%	4	16.68	235,056.87	0.000%	2	12.02	0.000%	6	28.70	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	80,989.52	0.000%	4	8.34	80,989.52	0.000%	2	6.18	0.000%	6	14.52	Ending
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,041,799.55	0.000%	4	6.37	1,041,799.55	0.000%	2	4.21	0.000%	6	10.58	Ending
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	11	561,693.60	0.000%	4	7.87	561,693.60	0.000%	2	5.65	0.000%	6	13.52	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.15	1	171	3	9	352,657.26	0.000%	4	6.56	283,953.15	0.000%	2	3.99	0.000%	6	10.55	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	405,466.47	0.000%	4	9.52	405,466.47	0.000%	2	8.63	0.000%	6	18.15	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	8	76,158.33	0.000%	4	9.57	76,158.33	0.000%	2	6.99	0.000%	6	16.56	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	43,598.91	0.000%	4	8.61	43,598.91	0.000%	2	5.30	0.000%	6	13.91	Ending
Avg M6	**150,504.73**	**6,532.27**	**188,305.18**	**27,211.99**	**1**	**142**	**13**	**15**	**394,635.99**	**0.000%**	**4.00**	**38.42**	**372,554.17**	**0.000%**	**2**	**33.37**	**0.000%**	**6.00**	**71.79**	**-**

Source: Own authorship.

Figure 21 – Comparison between best R&F-F&O approach and proposed model in relation to upper bound.



(a)

(b)

Source: Own authorship.

Chapter 4

A branch-and-price algorithm for aircraft routing with crew assignment in the on-demand air transportation

Among the exact approaches applicable to variants of VRP, one that stands out as a successful technique for solving practical and large-scale problems is the branch-and-price (B&P) algorithm (BALDACCI; MINGOZZI; ROBERTI, 2012; COSTA; CONTARDO; DESAULNIERS, 2019; PESSOA et al., 2020). Once the problem discussed in Chapter 3 is analogous to VRP, we were motivated to develop a B&P method to be able to solve realistic-sized instances in acceptable computational times.

In the B&P approach, a branch-and-bound (B&B) tree (LAND; DOIG, 1960) is constructed relying on the column generation (CG) technique (LÜBBECKE; DESROSIERS, 2005) to solve each node of the tree, commonly employed in conjunction with the Danzig-Wolfe decomposition (DWD). Briefly, DWD (DANTZIG; WOLFE, 1960) is a classical reformulation method for solving MIP optimization problems whose constraint matrix involves a set of independent blocks, dispersed in a special structure of primal block-angular and linked by coupling constraints. This block structure suggests a decomposition into new problems with smaller dimensions, referenced as *pricing subproblems*. Coupling constraints together with the property that allows rewriting every point of a non-empty and bounded convex polyhedron, can be represented as a linear convex combination of its extreme points and linear combination of its extreme rays, making it possible to obtain an equivalent problem with smaller number of constraints, called the *master problem*.

Given this decomposition into master problem and subproblems, the CG technique is adopted to effectively solve the linear relaxation of the master problem. It considers only a subset of columns in this linear relaxation, resulting in what is known as the *restricted master problem* (which has a smaller number of variables). Column generation uses the dual information from the restricted master problem in the subproblems to generate new columns (variables) and then adds them to this problem, until the optimal solution of the master problem is reached. In the case of MIP problems, we still have to make sure that the obtained solution is not fractional. Therefore, the B&B algorithm ends up making partitions in the solution space, taking advantage of the dual bounds computed during the enumeration to achieve faster convergence to optimality (i.e. we have an efficient node pruning because B&B avoids unnecessary exploration of solution space areas where the optimal solution cannot be found). Additionally, if there are cutting planes (valid inequalities) to tighten the LP relaxations within the B&B tree, this results in a method named branch-price-and-cut (BP&C). Most B&P algorithms are developed specifically for each type of problem, since there are several specificities regarding the type of pricing subproblems, effective branching rules, auxiliary heuristics, among many other components, which hinder the development of general-purpose implementations.

In this chapter, Section 4.1 introduces the master problem (represented by the well-known set partitioning formulation). In Section 4.2, we reveal the development of a tailored labeling algorithm for solving the subproblems. Section 4.3 formalizes the column generation technique, while Section 4.4 exhibit the branching rules used in B&B tree. Ultimately, Section 4.6 concludes the chapter with computational experiments conducted for our B&P algorithm. For these sections, consider all the notation that has already been presented in Chapter 3. A faster query of these sets, parameters and decision variables, can be done through Appendix A.1.

4.1 Set partitioning formulation

In general, when the DWD is applied to variants of VRP, the resulting master problem (MP) falls into the classic set partitioning formulation, because it is formed by vehicle coupling constraints, in which the columns of coefficient matrix corresponds to independent routing decisions, thus establishing the subproblems. Moreover, the most effective exact methods for VRPs are based on those mathematical models (PECIN et al., 2017; TOTH; VIGO, 2014).

4.1.1 Master Problem

Before explaining the set partitioning formulation related to our DAFP, consider the following notation:

- Λ^v: set of all feasible routes for aircraft $v \in \mathcal{V}$ (where each route corresponds to a sequence of requests served by the aircraft);
- $\alpha^v_{r\rho}$: binary parameter that indicates whether request $r \in \mathcal{L} \cup \mathcal{M}$ is attended or not by route ρ, originating from aircraft v;
- ς^v_ρ: operational cost of route ρ, when executed by aircraft v (which implies the inclusion of positioning, upgrade and overtime costs);
- λ^v_ρ: binary variable that takes value 1, if only if, route $\rho \in \Lambda^v$ is chosen in the solution.

The set partitioning is formulated by:

$$\min \sum_{v\in\mathcal{V}} \sum_{\rho\in\Lambda^v} \varsigma^v_\rho . \lambda^v_\rho + \sum_{r\in\mathcal{L}} Cout_r . out_r; \tag{218}$$

s.t.

$$\sum_{\substack{v\in\mathcal{V}: \\ \check{p}^v \geq \hat{p}^r}} \sum_{\rho\in\Lambda^v} \alpha^v_{r\rho} . \lambda^v_\rho + out_r = 1; \ \forall\ r \in \mathcal{L}; \tag{219}$$

$$\sum_{\rho\in\Lambda^{v^r}} \alpha^{v^r}_{r\rho} . \lambda^{v^r}_\rho = 1; \ \forall\ r \in \mathcal{M}; \tag{220}$$

$$\sum_{\rho\in\Lambda^v} \lambda^v_\rho \leq 1; \ \forall\ v \in \mathcal{V}; \tag{221}$$

$$\lambda^v_\rho \in \{0,1\}; \ \forall\ v \in \mathcal{V}; \rho \in \Lambda^v; \tag{222}$$

$$out_r \geq 0; \ \forall\ r \in \mathcal{L}. \tag{223}$$

Objective function (218) consists of two parts. The first is related to the operational costs obtained through the routes, and the second represents the outlay for outsourcing customer requests. Constraints (219) impose that, if customer request r is not subcontracted ($out_r = 0$), then it is served in exactly one route ρ of an aircraft belonging to the requested type or higher ($\check{p}^v \geq \hat{p}^r$). Each maintenance request r that is complied with a route executed by the specified aircraft v^r is guaranteed by constraints (220). In turn, constraints (221) ensure that at most a single route ρ can be made for each aircraft v (which also limits the number of routes that can be used in the solution by V). Finally, the domain of the decision variables is defined in constraints (222) and (223). Observe that in (223), the variable out_r did not need to be declared as binary because constraints (219) and (222) enforce this domain in a feasible solution.

4.1.2 Restricted master problem

Given the extremely large number of possible routes that we can have in Λ^v, solving this MP requires the use of column generation technique, which is applied to the linear relaxation of the problem. This technique is an iterative method that, at each iteration, solves the linear relaxation defined by only a subset of routes. Thus, restricting each aircraft v to a delimited subset of routes, which means having $\overline{\Lambda}^v \subset \Lambda^v$, in a given iteration of the CG technique we solve the following problem, denominated by restricted master problem (RMP), as already mentioned:

$$\min \sum_{v\in\mathcal{V}} \sum_{\rho\in\overline{\Lambda}^v} \varsigma^v_\rho.\lambda^v_\rho + \sum_{r\in\mathcal{L}} Cout_r.out_r; \tag{224}$$

s.t.

$$\sum_{\substack{v\in\mathcal{V}:\\ \check{p}^v\geq\hat{p}^r}} \sum_{\rho\in\overline{\Lambda}^v} \alpha^v_{r\rho}.\lambda^v_\rho + out_r = 1;\ \forall\ r\in\mathcal{L}; \qquad (\omega_r) \tag{225}$$

$$\sum_{\rho\in\overline{\Lambda}^{v^r}} \alpha^{v^r}_{r\rho}.\lambda^{v^r}_\rho = 1;\ \forall\ r\in\mathcal{M}; \qquad (\varphi_r) \tag{226}$$

$$\sum_{\rho\in\overline{\Lambda}^v} \lambda^v_\rho \leq 1;\ \forall\ v\in\mathcal{V}; \qquad (\varrho_v) \tag{227}$$

$$\lambda^v_\rho \in [0,1];\ \forall\ v\in\mathcal{V}; \rho\in\overline{\Lambda}^v; \tag{228}$$

$$out_r \geq 0;\ \forall\ r\in\mathcal{L}. \tag{229}$$

Looking at the current RMP, we notice the presence of ω_r, φ_r and ϱ_v on the right-hand side of constraints (225)-(227). The first two, $\omega_r \in \mathbb{R}$ and $\varphi_r \in \mathbb{R}$, represent the dual variables of coupling constraints (225) and (226), and $\varrho_v \leq 0$ is the dual of each convexity constraint in (227). Although this type of formulation demands more sophisticated solution methods than its counterpart, which is based on the traditional resolution of the original model through a general-purpose solver, the most common benefits of the column generation technique lie in the possibility of providing reduced memory consumption (coming from the CG itself), and offering considerably tighter bounds than the original problem (due to the imposition of integrality in the subproblems), typically producing more improved results at better computational times (MUNARI; GONDZIO, 2013; PECIN et al., 2017; ALVAREZ; MUNARI, 2017; MUNARI; MORABITO, 2018).

4.2 Labeling algorithm

To generate new columns for the RMP at each iteration of the CG algorithm, it is necessary to solve a subproblem for each aircraft $v \in \mathcal{V}$. The subproblem can be represented by a MIP model, to then be optimized by a solver. In the VRP literature, we often find subproblems belonging to the NP-hard class, which makes it impracticable to call the solver for optimizing a MIP model at each iteration of the CG. In our case, the subproblem refers to a resource-contrained elementary shortest-path problem (RCESPP), which is NP-hard (DROR, 1994). One way to solve this subproblem more efficiently is to apply dynamic programming (DP) methods, such as the labeling algorithm (DESROCHERS,

1986; BEASLEY; CHRISTOFIDES, 1989; IRNICH; DESAULNIERS, 2005; FEILLET, 2010).

In this algorithm's framework, we associate a *bucket* (list) with each node (or request) of the RCESPP's graph. Inside each bucket, there are *labels* (data structures) responsible for characterizing (labeling) by means of attributes/states a partial path (route segment) that comes to the respective designated node. A label can inform, for example, the subset of visited nodes, the accumulated reduced cost, and resource consumption (e.g., time, weight, etc.) from the source/origin node. Bearing in mind the crew rules of our DAFP (Subsection 3.1), we define the following resources for a given label ℓ_s at node s:

- $\overline{\varsigma}(\ell_s)$: total reduced (or relative) cost of the partial path;
- $\Re(\ell_s)$: set of visited vertices along the partial path;
- $\overline{W}(\ell_s)$: exact time at which request s is serviced along the partial path;
- $\overline{U}(\ell_s)$: accumulated work time along the partial path;
- $\overline{Q}(\ell_s)$: accumulated flight time along the partial path;
- $\overline{aGT}(\ell_s)$: accumulated ground time along the partial path;
- $\overline{firstM}(\ell_s)$: binary artifice that assumes 1, if only maintenance is performed along the partial path (starting from the dummy); 0, otherwise. It is therefore used to find out when a crew's workday actually begins;
- $\overline{\mathcal{CO}}(\ell_s)$: overtime cost arising from the arc at which label ℓ_s is related (i.e., it is not an accumulated resource).

In the context of RCESPP, we assign a different index number for the source and sink/destination node. Let $R = |\mathcal{L} \cup \mathcal{M}|$, we declare 0 as the source node and $R+1$ as the sink node, and define $\mathcal{R}^+ = \{0\} \cup \mathcal{L} \cup \mathcal{M} \cup \{R+1\}$ as the set with all nodes.

4.2.1 Duty cases of extensions

At each iteration, a parent label is extended when we create a child (new) label to compose the bucket of a next node (i.e. the partial path is expanded by inserting a new vertex), considering the attributes that were defined by the parent label (from a previous bucket). The extension can be progressive (from the source node to its successors, a.k.a forward propagation), regressive (from the sink node to its predecessors, a.k.a backward propagation), or even bidirectional (the combination of both) (TILK; GOEL, 2020). In our labeling algorithm, we used a progressive extension based on six duty cases consistent with the crew labor rules (they are illustrated in Figure 22):

- *Case 1* – hoard resources (without putting a rest or work break). This case is the option of executing request r and immediately preparing to start request s, without any rest or work break. Therefore, we only seek to calculate the resulting resources from the label to be created for the next bucket. Additionally, we check the opportunity to take advantage of maintenance time in the form of ground time ($\overline{aGT}(\ell_s)$), so that it can be used to add a rest (in duty cases 2 to 5) or to reduce the duty (in *Case 6*).

- *Case 2* – perform a minimum rest when there is no ferry leg. This aims to end the crew's workday by including $POS + minRest + PRE$, which means resetting the resources $\overline{U}(\ell_s)$, $\overline{Q}(\ell_s)$ and $\overline{aGT}(\ell_s)$ belonging to the subsequent bucket s, calculating the overtime cost in the current bucket r at the situation $\overline{U}(\ell_r) > maxDuty$ or $\overline{Q}(\ell_r) > maxFlying$.

- *Case 3* – perform a minimum rest before a ferry leg. This case is similar to *Case 2*, but it considers the occurrence of a positioning flight between request r and s in the duty case. As $POS + minRest + PRE$ is putting before the ferry leg, the resources $\overline{U}(\ell_s)$ and $\overline{Q}(\ell_s)$ must take into account the ferry execution time at the beginning of the next duty period.

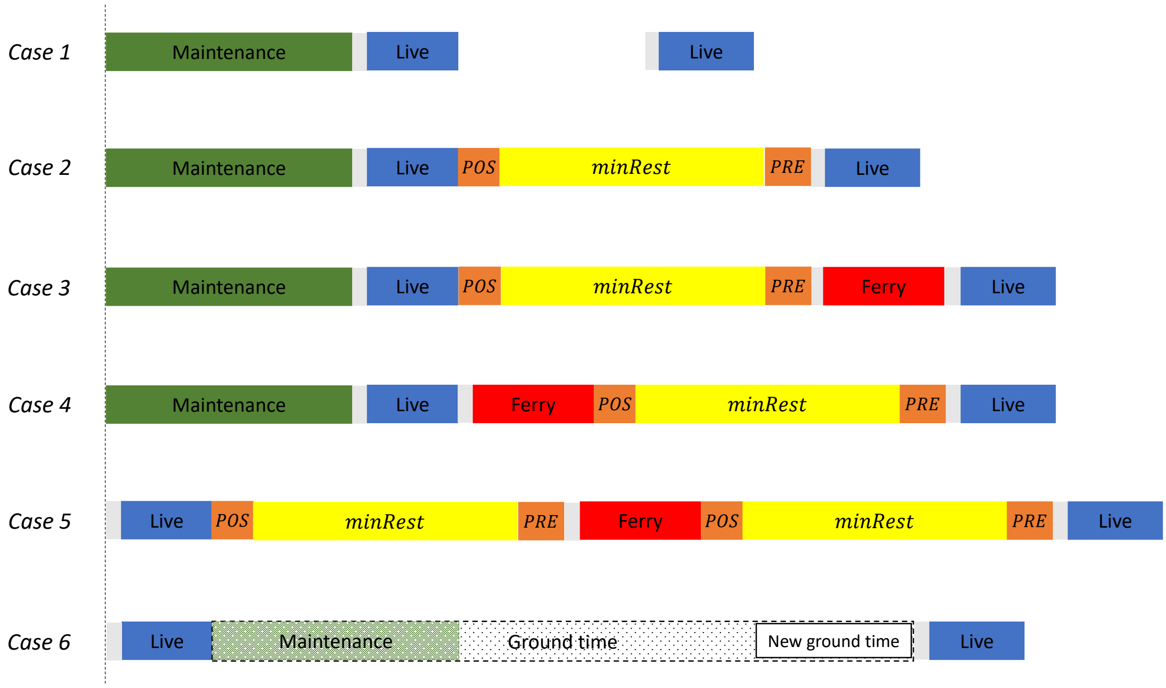
- *Case 4* – perform a minimum rest after a ferry leg. Again, we have a case analogous to cases 2 and 3, but the difference lies in the quantification of current resources, since the ferry time is included in $\overline{U}(\ell_r)$ and $\overline{Q}(\ell_r)$, where the duty period ends.

- *Case 5* – perform a minimum rest before and after a ferry leg. It is a combination of cases 3 and 4, and represents an attempt to have a minimum rest before and after a positioning flight. This case is especially useful on particularly long positioning trips. The benefit of this strategy over cases 3 and 4 is that it does not account for the ferry time on the resources in bucket r, as well as in bucket s. However, the disadvantage is how much $\overline{W}(\ell_s)$ increases, after all, $POS + minRest + PRE$ is included twice.

- *Case 6* – apply the split duty rule, where ground time is reduced in order to take a work break. In this situation, the current ground time (which includes $\overline{aGT}(\ell_r)$) must be between 90 min and 6 hours, or greater than 6 hours (but not exceeding $minRest$), to then allow it to be reduced so that $\overline{U}(\ell_s)$ has a lower value, preventing $\overline{W}(\ell_s)$ from surpassing its time window by placing the rest.

It is important to mention that in all cases, we also need to check whether the flight events (live and ferry legs) are covered by any pilot's time window. If they are not within a window, we may try to bring the given flight event to the nearest adjacent window, checking if $\overline{W}(\ell_s)$ is still into its limits.

Figure 22 – The six cases of extending a label.



Source: Own authorship.

As overtimes are indirectly related to the position of $\overline{W}(\ell_s)$ in the time horizon, and they are included in the optimization criteria, in addition to considering its earlier start in the recursive expressions (common in the literature), we must also take into account its later start.

Delaying $\overline{W}(\ell_s)$ on purpose makes it possible to abate the length of a duty period. In *Case 6*, if the accumulated ground time exceeds 6 hours, it is then reduced to just one hour, a much greater diminution than that provided by the previous interval, when the ground time is between 90 min and 6 hours (see Table 3 in Chapter 3). Another advantage is when we have a minimum rest before a maintenance request (cases 2, 4 and 5). The maintenance postponement in this situation makes the beginning of a next duty to be late, which reduces its length. As a consequence, we have created two variants for each case, resulting in a total of 12 variants (of the six cases). Let $1a$ and $1b$ be the variants of *Case 1*, representing the earliest and latest start, respectively, at which a request can be serviced. In this way, we generalize $\mathbb{V} = \{1a, 1b, 2a, 2b, \ldots, 6a, 6b\}$ as the set of variants from the six cases, and define $\overline{var}(\ell_s)$ as the case variant associated with label ℓ_s (thus, if $\overline{var}(\ell_s) = 2b$, we know that label ℓ_s has assigned to *Case 2*, with $\overline{W}(\ell_s)$ starting later). Accordingly, when a parent label generates a child label in another bucket, the extension always considers one of the 12 variant cases.

In general terms, the progressive extension of a label ℓ_r to a bucket s (which creates a label ℓ_s), taking as a reference one of the variants (var) of *Case 1*, for example, is done as follows. Let tw be the earliest or latest start time at which request s can be executed, computed by:

$$tw = \begin{cases} st_s, & var = 1a; \\ st_s + \Delta_{\mathcal{L}}, & var = 1b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 1b \wedge s \in \mathcal{M}; \end{cases}$$

for distinct requests $r, s \in \mathcal{L} \cup \mathcal{M}$ from an aircraft $v \in \mathcal{V}$, $\overline{W}(\ell_s)$ is determined as:

$$\overline{W}(\ell_s) = \begin{cases} \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TL_r + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

Given the time imposition arising from the arc (r, s), $\overline{W}(\ell_s)$ is checked to assess if it continues within the time window required by request s. To do this, we use the auxiliary binary variable $checkTW$, expressed by:

$$checkTW = \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

If $checkTW = 1$, we proceed to the next step, which consists of verifying whether all flight events (live or ferry legs) in the arc (r, s) are covered by any pilot's time window. If the validation is false, we still try to bring the respective flight event to the nearest adjacent window, which implies delaying $\overline{W}(\ell_s)$ even more. If $checkTW$ remains true, then we advance to calculate the remaining resources $\overline{aGT}(\ell_s)$, $\overline{U}(\ell_s)$, $\overline{Q}(\ell_s)$, $\overline{firstM}(\ell_s)$ and $\overline{\mathcal{CO}}(\ell_s)$.

The resource $\overline{aGT}(\ell_s)$ is determined by removing the flight times:

$$\overline{aGT}(\ell_s) = \begin{cases} [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TL_r + TF^{\check{p}^v}_{j^r i^s}), & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \end{cases}$$

notice that among maintenance requests without a ferry occurrence, the accumulated ground time from previous arcs, $\overline{aGT}(\ell_r)$, is also counted in $\overline{aGT}(\ell_s)$, so that we can use the duration of these requests as a rest or break in the next bucket (for the other duty cases). If $\overline{aGT}(\ell_r)$ is not availed, then it is used in the resource $\overline{U}(\ell_s)$, which defines this time as work within the duty period (see the situations that request $r \in \mathcal{M}$):

$$\overline{U}(\ell_s) = \begin{cases} \overline{U}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r), & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases} \tag{230}$$

In an arc (r, s), the resource $\overline{Q}(\ell_s)$ just accumulates the flight times, keeping the same value in the situation where we only have maintenance request:

$$\overline{Q}(\ell_s) = \begin{cases} \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ \overline{Q}(\ell_r), & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

As was done in the proposed model of Chapter 3, we use a mathematical artifice, $\overline{firstM}(\ell_s)$, to find out when the duty period starts, i.e., the moment at which the first flight event appears, indicated by $\overline{firstM}(\ell_s) = 0$. The resource $\overline{firstM}(\ell_s)$ is calculated by:

$$\overline{firstM}(\ell_s) = \begin{cases} 1, & \text{if } r = 0 \wedge s \in \mathcal{M} \wedge k^v = i^s; \\ \overline{firstM}(\ell_r), & \text{else if } r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ 0, & \text{otherwise.} \end{cases}$$

When $\overline{firstM}(\ell_s) = 1$, the resource $\overline{U}(\ell_s)$ has the value corresponding to the beginning of a duty period (note that the recursive expression (230) was defined for $\overline{firstM}(\ell_s) = 0$):

$$\overline{U}(\ell_s) = \begin{cases} PRE + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s \wedge \overline{firstM}(\ell_r) = 1; \\ PRE + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 1; \\ PRE + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 1. \end{cases}$$

Finally, to calculate the resource $\overline{\mathcal{CO}}(\ell_s)$, we first need to quantify the overtime ($\overline{\mathcal{O}}$), given by the expression:

$$\overline{\mathcal{O}} = \max\{0, \overline{U}(\ell_s) - maxDuty, \overline{Q}(\ell_s) - maxFlying\}$$

.

As the overtime costs are penalized in the objective function, the only circumstances in which these costs worthwhile are: the first moment that the crew exceeds $maxDuty$ or $maxFlying$ (computed by *Cases 2* to *5*) in the current duty period, and at the end of the last duty performed by the aircraft. In *Case 1*, the overtime cost is formulated by:

$$\overline{\mathcal{CO}}(\ell_s) = Cover_r.\overline{\mathcal{O}};\ \forall\ r \in \mathcal{L} \cup \mathcal{M}, s = R+1;$$

where, $Cover_r$ is the overtime cost per minute for request r.

When comparing *Case 1* in relation to the cases that refer to a rest inclusion (*Cases 2* to *5*), basically what changes in the resource $\overline{W}(\ell_s)$ is the addition of a full minimum rest $(POS + minRest + PRE)$ immediately after a flight event, and a rest that can be partial or even null, defined by $RestM = \max\{0, (POS + minRest + PRE) - (\overline{aGT}(\ell_r) + TL_r)\}$, immediately after a maintenance request.

Regarding the resource $\overline{U}(\ell_s)$, in *Cases 2*, *4* and *5*, where there is no ferry leg at the beginning of a duty period, the reset of this resource is expressed by $\overline{U}(\ell_s) = PRE + tat^s_{i^s}$. In *Case 3*, where the ferry leg is considered at the beginning of a duty period, the reset is:

$$\overline{U}(\ell_s) = \begin{cases} PRE + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s; \\ PRE + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

For the other resources, the reset is also the same in *Cases 2*, *4* and *5*: $\overline{aGT}(\ell_s) = 0$, $\overline{Q}(\ell_s) = 0$ and $\overline{firstM}(\ell_s) = 0$. In *Case 3*, the only resource that changes is $\overline{Q}(\ell_s) = TF^{\check{p}^v}_{j^r i^s}$, remaining null $\overline{aGT}(\ell_s)$ and $\overline{firstM}(\ell_s)$. Concerning the accumulated work/flight time that closes a duty period, it is counted in the overtime $\overline{\mathcal{O}}$, which may include a ferry time, according to *Cases 4* and *5*.

Lastly, *Case 6* only happens when the current ground time (which considers $\overline{aGT}(\ell_r)$), defined by GT, is within the time ranges in which we have duty reduction. Given this case, GT is determined as:

$$GT = \begin{cases} [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] + \overline{aGT}(\ell_r) - tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] + \overline{aGT}(\ell_r) - (TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}), & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

As stated in the split duty rule, the new ground time, denoted by $newGT$, is calculated as:

$$newGT = \begin{cases} (GT - 90)/2 + 90, & 90 < GT \leq 360; \\ 60, & 360 < GT \leq minRest. \end{cases}$$

Therefore, *newGT* is considered in the formulation of $\overline{U}(\ell_s)$ as:

$$\overline{U}(\ell_s) = \begin{cases} \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + newGT + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + newGT + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + newGT + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + newGT + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

Note that in *Case 6*, $\overline{U}(\ell_s)$ has a smaller value compared to the same resource determined in expression (230) (from *Case 1*).

All the six duty cases are represented in more detail by algorithms 8-13 (named *CheckDutyCase1*, ..., *CheckDutyCase6*), available in Appendix B.1.

4.2.2 Dominance rules

The efficiency of a labeling algorithm is improved by the presence of dominance rules, as they can drastically reduce the proliferation of labels during the algorithm execution, avoiding exhaustive enumeration. These rules are based on eliminating labels that cannot lead to an optimal solution. We propose specific dominance rules tailored to the addressed problem, defined as follows. Given two labels ℓ_r and ℓ'_r associated with the same bucket of request $r \in \mathcal{L} \cup \mathcal{M}$, ℓ_r does not dominate ℓ'_r if at least one of the following conditions is satisfied:

$$\begin{aligned} &\overline{\varsigma}(\ell_r) > \overline{\varsigma}(\ell'_r); \\ &\Re(\ell_r) \supset \Re(\ell'_r); \\ &\overline{W}(\ell_r) > \overline{W}(\ell'_r); \\ &\overline{U}(\ell_r) > \overline{U}(\ell'_r); \\ &\overline{Q}(\ell_r) > \overline{Q}(\ell'_r); \\ &\left(\overline{var}(\ell_r) = 1a \wedge \overline{var}(\ell'_r) = 1b\right); \\ &\left(\overline{var}(\ell_r) = 2a \wedge \overline{var}(\ell'_r) = 2b\right); \\ &\left(\overline{var}(\ell_r) = 3a \wedge \overline{var}(\ell'_r) = 3b\right); \\ &\left(\overline{var}(\ell_r) = 4a \wedge \overline{var}(\ell'_r) = 4b\right); \\ &\left(\overline{var}(\ell_r) = 5a \wedge \overline{var}(\ell'_r) = 5b\right); \\ &\left(\overline{var}(\ell_r) = 6a \wedge \overline{var}(\ell'_r) = 6b\right). \end{aligned}$$

The first line (inequality) refers to the reduced cost, computed for label ℓ_s as $\overline{\varsigma}(\ell_s) = \overline{\varsigma}(\ell_r) + Cf^v_{rs} + Cup^v_s + \overline{\mathcal{CO}}(\ell_s) - (\omega_s + \varphi_s + \varrho_v)$. We must also guarantee that there is no dominance relationship among two variants of the same case (see the last six lines), since variant a tends to have better attributes than b when both are compared to the same bucket, however, b can benefit the reduced cost of a subsequent label, as already discussed.

4.2.3 Unreachable labels

Another way to reduce the number of extended labels is to identify those so-called unreachable (FEILLET, 2010), which in our scope are labels outside the following condition (assuming that an aircraft v performs a route in which request r immediately precedes request s):

$$\begin{aligned}
\exists(v,r,s) := &(r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge \check{p}^v \geq \hat{p}^r \wedge \hat{p}^s \geq \hat{p}^r \wedge st_r \leq st_s + \Delta_{\mathcal{L}}) \vee \\
&(r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge \check{p}^v \geq \hat{p}^r \wedge v = v^s \wedge st_r \leq st_s + \Delta_{\mathcal{M}}) \vee \\
&(r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge v = v^r \wedge \hat{p}^r \geq \hat{p}^s \wedge st_r \leq st_s + \Delta_{\mathcal{L}}) \vee \\
&(r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge v = v^r \wedge v^r = v^s \wedge st_r \leq st_s + \Delta_{\mathcal{M}}) \vee \\
&\left(r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge st_r + TF^{\hat{p}^r}_{i^r j^r} < st_s + \Delta_{\mathcal{M}}\right) \vee \\
&(r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge st_r - \Delta_{\mathcal{M}} + TL_r < st_s + \Delta_{\mathcal{L}}) \vee \\
&\left(r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge st_r + TF^{\hat{p}^r}_{i^r j^r} < st_s + \Delta_{\mathcal{L}}\right) \vee \\
&(r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge st_r - \Delta_{\mathcal{M}} + TL_r < st_s + \Delta_{\mathcal{M}}) \vee \\
&(r = 0 \wedge s < R+1) \vee \\
&(r > 0 \wedge s = R+1)\,; \\
&\forall\ v \in \mathcal{V}; r, s \in \mathcal{R}^+ \mid r \neq s.
\end{aligned}$$

In this condition, the first four lines permit only valid allocations between aircraft and requests, i.e., those that satisfy upgrade (for live requests) and pre-assignment (for maintenance requests) requirements of the problem. They also consider the request time windows. The next four lines follow the same logic stipulated for fixing variables in the proposed model (Subsection 3.2.2.7), leaving out situations where a reversal of precedence among requests is impossible. Finally, the last two lines include the dummy nodes (0 and $R+1$).

Based on this condition, we define A_v as the set of feasible (reachable) arcs for aircraft v, and $\hat{\delta}_v$ as set of requests compatible with aircraft v.

4.2.4 Formalizing the algorithm

Algorithm 3 portrays the labeling method as described so far, responsible for calling algorithms 8-13 (Appendix B.1). The algorithm 3 is initialized by creating buckets ($\mathcal{B}(r)$) in the nodes/requests that are compatible with an aircraft v, and the label ℓ_0 (with its attributes starting with initial values) associated with the dummy node 0. To design the algorithm, we defined $\overline{\mathcal{B}}(r) \subseteq \mathcal{B}(r)$, as the set of labels that have not yet been extended, and E, as the set of requests to evaluate. Hence, the stopping criterion is reached when we have no more labels to be processed ($E = \emptyset$). According to the loop shown in steps 5-36, at each iteration, an unprocessed label ℓ_r is extended to a reachable bucket $\mathcal{B}(s)$, taking into account a case variant *var*, if it satisfies all requirements of the problem, evaluated by the algorithm that represents the corresponding duty case (i.e., $Extension = 1$). For labels belonging to the same bucket, the dominance rule is applied to eliminate those that are unnecessary. The procedure terminates by returning the best path, which has the lowest reduced cost.

4.3 Primal-dual column generation method

In the traditional CG technique, the RMP is solved optimally at each iteration, typically by a simplex-type method, so that subproblems receive extreme points from the set of optimal dual solutions to then generate new columns. As a result, classic drawbacks are observed during the CG's execution (VANDERBECK, 2005): slow convergence near the optimal solution (*tailing-off effect*); production of irrelevant columns and dual values (*heading-in effect*); remaining at the same optimal value of RMP for several iterations (*plateau effect*); instability in the dual solutions that jump from one extreme value to another (*yo-yo effect*).

Several alternatives to overcome such weaknesses have been proposed in the literature. Some of them modify the RMP by adding penalties and/or constraints to limit large variations in the dual solutions (MARSTEN; HOGAN; BLANKENSHIP, 1975; AMOR; DESROSIERS; FRANGIONI, 2009; BRIANT et al., 2008). Other alternatives propose smoothing techniques, which use convex combinations of central and optimal dual solutions (WENTGES, 1997; PESSOA et al., 2013). In addition, there exist approaches that work by keeping the dual solutions centered in the feasible set, instead of depending on an optimal point (GOFFIN; VIAL, 2002; ROUSSEAU; GENDREAU; FEILLET, 2007). Interior-point-based methods also allow obtaining well-centered and sub-optimal dual solutions of the RMP, providing more stable columns and valid inequalities, which significantly improves the CG's performance by reducing the number of iterations and computation time (GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2013; MUNARI; GONDZIO, 2013; GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2016).

In this dissertation, we rely on the interior-point CG algorithms presented by Gondzio, González-Brevis and Munari (2013), and, Gondzio, González-Brevis and Munari (2016), known as the primal-dual column generation method (PDCGM) (GONDZIO, 2012). Given a feasible primal-dual solution $(\overline{\lambda}, \overline{\varpi}, \overline{\varphi}, \overline{\varrho})$ of the RMP, which may not be optimal, we can obtain both a lower and upper bound of the RMP's optimal solution by using the primal and dual values from the objective function as follows:

$$Z^{UB}(\overline{\lambda}) = \sum_{v \in \mathcal{V}} \sum_{\rho \in \overline{\Lambda}^v} \varsigma^v_\rho . \overline{\lambda}^v_\rho + \sum_{r \in \mathcal{L}} Cout_r . out_r; \tag{231}$$

$$Z^{LB}(\overline{\varpi}, \overline{\varphi}, \overline{\varrho}) = \sum_{r \in \mathcal{L}} \overline{\varpi}_r + \sum_{r \in \mathcal{M}} \overline{\varphi}_r + \sum_{v \in \mathcal{V}} \overline{\varrho}_v . \tag{232}$$

Algorithm 3: *Forward labeling algorithm*

Input: problem instance, aircraft v.
// Initialization
Create buckets $\mathcal{B}(r) \leftarrow \emptyset, \forall\ r \in \hat{\delta}_v$;
Create label ℓ_0 in bucket $\mathcal{B}(0)$ with initial attribute values ($\ell_0 \leftarrow (\overline{var}(\ell_0) \leftarrow$ "", $\Re(\ell_0) \leftarrow \emptyset$, $\overline{\varsigma}(\ell_0) \leftarrow 0, [\overline{W}(\ell_0) \leftarrow 0, \overline{U}(\ell_0) \leftarrow 0, \overline{Q}(\ell_0) \leftarrow 0, \overline{aGT}(\ell_0) \leftarrow 0, \overline{firstM}(\ell_0) \leftarrow 0, \overline{\mathcal{CO}}(\ell_0) \leftarrow 0]))$;
For each node $r \in \hat{\delta}_v$, make a copy $\overline{\mathcal{B}}(r)$ of bucket $\mathcal{B}(r)$, denoting a set of labels that have not yet been extended;
Let E be a list of active nodes, i.e., at which there are unprocessed labels, initializing $E \leftarrow \{0\}$;
// Search
repeat
  select $r \in E$;
  // Extension
  foreach *label* $\ell_r \in \overline{\mathcal{B}}(r)$ **do**
    foreach $s \in \hat{\delta}_v$, if $(r, s) \in A_v \wedge s \notin \Re(\ell_r)$ **do**
      foreach $var \in \mathbb{V}$ **do**
        Create label ℓ_s with initial attribute values;
        switch *var* **do**
          case $1a$: **case** $1b$ **do**
            Extension $\leftarrow$ *CheckDutyCase1*(var, ℓ_r, ℓ_s);
          case $2a$: **case** $2b$ **do**
            Extension $\leftarrow$ *CheckDutyCase2*(var, ℓ_r, ℓ_s);
          case $3a$: **case** $3b$ **do**
            Extension $\leftarrow$ *CheckDutyCase3*(var, ℓ_r, ℓ_s);
          case $4a$: **case** $4b$ **do**
            Extension $\leftarrow$ *CheckDutyCase4*(var, ℓ_r, ℓ_s);
          case $5a$: **case** $5b$ **do**
            Extension $\leftarrow$ *CheckDutyCase5*(var, ℓ_r, ℓ_s);
          case $6a$: **case** $6b$ **do**
            Extension $\leftarrow$ *CheckDutyCase6*(var, ℓ_r, ℓ_s);
        if *Extension* $= 1$ **then** // verify the feasibility condition
          // Compute the remaining attributes
          $\overline{\varsigma}(\ell_s) \leftarrow \overline{\varsigma}(\ell_r) + Cf^v_{rs} + Cup^v_s + \overline{\mathcal{CO}}(\ell_s) - (\omega_s + \varphi_s + \varrho_v)$;
          $\Re(\ell_s) \leftarrow \Re(\ell_r) \cup \{s\}$;
          // Check dominance
          if ℓ_s *dominates any label* $\ell_s' \in \mathcal{B}(s)$ **then**
            Discard the dominated labels from inside bucket $\mathcal{B}(s)$ (and from $\overline{\mathcal{B}}(s)$, if there is);
          if ℓ_s *is not dominated by any label* $\ell_s' \in \mathcal{B}(s)$ **then**
            Extend label ℓ_r (assign label ℓ_s) to bucket $\mathcal{B}(s)$: $\mathcal{B}(s) \leftarrow \mathcal{B}(s) \cup \{\ell_s\}$;
            $\overline{\mathcal{B}}(s) \leftarrow \overline{\mathcal{B}}(s) \cup \{\ell_s\}$; // the new label also is left as non-extended
      if $(\overline{\mathcal{B}}(s) \neq \emptyset) \wedge (s \neq R + 1)$ **then**
        $E \leftarrow E \cup \{s\}$;
  $\overline{\mathcal{B}}(r) \leftarrow \emptyset$;
  $E \leftarrow E \backslash \{r\}$;
until $E = \emptyset$;
return *best path in* $\mathcal{B}(R + 1)$;

The solution $(\overline{\lambda}, \overline{\omega}, \overline{\varphi}, \overline{\varrho})$ is called suboptimal or ϵ-optimal solution, if it satisfies:

$$0 \leq Z^{UB}(\overline{\lambda}) - Z^{LB}(\overline{\omega}, \overline{\varphi}, \overline{\varrho}) \leq \epsilon. \left(10^{-10} + \left|Z^{UB}(\overline{\lambda})\right|\right); \tag{233}$$

for some tolerance $\epsilon > 0$. Notice that the PDCGM gives well-centered dual solutions because the complementary products are kept close to the centroid's central path until the convergence of optimal solutions. More explicitly, a point is well-centralized if it satisfies:

$$\gamma.\mu \leq \left(\varsigma_\rho^v - \sum_{\substack{r \in \mathcal{L}: \\ \check{p}^v \geq \check{p}^r}} \overline{\omega}_r.\alpha_{r\rho}^v - \sum_{\substack{r \in \mathcal{M}: \\ v = v^r}} \overline{\varphi}_r.\alpha_{r\rho}^v - \overline{\varrho}_v \right).\overline{\lambda}_\rho^v \leq \frac{\mu}{\gamma};\ \forall\ v \in \mathcal{V}, \rho \in \overline{\Lambda}^v; \tag{234}$$

where $\gamma \in (0, 1)$ and μ is a barrier parameter that defines the central path in the PDCGM. Therefore, in virtue of this centrality, the resulting dual solutions oscillate less from one iteration to another, without requiring any artificial resources such as variable bounds or penalty costs. The PDCGM dynamically adjusts the tolerance for solving each RMP by initially setting a loose value and tightening it as CG progresses to optimality.

Regarding the practice of adding columns in the RMP, we follow the standard. Let $\overline{y}_\rho^v = \{\overline{y}_{rs}^{\rho v}\}_{r,s \in \mathcal{R}}$ be an incidence binary vector, such that, $\overline{y}_{rs}^{\rho v} = 1$ if and only if route $\rho \in \overline{\Lambda}^v$ traverses request r, and proceeds directly to request s. After the end of our labeling algorithm (Section 4.2), when $Z_v^{sp} < 0$ (the minimum reduced cost value found for a given subproblem v), we have an extreme optimal point $\overline{y}_\rho^v$ that is not in the current RMP. Thereby, there is a variable λ_ρ^v with negative reduced cost to be included in the RMP $(\{\rho\} \cup \overline{\Lambda}^v)$, resulting in the addition of column $\begin{bmatrix} \varsigma^v.\overline{y}_\rho^v \\ A^v.\overline{y}_\rho^v \\ \varepsilon^v \end{bmatrix}$, in which, $\varsigma^v.\overline{y}_\rho^v$ is the objective function part; $A^v.\overline{y}_\rho^v$, the coupling constraint; and ε^v, the insertion of value 1 in the v-th convexity constraint. Otherwise, subproblem v has an optimal solution, but as the reduced cost of λ_ρ^v is positive, the optimal extreme point $\overline{y}_\rho^v$ is already included in the current RMP. Hence, the column generation technique ends when there are no more columns with a negative relative cost to be inserted.

Finally, algorithm 4 describes in a simplified way our PDCGM version. We can observe that the adjustment of ϵ occurs on the last line of this algorithm, being ϵ^{max} responsible for imposing an upper bound on the ϵ's value. $D > 1$ is the optimality degree that controls the reduction of ϵ as a function of the relative gap, and, δ establishes a termination condition based on the relative gap.

4.4 Branching strategies

As mentioned earlier, in order to obtain a B&P procedure (i.e., guarantee the integralization of all fractional variables that are imposed as binary in the original problem), we

Algorithm 4: *Primal-dual column generation method*

Input: Valid initial columns (feasibility); parameters: $\epsilon^{max} > 0$, $D > 1$, $\delta > 0$.
Set $LB \leftarrow -\infty$, $UB \leftarrow \infty$, $gap \leftarrow \infty$, $\epsilon \leftarrow 0.5$;
while $gap \geq \delta$ **do**
  $\rho \leftarrow \rho + 1$;
  Find a well-centered ϵ-optimal solution $(\overline{\omega}, \overline{\varphi}, \overline{\varrho})$ of the RMP, for iteration ρ;
  $UB \leftarrow \min\{UB, Z^{UB}(\overline{\lambda})\}$;
  foreach $v \in \mathcal{V}$ **do**
    Solve the subproblem v using the labeling algorithm;
    if $Z_v^{sp} < 0$ **then**
      Add column to RMP: $\overline{\Lambda}^v \leftarrow \overline{\Lambda}^v \cup \{\rho\}$;
  $LB \leftarrow \max\{LB, Z^{LB}(\overline{\omega}, \overline{\varphi}, \overline{\varrho}) + \sum_{var \in \mathbb{V}} Z_v^{sp}\}$;
  $gap \leftarrow (UB - LB)/(10^{-10} + |UB|)$;
  Update the tolerance: $\epsilon \leftarrow \min\{\epsilon^{max}, gap/D\}$;

need to embed the CG within each node of a B&B tree. In our implementation, we develop two well-established branching strategies in the VRP literature, known as *two-step branching* (Subsection 4.4.1) and *strong branching* (Subsection 4.4.2). For each one, the search tree is explored using the best-first rule, that is, a node with the smallest lower bound is the node to be processed next.

4.4.1 Two-step branching

This strategy follows the branching scheme used by Desaulniers, Lessard and Hadjar (2008), which is done in two steps, described as follows. Firstly, we attempt to branch on the number of aircraft (or routes) used at a given solution $\overline{\lambda}$ of the RMP, calculated as:

$$nV = \sum_{v \in \mathcal{V}} \sum_{\rho \in \overline{\Lambda}^v} \overline{\lambda}_\rho^v. \tag{235}$$

Thus, if the value of nV is fractional, then the branch is made by creating two child nodes, forcing $\sum_{v \in \mathcal{V}} \sum_{\rho \in \Lambda^v} \lambda_\rho^v \leq \lfloor nV \rfloor$ for the first child, and $\sum_{v \in \mathcal{V}} \sum_{\rho \in \Lambda^v} \lambda_\rho^v \geq \lceil nV \rceil$ for the second one.

If nV is an integer, the branching strategy goes to the second step, which is directed to the arc flows:

$$\overline{y}_{rs}^v = \sum_{\rho \in \overline{\Lambda}^v} \overline{y}_{rs}^{\rho v} . \overline{\lambda}_\rho^v; \ \forall \ v \in \mathcal{V}; \ r, s \in \mathcal{R}; \tag{236}$$

where $\overline{y}_\rho^v$ is the same vector as the one defined at the end of Section 4.3. When selecting a fractional component $\overline{y}_{rs}^v$, the procedure generates two new child nodes, enforcing $\sum_{v \in \mathcal{V}} y_{rs}^v = 0$ for one, and $\sum_{v \in \mathcal{V}} y_{rs}^v = 1$ for the other.

Notice that the branching rule in the first step only modifies the RMP of the child nodes, while the rule in the second step changes both subproblems and the RMP of the child nodes. In the circumstance of $\sum_{v \in \mathcal{V}} y_{rs}^v = 0$, the removal of arc (r, s) must be performed in all subproblems, and it is necessary to delete all columns associated with the routes that pass among requests r and s. For $\sum_{v \in \mathcal{V}} y_{rs}^{v} = 1$, all arcs (r, s) are eliminated for the subproblems given by $\Lambda^{v'}$ such that $v' \neq v$, whereas, in the subproblem defined by Λ^{v}, all arcs starting from node r are discarded, with except the one that goes directly to node s. The associated RMP is also modified in a similar way.

4.4.2 Strong branching

As a second option, we implement a branching rule capable of producing significantly smaller B&B trees, compared to the previous strategy, called strong branching (APPLEGATE et al., 1995; LINDEROTH; SAVELSBERGH, 1999; ACHTERBERG; KOCH; MARTIN, 2005). The main idea is: given a node to be branched on, the procedure assesses which of the candidate fractional variables offers the best improvement to the objective function (i.e., the child nodes are solved for every choice of a set of different candidates) before actually branching on the promising variable. This assessment should be relatively quick in identifying good branches, for the reason that in the worst case, it would be extremely expensive to solve $2K$ RMPs (where K is the number of fractional variables) for making just one branching decision (DEY et al., 2023). To avoid the prohibitive computational time, we just consider a subset of candidate branching variables, and solve the RMPs by running only one iteration of the CG algorithm, without generating any new columns. We are therefore proposing an *approximate strong branching*, since it computes pseudo-costs. For simplicity, our method is henceforth referred to only as strong branching.

There are different ways of developing a strong branching strategy in B&P methods, especially because branching manners can be very specific to the problem (KLABJAN et al., 2001; FUKASAWA et al., 2006; SANTOS et al., 2015). For the B&P method proposed here, when nV is integer, we obtain a corresponding branching from a given fractional solution $\tilde{\lambda}$ of the RMP, in terms of the arc variables $\tilde{y}_{rs}^{v}$ as in (236). Then, for each pair of indices $r, s \in \mathcal{R}$ such that $\sum_{v \in \mathcal{V}} \tilde{y}_{rs}^{v}$ is fractional, we impose independently $\sum_{v \in \mathcal{V}} y_{rs}^{v} = 0$ and $\sum_{v \in \mathcal{V}} y_{rs}^{v} = 1$ on the child nodes, and re-optimize the resulting RMP. Let z_{rs}^{0} and z_{rs}^{1} be the solution values of the RMP after placing $\sum_{v \in \mathcal{V}} y_{rs}^{v} = 0$ and $\sum_{v \in \mathcal{V}} y_{rs}^{v} = 1$, respectively, for the pair $r, s \in \mathcal{R}$. If a resulting RMP becomes infeasible, we define its solution value (z_{rs}^{0} or z_{rs}^{1}) as the best upper bound found in the B&B tree, branching the pair of indices that results in the highest value $z_{rs}^{0} + z_{rs}^{1}$. Since we do not generate new columns, z_{rs}^{0} and z_{rs}^{1} can be quickly computed by re-optimizing the RMP with a subset of the master variables fixed at zero.

4.5 Primal heuristic for the RMP

To improve the upper bound of our B&P method, we incorporate a primal heuristic based on the RMP to try to obtain good feasible/integer solutions prematurely. As any column in the RMP corresponds to a feasible route of the original problem, we can seek a complete and feasible solution by combining a subset of these routes in such a way that all requests are served exactly once. Therefore, a simple MIP-based primal heuristic can be used to impose integrality constraints on variables λ_ρ^v of the RMP, and solving the respective model with a general-purpose MIP solver. This idea has been successfully employed in several VRP variants (SUBRAMANIAN; UCHOA; OCHI, 2013; ARCHETTI; SPERANZA, 2014; ALVAREZ; MUNARI, 2017).

To ensure a fast MIP-based heuristic, we may limit the total runtime of the solver. Knowing that all relevant columns may not be available at the RMP and the stipulated time limit may be reached during execution, we may have as return a solution that is not optimal for the original problem or even no feasible solution at all. Nonetheless, computational results presented in the VRP literature have shown that this type of heuristic commonly brings feasible solutions and upper bounds that are useful for improving the general performance of B&P methods.

4.6 Computational experiments

In this section, we present the results of computational experiments with our B&P algorithm. The method has been coded in C++ on top of the PDCGM library (GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2013; GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2016), which offers a efficient stabilized interior-point column generation method. The B&P search tree follows the interior-point B&P framework described in detail by Munari and Gondzio (2013). In addition, the primal heuristic (Section 4.5) uses the IBM CPLEX Optimization Studio version 12.10 to solve the resulting MIP problems. All the experiments were run on a PC with an Intel Core i7-4790 3.6GHz processor and 16 GB of memory (which are the same hardware conditions indicated in Chapter 3). Our experimentation uses the same six months of journey logs (M1-M6) as defined in Section 3.4. We remind that the real-life-based instances are presented in Table 4.

The next subsections aim to analyze and compare the computational results obtained from B&P variants and then confront the best variant with the approaches presented in Chapter 3. We set a time limit of one hour and a relative optimality tolerance of 0.01%.

4.6.1 Results from B&P variants

To better assess the contribution that each component/method (Subsections 4.2-4.5) plays within our B&P, we solve the instances, evaluating:

- Switch between two-step and strong branching (Subsection 4.4); and
- Turn off and on the primal heuristic (PH) (Subsection 4.5).

Hence, the experiment addresses a total of four B&P variants, which are represented by the nomenclature [*Branching strategy*: 0→two-step branching or 1→strong branching, *PH*: 0→off or 1→on]. Therefore, B&P[1, 0] refers to a B&P using the strong branching strategy, without applying the primal heuristic.

Table 10 reports the averages for each month and B&P variant, obtained by solving the real-life instances. It was structured as follows. The first and second columns identify the month and B&P variant. Columns OF_{lb} and OF_{ub} are, in that order, the dual (lower) bound and the primal (upper) bound. Consecutively, the fourth column (*gap*) calculates the relative optimality gap, given by $100\%.(OF_{ub} - OF_{lb})/(OF_{ub} + 10^{-10})$. $nCol$ and $nNode$ bring the number of columns and nodes generated by the method. Finally, the last three columns ($RMPt$, SPt and $CPUt$) correspond the computation time related to the RMPs, subproblems and the B&P method as a whole, respectively. As a complement, we present the results of this experimentation per instance in Appendix B.2.

As can be seen in Table 10, all the B&P variants were able to achieve optimal results in all analyzed instances. However, in terms of computing time, there are some interesting insights when comparing the B&P variants. Firstly, we can notice that months M4 and M5 tend to require the most computational effort, in accordance with what was related in Subsections 3.4.2 and 3.4.3.

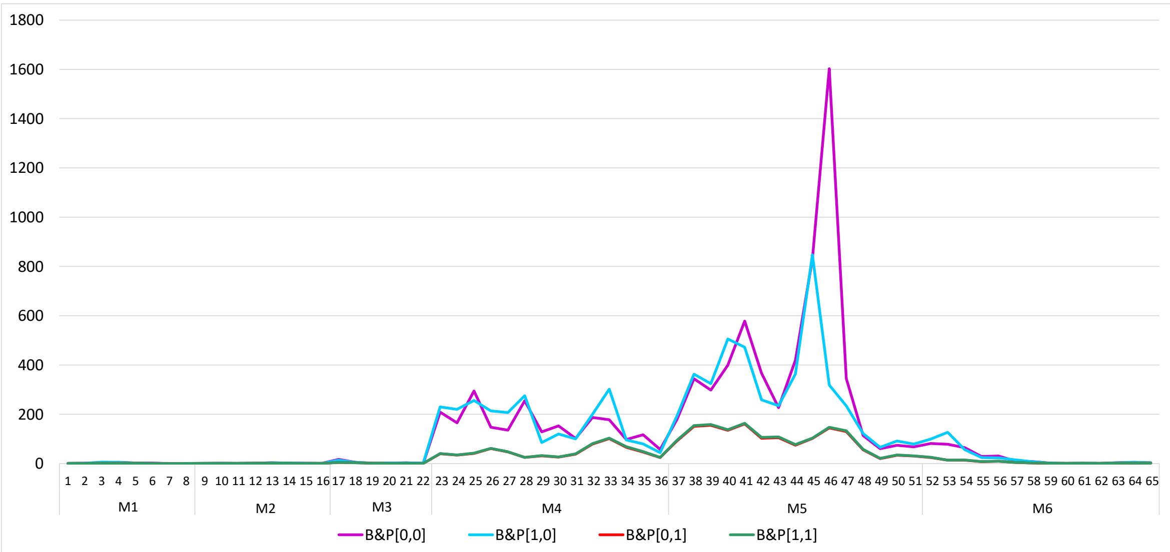
Table 10 – Average of computational results obtained by the B&P variants.

Month	B&P var	OF_{lb}	OF_{ub}	*gap*	$nCol$	$nNode$	$RMPt$	SPt	$CPUt$
M1	B&P[0, 0]	74,726.87	74,726.87	0.000%	416.00	2.75	0.113	1.699	1.898
	B&P[1, 0]	74,726.87	74,726.87	0.000%	497.63	2.75	0.122	1.754	2.078
	B&P[0, 1]	74,726.49	74,726.87	0.001%	406.63	0.88	0.064	0.699	0.811
	B&P[1, 1]	74,726.49	74,726.87	0.001%	406.63	0.88	0.064	0.705	0.862
M2	B&P[0, 0]	164,788.01	164,788.01	0.000%	317.00	1.63	0.060	1.182	1.307
	B&P[1, 0]	164,788.01	164,788.01	0.000%	356.88	1.63	0.063	1.198	1.342
	B&P[0, 1]	164,788.01	164,788.01	0.000%	316.38	1.00	0.047	0.787	0.887
	B&P[1, 1]	164,788.01	164,788.01	0.000%	316.38	1.00	0.049	0.794	0.915
M3	B&P[0, 0]	292,138.81	292,138.81	0.000%	594.33	4.50	0.172	4.778	5.095
	B&P[1, 0]	292,138.81	292,138.81	0.000%	655.83	3.83	0.164	4.087	4.565
	B&P[0, 1]	292,138.78	292,138.78	0.000%	588.83	1.00	0.087	1.715	1.861
	B&P[1, 1]	292,138.78	292,138.78	0.000%	588.83	1.00	0.087	1.731	1.950
M4	B&P[0, 0]	415,261.37	415,261.37	0.000%	1,901.14	18.71	2.425	155.405	159.094
	B&P[1, 0]	415,261.37	415,261.37	0.000%	2,215.07	18.50	2.705	164.825	173.480
	B&P[0, 1]	415,260.24	415,261.42	0.000%	1,815.57	0.93	0.989	45.944	47.263
	B&P[1, 1]	415,260.22	415,261.39	0.000%	1,815.57	0.93	0.990	46.767	48.488
M5	B&P[0, 0]	910,642.52	910,642.52	0.000%	1,910.27	21.27	3.890	387.947	393.884
	B&P[1, 0]	910,642.51	910,642.51	0.000%	2,015.13	15.27	3.202	288.961	298.276
	B&P[0, 1]	910,638.18	910,642.48	0.000%	1,801.33	0.93	1.541	97.604	99.602
	B&P[1, 1]	910,638.16	910,642.46	0.000%	1,801.33	0.93	1.536	99.453	101.933
M6	B&P[0, 0]	372,179.53	372,179.53	0.000%	978.57	7.64	0.486	21.908	22.748
	B&P[1, 0]	372,179.53	372,179.53	0.000%	1,064.79	7.50	0.552	24.433	26.379
	B&P[0, 1]	372,179.51	372,179.51	0.000%	942.29	1.00	0.192	5.740	6.030
	B&P[1, 1]	372,179.51	372,179.51	0.000%	942.29	1.00	0.192	5.786	6.206

Source: Own authorship.

Now for comparison purposes, consider Figures 23 and 24. Figure 23 shows a line chart, where on the abscissa axis there are the instances of M1-M6, numbered sequentially from 1 to 65 (M1_1to3, . . . , M6_14to16) to facilitate the visualization, and the ordinate axis shows $CPUt$. Therefore, Figure 23 portrays the computing time performance of the variants when solving the instances. The line chart makes clear the formation of two competing groups, B&P variants with PH turned off and on. Comparing the first group separately (B&P[0, 0] and B&P[1, 0]), we found that the strong branching strategy tends to improve the overall performance of the B&P method. This can be explained due to the behavior that strong branching has (of first evaluating a set of variables and then actually branching), combined with the fact that in these B&P variants it was necessary to go much further down on the B&B tree than in the second group (see column $nNode$ in tables 40 and 41 at Appendix B.2), which gives this strategy an advantage. Although, in the second group (B&P[0, 1] and B&P[1, 1]), two-step branching proved to be a slightly more efficient, since the exploration of nodes in the B&B tree was limited to at most one (as can be seen in $nNode$ from Tables 42 and 43).

Figure 23 – Comparison of the B&P variants in relation to the runtime.

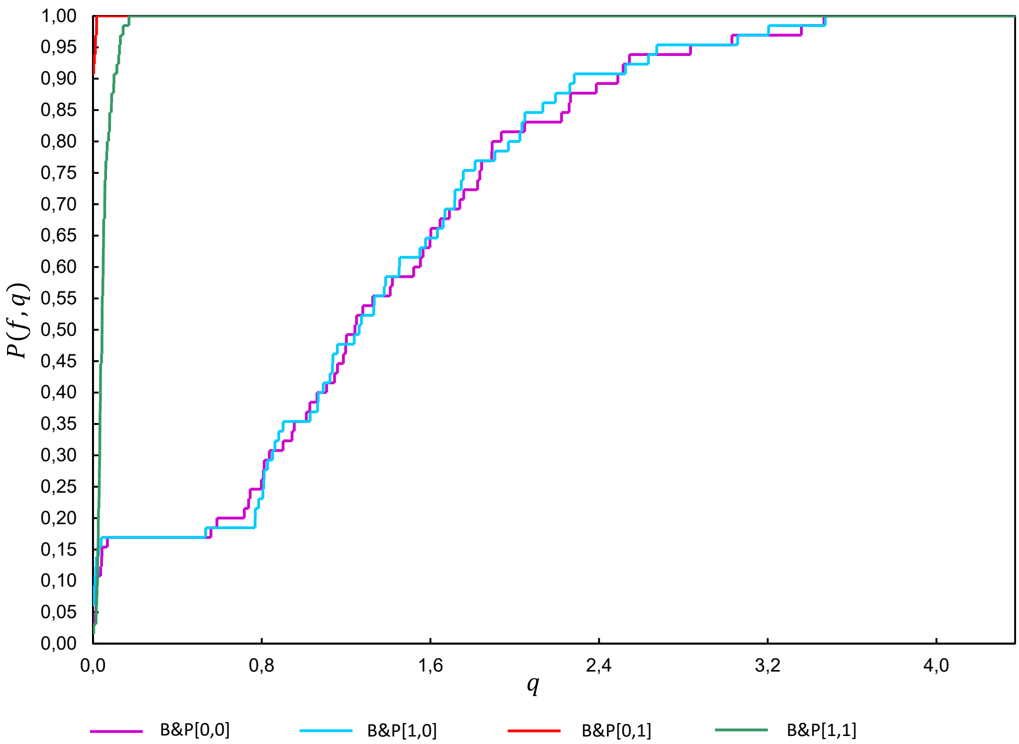


Source: Own authorship.

What was verified in Figure 23 can be further explained in Figure 24, which illustrates the computational time using a performance profile chart. This technique was introduced in Dolan and Moré (2002), and since then it has been widely used to compare algorithms and their computational implementation. Roughly speaking, a performance profile of a software or algorithm can be defined as the cumulative distribution function for a given performance metric, which in our case is $CPUt$. By the chart, $P(f, q)$ is a function that is associated to a tolerance value $q \in \mathbb{R}$, which provides the fraction of instances solved by an approach with performance within a factor q of the best performance metric. In other words, $P(f, q)$ can be understood as the probability that an approach solves a given instance in no more than q times the minimum computing time taken by any proposed algorithm. To fix ideas, suppose we set $q = 4$. This is equivalent to saying that we are willing to accept a performance on each instance up to four times worse than the best performance on that same problem instance. Thus, if $P(f, 4) = 0.7$, for example, this means the solution approach is able to solve 70% of the instances within this tolerance (see Dolan and Moré (2002) for a detailed explanation of this performance profiles technique).

From Figure 24, we observe that the performance profile curve of the B&P$[0, 1]$ variant performed best compared to the others in around 90% of the solved instances ($q = 0$). Since the performance curve associated with the B&P$[0, 1]$ variant is above the others over the interval $q \in [0, 4.47]$, then it dominates those remaining. This means that B&P$[0, 1]$ stands out in relation to B&P$[0, 0]$, B&P$[1, 0]$ and B&P$[1, 1]$ algorithms, in the sense that it managed to solve more problems within a factor q of the performance of any other approach. The previous comment is also valid when comparing only the performance curves of B&P$[0, 0]$, B&P$[1, 0]$ and B&P$[1, 1]$ variants, as the curve of B&P$[1, 1]$ is above the other performance profile curves associated with B&P$[0, 0]$, B&P$[1, 0]$. As previously discussed in Figure 23, we can notice the same two competing groups.

Figure 24 – Performance profile of the B&P variants, using the runtime as metric.



Source: Own authorship.

4.6.2 Comparison with the approaches from Chapter 3

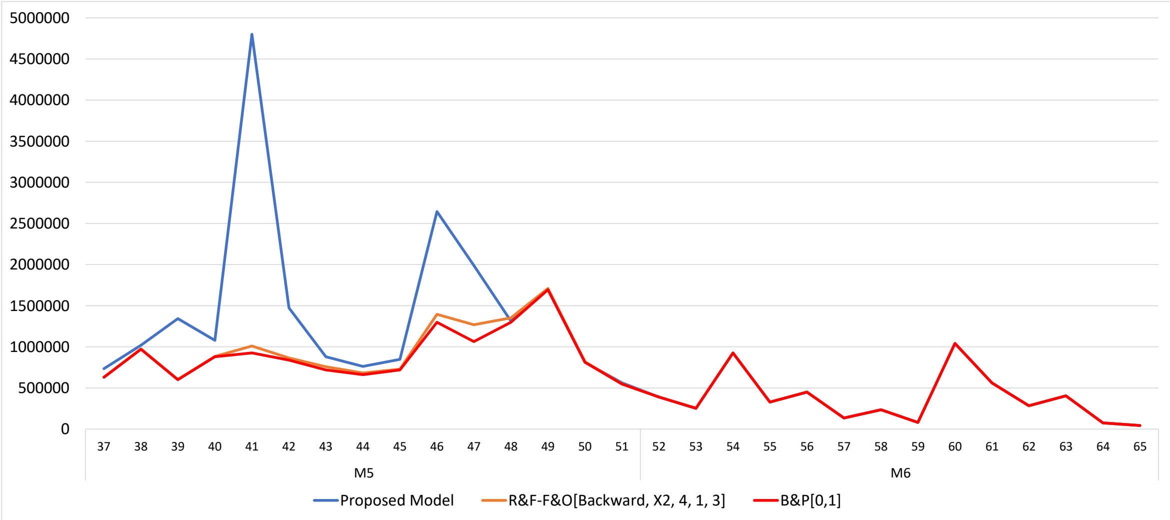
In the previous chapter, we presented a compact MIP model capable of representing the problem in detail, to then be solved by a general-purpose MIP solver in order to have a solution approach (called the proposed model). Furthermore, we suggest MIP-based heuristics such as relax-and-fix (R&F) and fix-and-optimize (F&O). The experiments tested different partition sizes, parameters and forward/backward temporal strategies in the R&F-F&O approach. Among the evaluated variants, the R&F-F&O heuristic with backward strategy, fixation about flight sequences, four R&F partitions, one partition with linearly relaxed variables and three F&O partitions, was the one that excelled, being the only one that found quality solutions in all instances. It is referenced by R&F-F&O[$Backward$, $X2$, 4, 1, 3].

In this subsection, we compare B&P$[0,1]$ with the proposed model and R&F-F&O[$Backward$, $X2$, 4, 1, 3], by the quality criteria of the objective function values and computational performances. Our comparison focuses on months M5 and M6, because they include all the crew rules (pilot time windows) and because they have the largest instances, consistent with what was done in Subsection 3.4.3.

Figures 25-28 present charts similar to those in Figures 23 and 24. The metric used in Figures 25 and 26 is the objective function, and in Figures 27 and 28, the computing time. As expected, B&P$[0,1]$ easily dominates R&F-F&O[$Backward$, $X2$, 4, 1, 3] and the proposed model and, in turn, R&F-F&O[$Backward$, $X2$, 4, 1, 3] dominates the proposed model. From Figures 26 and 28, since the performance profile values of B&P$[0,1]$ coincide at $q = 0$ and $q \to \infty$, we can conclude that, in all instances, B&P$[0,1]$ always determined solutions with the lowest objective function value and computing time when compared to the values obtained by the other approaches.

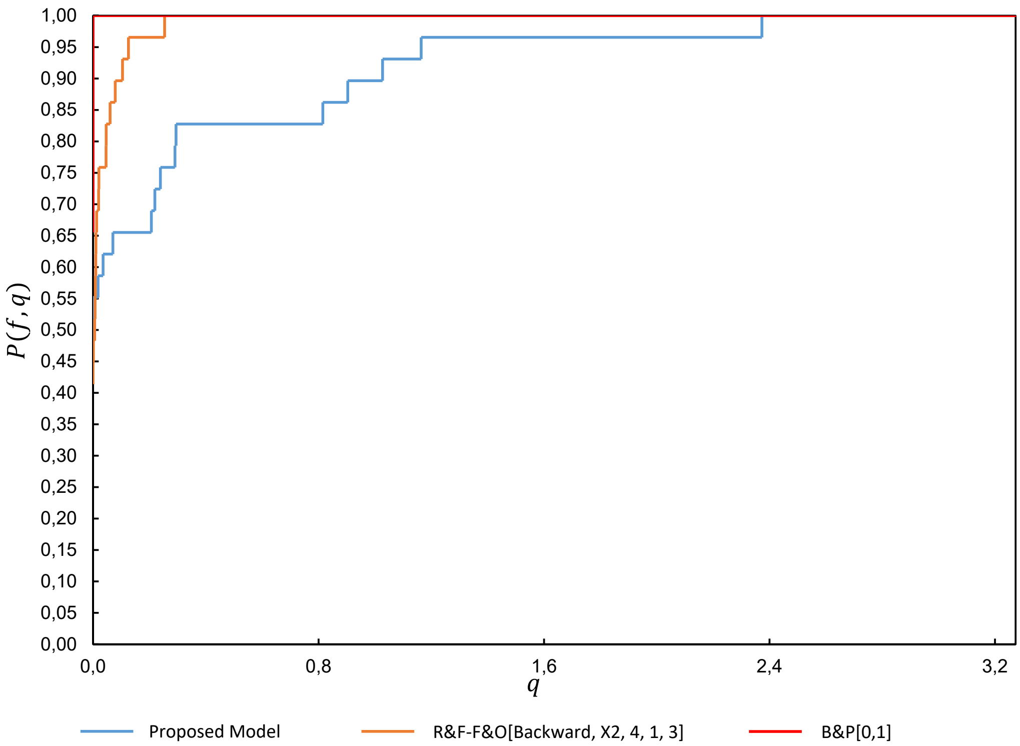
According to the analyzed results, although the B&P algorithm is exact, and the R&F-F&O approach is a heuristic, all the B&P variants proved to be superior, both in terms of solution quality (in which all results were optimal) and computational performance (taking less than 400 s on average). Therefore, we can conclude that the reformulation and B&P variants are appropriate for solving real-life instances in practice, helping the decision-makers to efficiently determine which routes each aircraft should perform, taking into account the fleet characteristics and the different operational costs, as well as aspects of crew regulation, considered fundamental.

Figure 25 – Comparison between B&P[0, 1] and the approaches from Chapter 3, in relation to the upper bound.



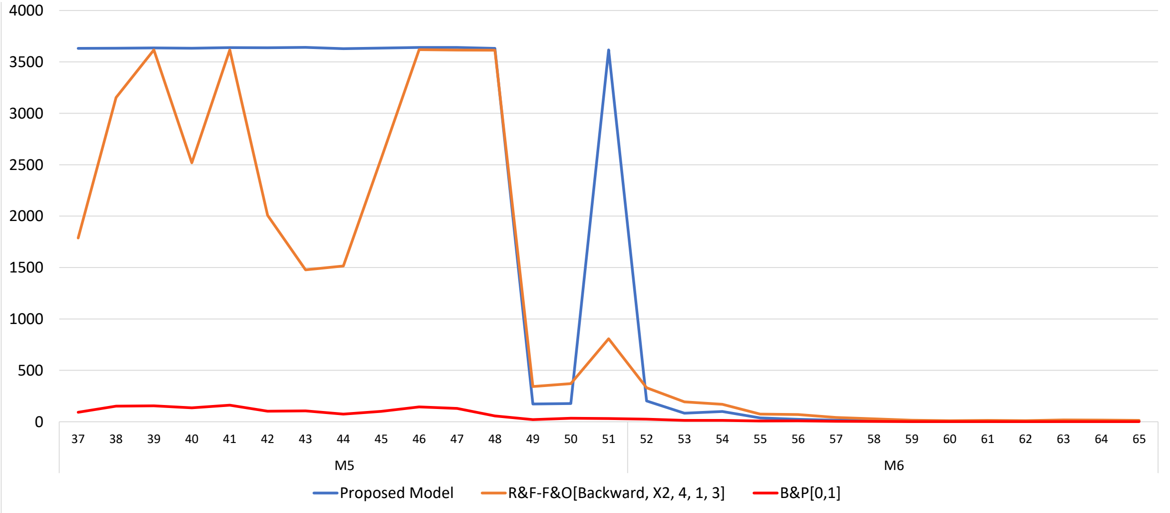
Source: Own authorship.

Figure 26 – Performance profile between B&P[0, 1] and the approaches from Chapter 3, using the upper bound as metric.



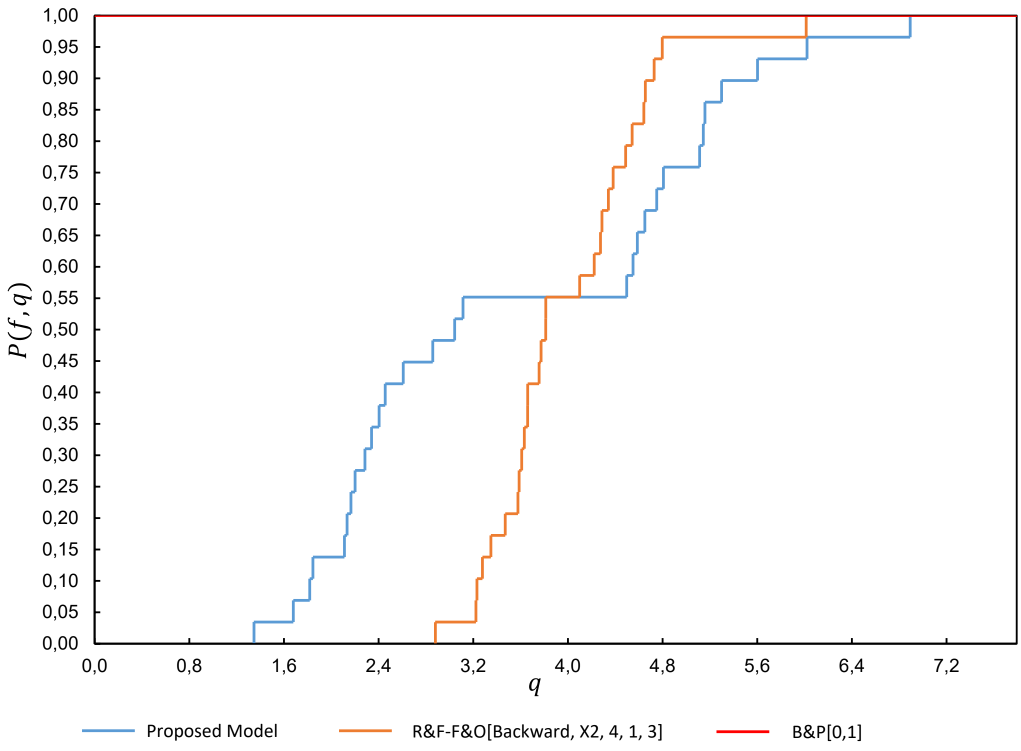
Source: Own authorship.

Figure 27 – Comparison between B&P[0, 1] and the approaches from Chapter 3, in relation to the runtime.



Source: Own authorship.

Figure 28 – Performance profile between B&P[0, 1] and the approaches from Chapter 3, using the runtime as metric.



Source: Own authorship.

Chapter 5

Rescheduling helicopter flights for personnel transportation in the oil industry

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 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, aircraft breakdowns, absent crew members, political reasons and other different causes. 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; HERMETO; HAWSON, 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. For simplicity, in this study, all problem parameters are recognized 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.

We assume that each flight is planned for only one maritime unit and there are no splitting or merging of flights. Furthermore, this ARP does not operate with connection flights, as passengers cannot wait at the maritime units due to safety reasons (each takeoff 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 work 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 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 chapter is organized as follows. In Section 5.1, we describe in details the addressed flight rescheduling problem. Then, we propose two alternative MIP formulations to model this problem in Section 5.2, and develop the tailor-made heuristic approaches in Section 5.3. Computational results using real-life data of the Brazilian company are presented and analyzed in Section 5.4. These results show that the heuristics are effective for solving realistic problem instances in practical settings.

5.1 Problem description

As aforesaid, 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 takeoff 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 takeoff 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 takeoffs 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.

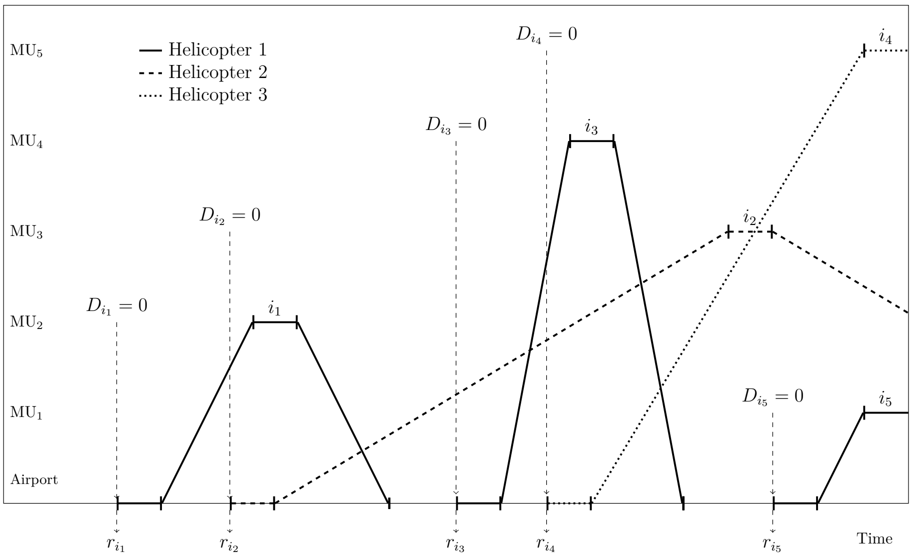
iii) Flights: Each flight is defined by its maritime unit destination and its departure time from the airport on the current day. We represent it as a simple sequence (route): airport - maritime unit - airport. Moreover, each flight has a previously assigned group of passengers going to the maritime 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 Figure 29 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 Figure 29. 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 illustration. Figure 30 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 Figure 29). 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 Figure 30 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 dissertation 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).

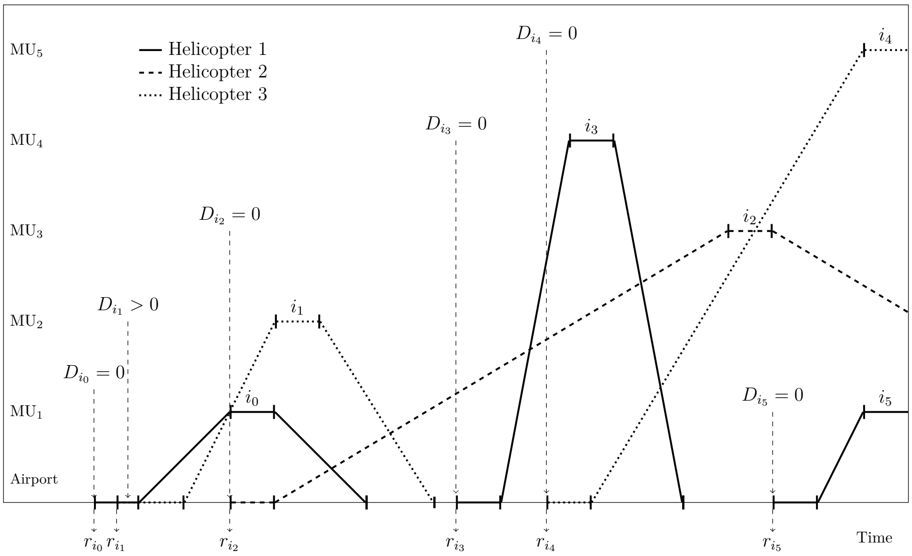
Figure 29 – Time-space diagram illustrating a schedule with five flights, five maritime units and three aircraft.



Source: Own authorship.

In the next section, we present two MIP models that fully formulates the described problem, while in Section 5.3 we develop a tailor-made heuristic able to find relatively good feasible solutions within acceptable computational times.

Figure 30 – Time-space diagram illustrating the reschedule with five flights, five maritime units and three aircraft.



Source: Own authorship.

5.2 Mathematical formulations

This section introduces the notation used throughout the chapter 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 5.1, 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 (called turnaround time);
- sb: minimum time between two consecutive takeoffs in the runway of the airport (known as safety briefing);
- tu_i: time spent in the maritime unit for flight i;
- $d_I^{\max}$: maximum delay (of type I) allowed in the departure time of a table flight (15 minutes);
- $d_{II}^{\max}$: 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 7:00 am and 6:00 pm);
- M: a sufficiently large positive number.

5.2.1 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. We create two dummy flights, 0 and $n+1$, and impose that the flight plan of each helicopter begins with flight 0 and ends with flight $n+1$. 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 $d_I^{\max}$, for $i \in \mathcal{I}_C \cup \mathcal{I}_0$;
- $B_i^{II} \in \{0,1\}$: assumes the value 1 if and only if the delay of flight i is greater than $d_I^{\max}$ but less than or equal to $d_{II}^{\max}$, for $i \in \mathcal{I}_C \cup \mathcal{I}_0$;
- $D_i \geq 0$: 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 (237) 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.

$$\min \quad f = \sum_{i=1}^{11} w_i.f_i; \tag{237}$$

where each component f_i is defined as:

$$f_1 := \left(|\mathcal{I}_C| - \sum_{i\in\mathcal{I}_C}\sum_{h\in\mathcal{H}_i} Y_{ih}\right); \quad f_2 := \left(|\mathcal{I}_2| - \sum_{i\in\mathcal{I}_2}\sum_{h\in\mathcal{H}_i} Y_{ih}\right); \quad f_3 := \left(|\mathcal{I}_1| - \sum_{i\in\mathcal{I}_1}\sum_{h\in\mathcal{H}_i} Y_{ih}\right);$$

$$f_4 := \left(|\mathcal{I}_0| - \sum_{i\in\mathcal{I}_0}\sum_{h\in\mathcal{H}_i} Y_{ih}\right); \quad f_5 := \sum_{h\in\mathcal{H}_s} V_h; \quad f_6 := \sum_{h\in\mathcal{H}_p} V_h; \quad f_7 := \sum_{h\in\mathcal{H}_n} V_h;$$

$$f_8 := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} B_i^{II}; \quad f_9 := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} B_i^{I}; \quad f_{10} := \sum_{i \in \mathcal{I}_C \cup \mathcal{I}_0} \sum_{\substack{h \in \mathcal{H}_i: \\ h \neq s_i}} Y_{ih}; \quad f_{11} := \sum_{i \in \mathcal{I}} D_i.$$

Having settled the objective function, we define the flow constraints (238)-(240). Detailing, (238) guarantee that if flight i is in planning, it has a single immediate success; (239) assure the flow conservation through the network; and (240) force dummy nodes 0 and $n+1$ to be the first and last flights of each helicopter, linking the aircraft usage by variable V_h.

$$\sum_{j \in \mathcal{I}_h \cup \{n+1\}} X_{ijh} = Y_{ih}; \;\; \forall \;\; i \in \mathcal{I}; h \in \mathcal{H}_i; \tag{238}$$

$$\sum_{j \in \{0\} \cup \mathcal{I}_h} X_{jih} = \sum_{j \in \mathcal{I}_h \cup \{n+1\}} X_{ijh}; \;\; \forall \;\; i \in \mathcal{I}; h \in \mathcal{H}_i; \tag{239}$$

$$\sum_{j \in \mathcal{I}} \sum_{h \in \mathcal{H}_j} X_{0jh} = \sum_{i \in \mathcal{I}} \sum_{h \in \mathcal{H}_i} X_{i(n+1)h} = \sum_{h \in \mathcal{H}} V_h. \tag{240}$$

Constraints (241) state that each flight can be assigned to at most one aircraft, while (242) assure that such assignment can only happen if this aircraft is used.

$$\sum_{h \in \mathcal{H}_i} Y_{ih} \leq 1; \;\; \forall \;\; i \in \mathcal{I}; \tag{241}$$

$$Y_{ih} \leq V_h; \;\; \forall \;\; h \in \mathcal{H}; i \in \mathcal{I}_h. \tag{242}$$

(243)-(246) are temporal constraints. (243) impose the synchronization of flight departures, including turnaround times; (244) preserve the flight durations, determining each arrival instant; (245) satisfy the planned departure time of the flights, quantifying the resulting delay for each flight; and (246) respect the closing time of the airport time window.

$$DT_j \geq AT_i + tat \;. \sum_{h \in \mathcal{H}_i \cap \mathcal{H}_j} X_{ijh} - M. \left(1 - \sum_{h \in \mathcal{H}_i \cap \mathcal{H}_j} X_{ijh}\right); \;\; \forall \;\; i,j \in \mathcal{I} \mid i \neq j; \tag{243}$$

$$AT_i = DT_i + tf_i. \sum_{h \in \mathcal{H}_i} Y_{ih}; \;\; \forall \;\; i \in \mathcal{I}; \tag{244}$$

$$r_i. \sum_{h \in \mathcal{H}_i} Y_{ih} \leq DT_i \leq r_i. \sum_{h \in \mathcal{H}_i} Y_{ih} + D_i; \;\; \forall \;\; i \in \mathcal{I}; \tag{245}$$

$$AT_i \leq tw^B. \sum_{h \in \mathcal{H}_i} Y_{ih}; \;\; \forall \;\; i \in \mathcal{I}. \tag{246}$$

Constraints (247) and (248) activate the binary variables related to the type of delays (I or II) incurring to table and entourage flights. Specifically, constraints (247)[1] ensure that a allocated flight can have at most one type of delay, while constraints (248) limit the delay duration according to the delay type. Note that, together with the weights assigned

[1] As a matter of fact, these constraints are valid inequalities because the objective function (237) penalizes variables B_i^I and B_i^{II}, so that only one of these variables is activated when we find a feasible (integer) solution.

to B_i^I and B_i^{II} in the objective function (237), these constraints model the piecewise linear behavior of the penalties applied in terms of the delay duration.

$$B_i^I + B_i^{II} \leq 1; \;\; \forall \; i \in \mathcal{I}_C \cup \mathcal{I}_0; \quad (247)$$
$$D_i \leq d_I^{\max}.B_i^I + d_{II}^{\max}.B_i^{II}; \;\; \forall \; i \in \mathcal{I}_C \cup \mathcal{I}_0. \quad (248)$$

If flights i and j are scheduled in a solution, constraints (249)-(251) impose a single order of precedence between them.

$$Z_{ij} + Z_{ji} \leq 1; \;\; \forall \; i,j \in \mathcal{I} \mid i \neq j; \quad (249)$$
$$Z_{ij} + Z_{ji} \geq \sum_{h \in \mathcal{H}_j} Y_{jh} + \sum_{h \in \mathcal{H}_i} Y_{ih} - 1; \;\; \forall \; i,j \in \mathcal{I} \mid i \neq j; \quad (250)$$
$$2.\,(Z_{ij} + Z_{ji}) \leq \sum_{h \in \mathcal{H}_i} Y_{ih} + \sum_{h \in \mathcal{H}_j} Y_{jh}; \;\; \forall \; i,j \in \mathcal{I} \mid i \neq j. \quad (251)$$

Constraints (252) and (253) enforce the synchronization of the departure of flights from the airport and their arrival at the maritime units, guarding the safety briefing and each time spent at a MU, respectively.

$$DT_j - DT_i \geq sb.Z_{ij} - M.(1 - Z_{ij}); \;\; \forall \; i,j \in \mathcal{I} \mid i \neq j; \quad (252)$$
$$DT_j - DT_i \geq tu_i.Z_{ij} - M.(1 - Z_{ij}); \;\; \forall \; p \in \mathcal{P}; i,j \in \mathcal{I}_p \mid i \neq j. \quad (253)$$

Constraints (254) 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 (255) block the maritime unit of destination of the entourage flights, while constraints (256) block the helicopter that performs them.

$$Z_{ij} = 0; \;\; \forall \; i \in \mathcal{I}_0 \cap \mathcal{I}_p; j \in (\mathcal{I}_1 \cup \mathcal{I}_2) \cap \mathcal{I}_p; p \in \mathcal{P}; \quad (254)$$
$$Z_{ij} = 0; \;\; \forall \; i \in \mathcal{I}_C \cap \mathcal{I}_p; j \in \mathcal{I} \setminus \mathcal{I}_C \cap \mathcal{I}_p; p \in \mathcal{P}. \quad (255)$$
$$X_{i(n+1)h} = Y_{ih}; \;\; \forall \; i \in \mathcal{I}_C; h \in \mathcal{H}_i. \quad (256)$$

Finally, constraints (257)-(262) define the type and domain of the decision variables.

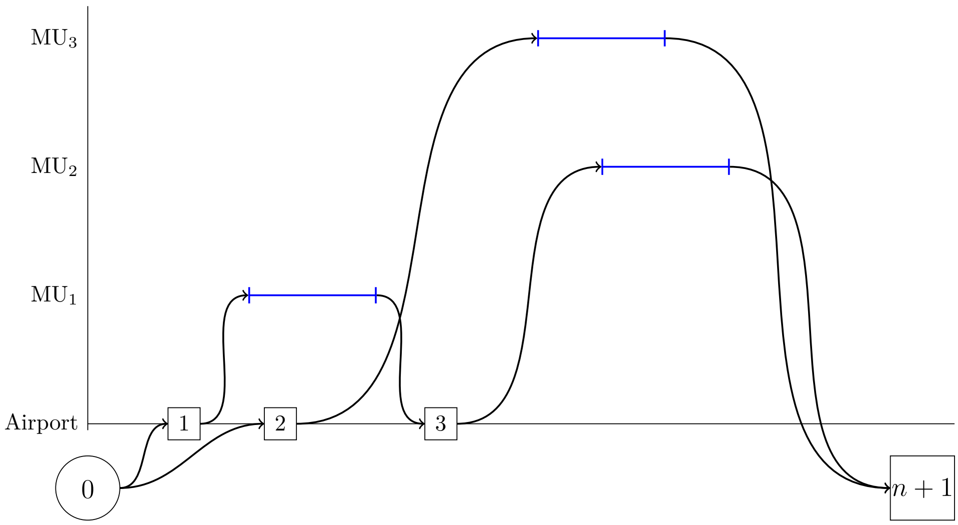
$$X_{ijh} \in \{0,1\}; \;\; \forall \; i \in \{0\} \cup \mathcal{I}; j \in \mathcal{I} \cup \{n+1\} \mid i \neq j; h \in \mathcal{H}; \quad (257)$$
$$Y_{ih} \in \{0,1\}; \;\; \forall \; i \in \mathcal{I}; h \in \mathcal{H}; \quad (258)$$
$$Z_{ij} \in \{0,1\}; \;\; \forall \; i,j \in \mathcal{I}; \quad (259)$$
$$V_h \in \{0,1\} \;\; \forall \; h \in \mathcal{H}; \quad (260)$$
$$B_i^I \in \{0,1\}, B_i^{II} \in \{0,1\}; \;\; \forall \; i \in \mathcal{I}; \quad (261)$$
$$DT_i \geq 0, AT_i \geq 0, D_i \geq 0; \;\; \forall \; i \in \mathcal{I}. \quad (262)$$

5.2.2 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 Figure 31, 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 takeoff 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 takeoff events coincides with the total number of flights in the instance (note that there are three takeoff events in Figure 31) and each flight must be assigned to an event.

Figure 31 – Schematic representation of the event-based formulation for a problem instance with three flights.



Source: Own authorship.

There are two routes in the feasible solution presented in Figure 31, 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 takeoff events from the airport and round-trip flights to maritime units. Thus, the problem is represented by a set of takeoff 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, as also done in the previous model. 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 takeoff event e;
- $W_e \geq 0$: starting time of event e.

The objective function is defined similarly as in the network-based model using (237), but since there are different types of variables in the event-based model, we need to redefine the following terms:

$$f_1 := \left(|\mathcal{I}_C| - \sum_{e\in\mathcal{E}} \sum_{i\in\mathcal{I}_C} \sum_{h\in\mathcal{H}_i} X_{eih}\right); \quad f_2 := \left(|\mathcal{I}_2| - \sum_{e\in\mathcal{E}} \sum_{i\in\mathcal{I}_2} \sum_{h\in\mathcal{H}_i} X_{eih}\right);$$

$$f_3 := \left(|\mathcal{I}_1| - \sum_{e\in\mathcal{E}} \sum_{i\in\mathcal{I}_1} \sum_{h\in\mathcal{H}_i} X_{eih}\right);$$

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

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 (237).

The first block of constraints guarantee that aircraft routes correspond to a sequence of airport takeoff events and round-trip flights. Constraints (263) ensure that if flight i is not transferred to the next day, then it should be performed by a single helicopter using a single takeoff event. Constraints (264) ensure that if the takeoff event e is used, then a single helicopter uses it to perform a single flight. Constraints (265) relate the usage of a helicopter with the assignment of this helicopter to a flight.

$$\sum_{e\,\in\mathcal{E}} \sum_{h\,\in\mathcal{H}_i} X_{eih} \leq 1; \;\; \forall \;\, i \in \mathcal{I}; \tag{263}$$

$$\sum_{i\,\in\mathcal{I}} \sum_{h\,\in\mathcal{H}_i} X_{eih} \leq 1; \;\; \forall \;\, e \in \mathcal{E}; \tag{264}$$

$$\sum_{e\,\in\mathcal{E}} X_{eih} \leq V_h, \forall \;\, i \in \mathcal{I}; h \in \mathcal{H}_i. \tag{265}$$

Constraints (266) ensure that takeoff events do not overlap in the airport runway, as they impose a minimum time interval sb 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 sb. Constraints (267) synchronize the starting times of two takeoff events of the same helicopter. If $X_{eih} = 1$ then helicopter h uses the takeoff 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 (268) 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.

$$W_e \geq W_{(e-1)} + sb.\sum_{i\in\mathcal{I}} \sum_{h\in\mathcal{H}_i} X_{eih}; \;\; \forall \;\, e \in \mathcal{E} \cup \{n+1\}; \tag{266}$$

$$W_g \geq W_e + (tf_i + tat).\left(X_{eih} + \sum_{j\in\mathcal{I}} X_{gjh} - 1\right); \qquad \forall \;\, e, g \in \mathcal{E} \mid g > e; i \in \mathcal{I}; h \in \mathcal{H}_i; \tag{267}$$

$$W_{n+1} \geq W_e + tf_i.X_{eih}; \;\; \forall \; e \in \mathcal{E}; i \in \mathcal{I}; h \in \mathcal{H}_i. \tag{268}$$

Constraints (269) act as time windows for the starting time of a takeoff event e assigned to a flight i. If there is an aircraft $h \in \mathcal{H}_i$ with this assignment (i.e., $X_{eih} = 1$), 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 (270) ensure that all events satisfy the airport time window.

$$r_i. \sum_{h \in \mathcal{H}_i} X_{eih} \leq W_e \leq r_i. \sum_{h \in \mathcal{H}_i} X_{eih} + D_i + tw^B. \left(1 - \sum_{h \in \mathcal{H}_i} X_{eih}\right); \;\; \forall \; e \in \mathcal{E}; i \in \mathcal{I}; \tag{269}$$

$$tw^A \leq W_e \leq tw^B; \;\; \forall \; e \in \mathcal{E} \cup \{0, n+1\}. \tag{270}$$

Constraints (271)[2] and (272) are similar to constraints (247) and (248) 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 (271), these constraints also ensure that a flight cannot be delayed if it is not assigned to any aircraft and event.

$$B_i^I + B_i^{II} \leq \sum_{e \in \mathcal{E}} \sum_{h \in \mathcal{H}_i} X_{eih}; \;\; \forall \; i \in \mathcal{I}_C \cup \mathcal{I}_0; \tag{271}$$

$$D_i \leq d_I^{\max}.B_i^I + d_{II}^{\max}.B_i^{II}; \;\; \forall \; i \in \mathcal{I}_C \cup \mathcal{I}_0. \tag{272}$$

Constraints (273) impose the minimum time interval between consecutive takeoffs of different helicopters performing different flights (i and j) to the same maritime unit destination (p) (i.e., if $\sum_{h \in \mathcal{H}_i} X_{eih} = \sum_{h \in \mathcal{H}_j} X_{gjh} = 1$). 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.

$$W_g \geq W_e + tu_i. \left(\sum_{h \in \mathcal{H}_i} X_{eih} + \sum_{h \in \mathcal{H}_j} X_{gjh} - 1\right); \; \forall \, p \in \mathcal{P}; i, j \in \mathcal{I}_p \mid i \neq j; e, g \in \mathcal{E} \mid g > e. \tag{273}$$

Constraints (274) impose the precedence order between a transferred flight and a table flight going to the same maritime unit. Constraints (275) and (276) guarantee that the helicopter performing an entourage flight blocks the corresponding maritime unit after landing, until the end of the day.

$$\sum_{h \in \mathcal{H}_j} \sum_{g=1}^{e-1} X_{gjh} \leq 1 - \sum_{h \in \mathcal{H}_i} X_{eih}; \;\; \forall \; p \in \mathcal{P}; i \in \mathcal{I}_2 \cup \mathcal{I}_1 \cap \mathcal{I}_p; j \in \mathcal{I}_0 \cap \mathcal{I}_p; e \in \mathcal{E}; \tag{274}$$

$$\sum_{h \in \mathcal{H}_j} \sum_{g=e+1}^{|\mathcal{E}|} X_{gjh} \leq 1 - \sum_{h \in \mathcal{H}_i} X_{eih}; \;\; \forall \; p \in \mathcal{P}; i \in \mathcal{I}_C \cap \mathcal{I}_p; j \in \mathcal{I} \setminus \mathcal{I}_C \cap \mathcal{I}_p; e \in \mathcal{E}; \tag{275}$$

$$\sum_{g=e+1}^{|\mathcal{E}|} X_{gjh} \leq 1 - X_{eih}; \;\; \forall \; i \in \mathcal{I}_C; j \in \mathcal{I} \mid j \neq i; e \in \mathcal{E}; h \in \mathcal{H}_i \cap \mathcal{H}_j. \tag{276}$$

[2] Analogous to (247), these constraints are also valid inequalities.

Finally, constraints (277)-(281) impose the domain of the decision variables.

$$X_{eih} \in \{0,1\}; \ \forall \ e \in \mathcal{E} \cup \{0, n+1\}; i \in \mathcal{I}; h \in \mathcal{H}; \tag{277}$$
$$V_h \in \{0,1\}; \ \forall \ h \in \mathcal{H}; \tag{278}$$
$$B_i^I \in \{0,1\}, B_i^{II} \in \{0,1\}; \ \forall \ i \in \mathcal{I}; \tag{279}$$
$$W_e \geq 0; \ \forall \ e \in \mathcal{E} \cup \{0, n+1\}; \tag{280}$$
$$D_i \geq 0; \ \forall \ i \in \mathcal{I}. \tag{281}$$

It is worthy of note that the proposed event-based formulation can be extended to the more general case that includes not only takeoff 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 takeoff and another for landing. We have not included landing events in our approach because the company disregards overlaps between two landing events or between takeoff 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.

5.3 Heuristic approach

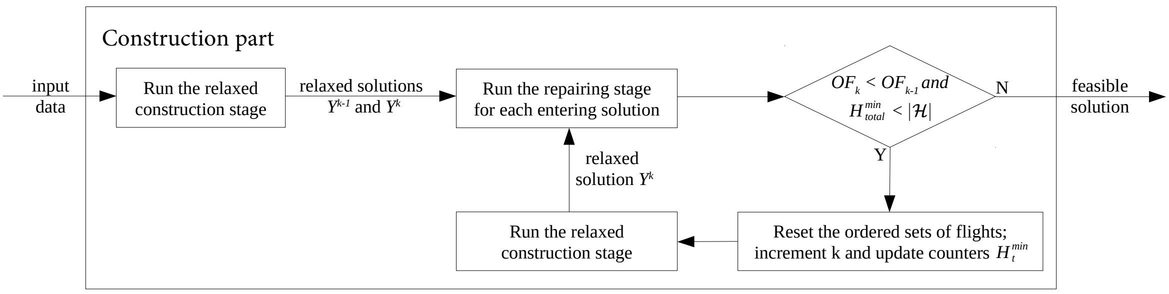
Our first heuristic approach to this problem was based on a flight sequencing and a straightforward aircraft allocation procedure. The flight sequencing was initially built by an insertion algorithm inspired on the classic NEH procedure (NAWAZ; ENSCORE; HAM, 1983). Then, three local search methods were applied to improve the sequence based on swapping neighborhoods. The implementation of this algorithm was able to run on a very short time (typically less than one second for realistic company instances), but the aircraft allocation procedure did not allow the algorithm to obtain good solutions. The main cause was that the heuristic had no look-ahead procedure, and hence the flights were allocated to the aircraft generating a good local choice, but that was later revealed as a poor global choice. This showed us the need for a different approach with a global improvement method.

Thus, a second tailor-made heuristic approach was developed with more focus on the aircraft allocation issues. This heuristic runs for longer in comparison with the first, yet it generates considerable better solutions. We present its details on the remainder of this section and analyze its computational performance in Section 5.4. It consists of a construction part followed by an improvement part. The construction part 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 part tries to improve this feasible solution based on six different local search movements, as described below and in more detail in Appendix C.

5.3.1 Construction part

The main steps of the construction part are summarized in Figure 32, 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 Subsection 5.3.1.1. This part 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 5.3.1.1 and 5.3.1.2, respectively.

Figure 32 – Main steps of the proposed construction part.



Source: Own authorship.

5.3.1.1 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 5.1, 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:

$$H_t^{min} = \begin{cases} \left\lceil \frac{\sum_{i \in \mathcal{Q}_t} (tat + tf_i)}{tw^B - tw^A} \right\rceil, & t \in \mathcal{HT}\backslash\{r\}; \\ 0, & t = r. \end{cases} \tag{282}$$

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:

$$H_{total}^{min} = \sum_{t \in \mathcal{HT}} H_t^{min}. \tag{283}$$

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:

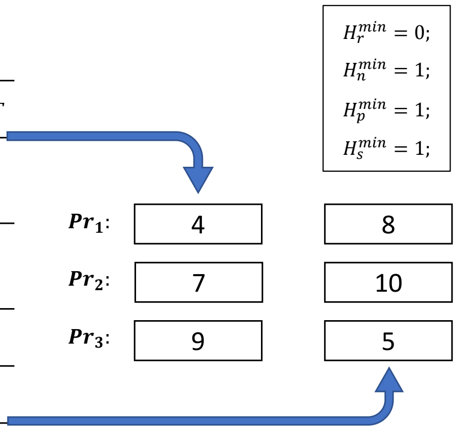
$$\Big(\bigcup_{t \in \mathcal{HT}} \mathcal{L}_t \Big) \cap \Big(\bigcup_{v=1}^{H_{total}^{min}} Pr_v \Big) = \varnothing.$$

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, Figure 33 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 (282), 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 Figure 33, 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.

Figure 33 – Application example of the initial flight sequencing.

Availability: 11:05

i	Type Flight	Type ACFT	r_i	tat	tf_i	$tat + tf_i$	Due Time $r_i + tf_i$	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



Source: Own authorship.

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 5.2. 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 5 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 5 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 (tu_i), safety briefing (sb), 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 ($DT_i - r_i > d_{II}^{\max}$); ($ii$) airport time window ($AT_i > tw^B$); and/or ($iii$) the precedence requirements between the 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, henceforth called *InsertFlights*, is detailed in Algorithm 6. 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 part). 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 $\mathcal{L}_t$

Algorithm 5: *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 $i = 1$ **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` tu_i `and` sb
        if $p_j = p_i \wedge DT_j - DT_i < tu_i$ **then**
          $factTime \leftarrow false$;
          $DT_j \leftarrow DT_j + tu_i - (DT_j - DT_i)$;
        if $DT_j - DT_i < sb$ **then**
          $factTime \leftarrow false$;
          $DT_j \leftarrow DT_j + sb - (DT_j - DT_i)$;
      `// 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 $v = 1$ **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 $0 < D_i \leq d_I^{max} \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)$ **then**
    $B_i^I \leftarrow 1$;
  if $d_I^{max} < D_i \leq d_{II}^{max} \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)$ **then**
    $B_i^{II} \leftarrow 1$;
  if $[D_i > d_{II}^{max} \wedge (i \in \mathcal{I}_0 \cup \mathcal{I}_C)] \vee AT_i > tw^B$ **then**
    $factD \leftarrow false$;
  foreach $j \in PR^k$ **do**
    if $DT_j > DT_i \wedge p_i = p_j \wedge [(i \in \mathcal{I}_0 \cup \mathcal{I}_C \wedge j \in \mathcal{I}_1 \cup \mathcal{I}_2) \vee (i \in \mathcal{I}_C \wedge j \in \mathcal{I} \backslash \mathcal{I}_C)]$ **then**
      $factD \leftarrow false$;

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 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).

Algorithm 6: *InsertFlights*

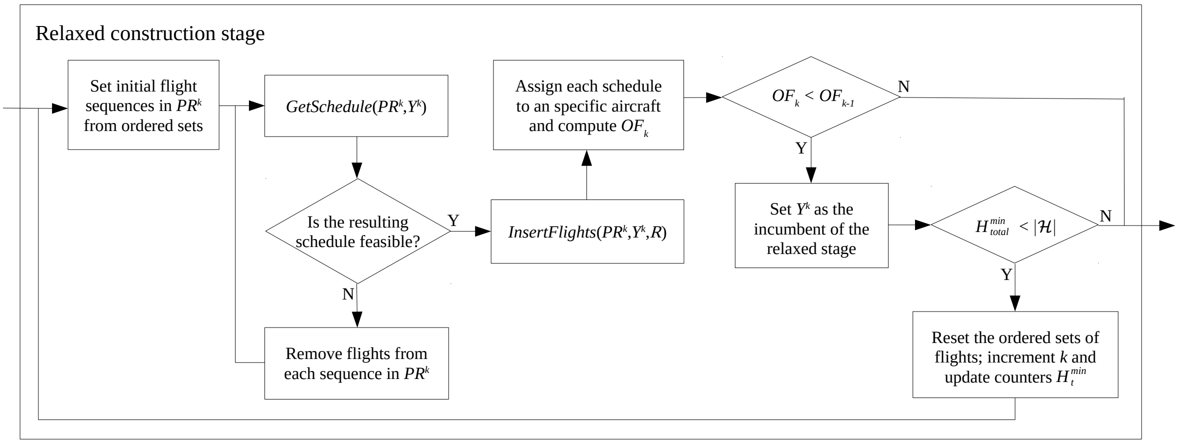
Input: instance parameters, PR^k, current solution Y^k and state ST.
Output: PR^k, Y^k.
1 Let $\mathcal{HP}$ be a set of aircraft, initializing $\mathcal{HP} \leftarrow \{\}$;
2 Let $\mathcal{HA}_j$ be a set of aircraft for flight j;
`// Verify if the heterogeneous fleet constraints should be ignored`
3 **if** $ST = R$ **then**
4 $\quad \mathcal{HA}_j \leftarrow \{1, \ldots, H^{min}_{total}\}, \ \forall\, j \in \mathcal{I}$;
5 **else** // $ST = NR$
6 $\quad \mathcal{HP} \leftarrow \mathcal{HP} \cup \{S^k_j\}, \ \forall\, j \in \mathcal{I}$;
7 $\quad \mathcal{HA}_j \leftarrow \mathcal{HP} \cap \mathcal{H}_j, \ \forall\, j \in \mathcal{I}$;
8 **repeat**
9 $\quad$ **foreach** $t \in \mathcal{HT},\ i \in \mathcal{L}_t,\ v \in \mathcal{HA}_i,\ j \in Pr_v$ **do**
10 $\quad\quad$ Set i as the immediate predecessor of j in Pr_v, updating PR^k and Y^k accordingly;
11 $\quad\quad$ *GetSchedule*(*instance parameters*, PR^k, Y^k);
12 $\quad\quad$ If the resulting schedule is feasible, set:
$\quad\quad \overrightarrow{TD}^{t,v}_{i,j} \leftarrow w_8 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{II}_l + w_9 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{I}_l + w_{11} \sum_{l \in \mathcal{I}} D_l$;
13 $\quad\quad$ Reset i as the immediate successor of j in Pr_v, updating PR^k and Y^k accordingly;
14 $\quad\quad$ *GetSchedule*(*instance parameters*, PR^k, Y^k);
15 $\quad\quad$ If the resulting schedule is feasible, set:
$\quad\quad \overleftarrow{TD}^{t,v}_{i,j} \leftarrow w_8 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{II}_l + w_9 \sum_{l \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{I}_l + w_{11} \sum_{l \in \mathcal{I}} D_l$;
16 $\quad\quad$ Remove i from Pr_v;
17 $\quad (i^*, j^*, t^*, v^*) \leftarrow \underset{i \in \mathcal{L}_t,\ t \in \mathcal{HT},\ v \in \mathcal{HA}_i,\ j \in Pr_v}{argmin} \{\min\{\overrightarrow{TD}^{t,v}_{i,j}, \overleftarrow{TD}^{t,v}_{i,j}\}\}$;
18 $\quad$ **if** $(i^*, j^*, t^*, v^*) \neq \varnothing$ **then**
19 $\quad\quad$ Move i^* from $\mathcal{L}_{t^*}$ to Pr_{v^*};
20 $\quad\quad$ **if** $\overrightarrow{TD}^{t^*,v^*}_{i^*,j^*} \leq \overleftarrow{TD}^{t^*,v^*}_{i^*,j^*}$ **then** $Y^{v^*,k}_{i^*,j^*} \leftarrow 1$;
21 $\quad\quad$ **else** $Y^{v^*,k}_{j^*,i^*} \leftarrow 1$;
22 $\quad\quad$ Update remaining solution Y^k;
23 $\quad$ **else**
24 $\quad\quad$ **foreach** $t \in \mathcal{HT},\ i \in \mathcal{L}_t$ **do**
25 $\quad\quad\quad$ Move i from $\mathcal{L}_t$ to the appropriate recovery set;
26 **until** $\mathcal{L}_t = \varnothing$ *for all* $t \in \mathcal{HT}$;

In the next step of the heuristic, each flight sequence Pr_v is assigned to a specific aircraft h, ignoring 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 $v = 1, \ldots, H^{min}_{total}$. 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 $S^k_i = h$ for each flight i in Pr_v, and redefine $Pr_h := Pr_v$ and $Y^{hk}_{ij} := Y^{vk}_{ij}$. The objective function value OF_k of iteration k is then calculated by:

$$OF_k = w_1.|\mathcal{R}_c| + w_2.|\mathcal{R}_2| + w_3.|\mathcal{R}_1| + w_4.|\mathcal{R}_0| + w_5.H^{\min}_s + w_6.H^{\min}_p + w_7.H^{\min}_n + w_8 \,. \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{II}_i + w_9 \,. \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C} B^{I}_i + \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_C : s_i \neq S^k_i} w_{10} + w_{11}. \sum_{i \in \mathcal{I}} D_i. \quad (284)$$

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 $H^{min}_{total} < |\mathcal{H}|$, a new iteration of the current stage starts by setting $k = k + 1$ and $H^{min}_t = H^{min}_t + 1$, if $H^{min}_t \leq H_t$ (where H_t is the number of aircraft of type t). The choice of which H^{min}_t 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 $H^{min}_{total} = |\mathcal{H}|$, terminating the relaxed construction stage of the heuristic. The main steps of this stage are depicted in the flowchart presented in Figure 34.

Figure 34 – Flowchart of the relaxed construction stage of the proposed heuristic.

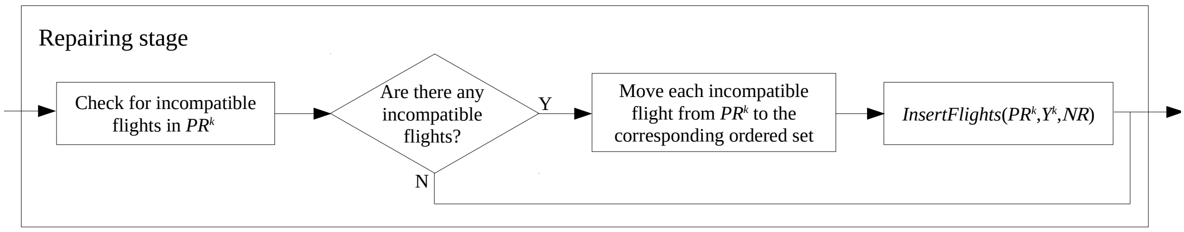


Source: Own authorship.

5.3.1.2 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^k_i 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 stage. 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 Figure 35.

Figure 35 – Flowchart of the repairing stage of the proposed heuristic.



Source: Own authorship.

In summary, considering the overall scope of the construction part (see Figure 32), 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 part ends with a feasible solution of the problem.

5.3.2 Improvement part

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. It tries to insert the flights that could not be allocated in the construction part. This procedure 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 tu_i, sb, 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 used aircraft.

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 part are presented in detail in Appendix C.

5.4 Computational experiments

In this section, we report the results of computational experiments with the network flow and the event-based formulations proposed in Section 5.2 and the heuristic methods developed in Section 5.3. 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 12.10. 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 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 11 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 $w_{11} = 0.001$. They also provided the following parameter values: $sb = 5$ 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].

Table 11 – Main information of the 20 real-life-based instances provided by the company.

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

Source: Own authorship.

In addition to the twenty real-life-based instances described in Table 11 (here called Scenario 0), we also defined other instances grouped into nine other different scenarios to be solved with the heuristic. The first eight of them are based on the larger instances I37C, I38C and I45C of Scenario 0, in which, their parameters were changed randomly for the purpose of investigation. The last scenario applies a sensitivity analysis for the all instances from Scenario 0. The description of each scenario is presented 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 (sb), 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 0, 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.

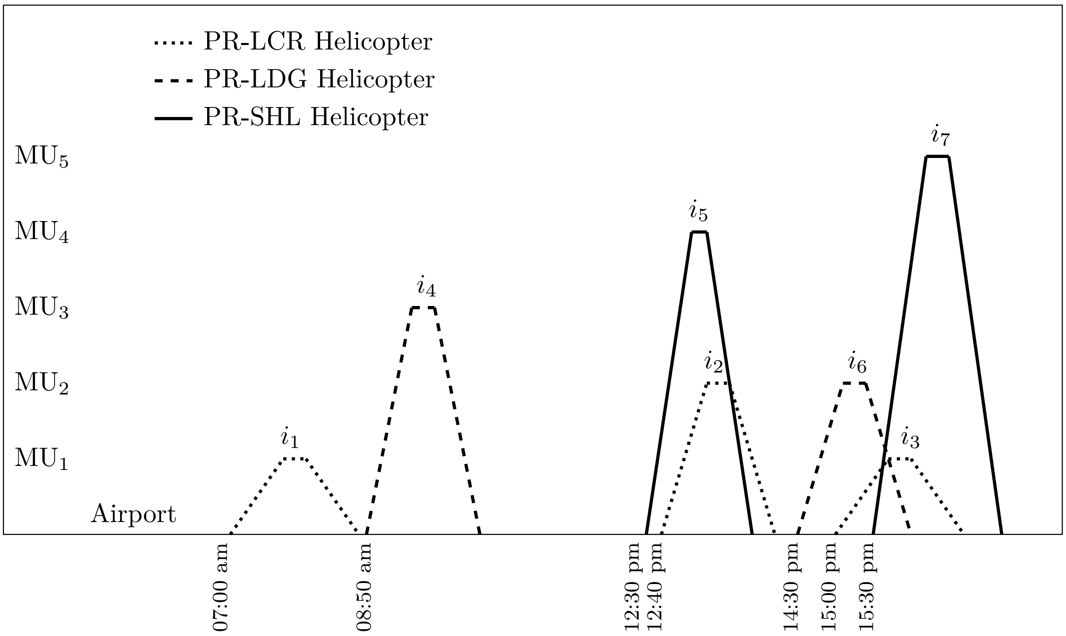
- Scenario 9: it alludes to a sensitivity analysis, in which three types of tests are performed changing the penalties of the objective function. Let us define the following sets: (i) weight families: $\mathcal{F}_1 = \{1, 2, 3, 4\}$ (types of transfers), $\mathcal{F}_2 = \{5, 6, 7\}$ (types of aircraft) 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, $Test_t^1 : w_i = 0, \forall\ i \in \mathcal{F}_t, t = 2, 3$. We note that the omission of family $\mathcal{F}_1$ in $Test_t^1$ 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, $Test_t^2 : w_i = AW_t, \forall\ i \in \mathcal{F}_t, t = 1, 2, 3$. 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: $Test_t^3 : w_i = 0, \forall\ i \in \bigcup_{j=(7-t+1)}^{7} rg_j, t = 1, \ldots, 7$ (we also note the omission of family $\mathcal{F}_1$ in $Test_t^3$).

5.4.1 Toy problem

Firstly, 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 36 and 37. The diagram in Figure 36 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 37. The schedule of these flights uses three helicopters, named as PR-LCR, PR-LDG and PR-SHL. Similarly to Figures 29 and 30, the maritime units MU_1 to MU_5 related to these nine flights are depicted in the vertical axis of Figures

36 and 37, whereas the scheduled/rescheduled departure times of the flights are presented in the horizontal axis. Figure 37 also presents the arrival time of flight i_9 at the airport.

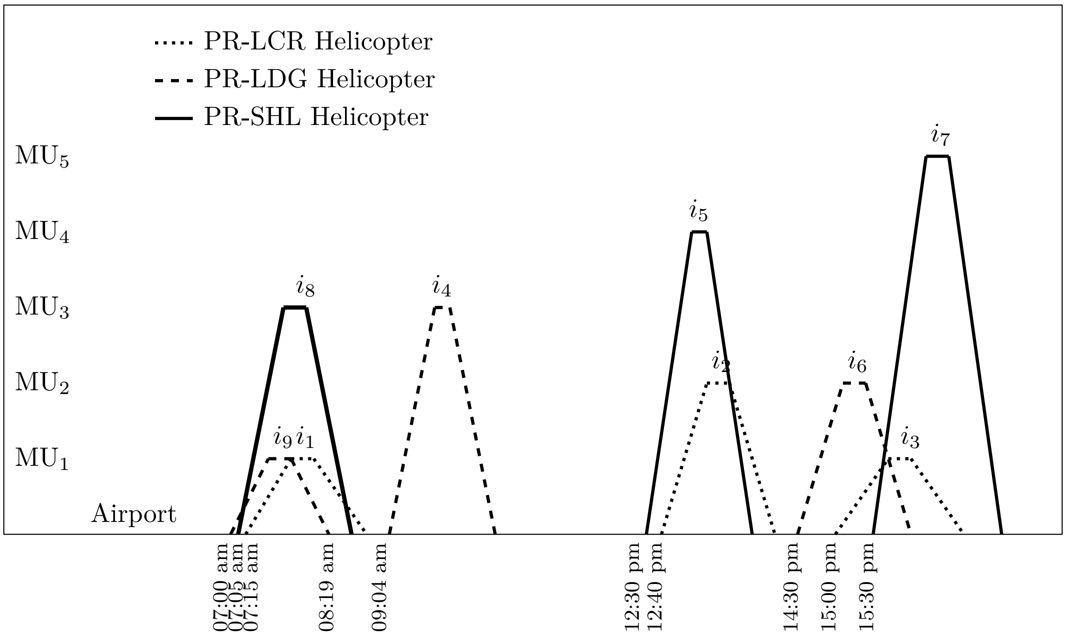
Figure 36 – Time-space diagram illustrating the originally planned schedule of instance I9A without the transferred flights.



Source: Own authorship.

In Figure 36, 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 Figure 37, 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 Figure 37 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 in Figure 37 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. Therefore, 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,

Figure 37 – Time-space diagram illustrating the reschedule of instance I9A with the transferred flights.



Source: Own authorship.

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 approaches proposed here with the manual solutions actually carried out by the company for these problem instances, the benefits of solving the models or applying the heuristic become evident. As an example, instance I10A (see Table 11) 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, conversely, provides a feasible reschedule including all the 10 flights and hence, no flights are transferred to the next day.

5.4.2 Results of the MIP formulations

Tables 12 and 13 present the results obtained using CPLEX, within the time limit of one 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 12 and 13 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, and the letter 'm' if the method stopped due to memory overflow.

The results in Table 12 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 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 enormous 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 13. 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 noticeable 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

Table 12 – 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	0.07	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

Source: Own authorship.

Table 13 – 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.

	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

Source: Own authorship.

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 high for the larger instances, particularly for those related to airport C.

Table 14 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 14 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 the approaches using the network-based model, either alone or in combination with the local branching, in particular for

Table 14 – Best results obtained in the experiments with the two models and the four B&C configurations.

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

Source: Own authorship.

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.

5.4.3 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 5.3 and analyzed in what follows. We first present the results for instances of Scenario 0 and then for instances of Scenarios 1 to 9.

5.4.3.1 Results for the instances of Scenario 0

Table 15 presents the heuristic results for the twenty real-life-based instances (Scenario 0). The meaning of each column is the same as in Table 14. 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 14, 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

Table 15 – Results of the heuristic for the 20 instances of Scenario 0.

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

Source: Own authorship.

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 (one 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 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 11) 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 16 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 14). 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 12 and 13).

Table 16 – Gaps of the heuristic solutions with respect to the best lower and upper bounds obtained by the models.

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			

Source: Own authorship.

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 runtime 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 17 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.

Table 17 – Best results from the computational experiment with the time limit of 24h for instance I45C.

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

Source: Own authorship.

5.4.3.2 Results for the instances of Scenarios 1-8

Table 18 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 15. 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.

Table 18 – Results for the simulated instances of Scenarios 1-8.

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**

Source: Own authorship.

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 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 sb 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.

5.4.3.3 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 11, 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.

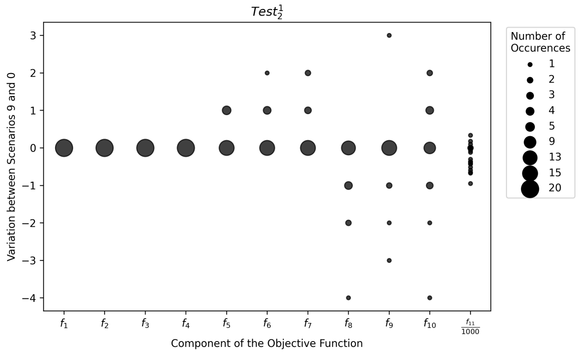
The graphs in Figures 38 depict some of the most interesting results of the first part of the experiments. They present the number of occurrences of each variation level for each objective function term. According to the results of $Test_2^1$ presented in Figure 38a, when canceling the fleet usage penalties, 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 38b), 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 38c) 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 38d), $w_{11}f_{11}$ dominates w_8f_8 and w_9f_9 (since $D >> B^I$ and $D >> B^{II}$) and as expected, the linear delays are reduced.

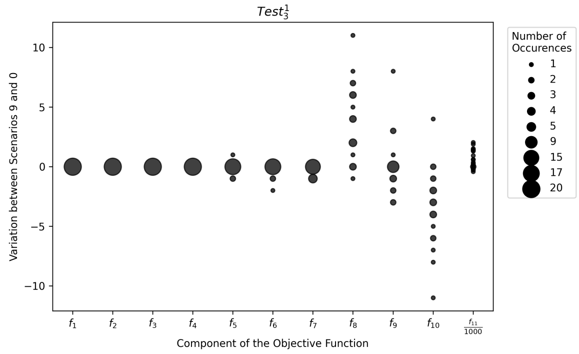
In $Test_1^3$, the linear delay is not penalized and this increases f_{11}, as expected, but does not reduce significantly the other delays. Canceling the weight ranges for $Test_2^3$ (Figure 38e) increases the linear delay and the change of helicopters to reduce delays of types I and II. About $Test_3^3$ (Figure 38f), 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 38g. $Test_5^3$ (Figure 38h) 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 19 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

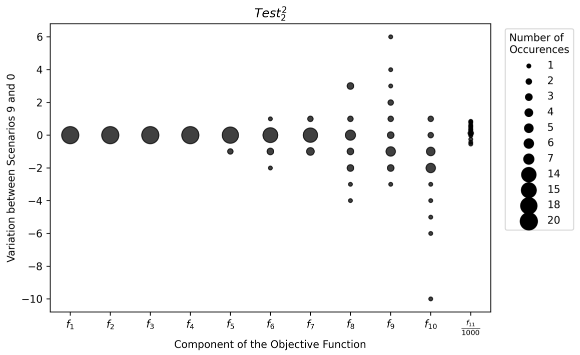
Figure 38 – Analysis of the main result variations between Scenarios 9 and 0.



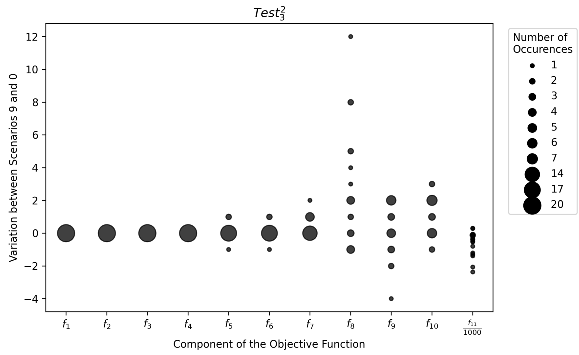
(a)



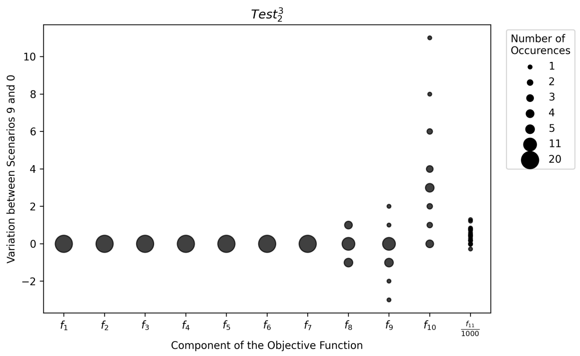
(b)



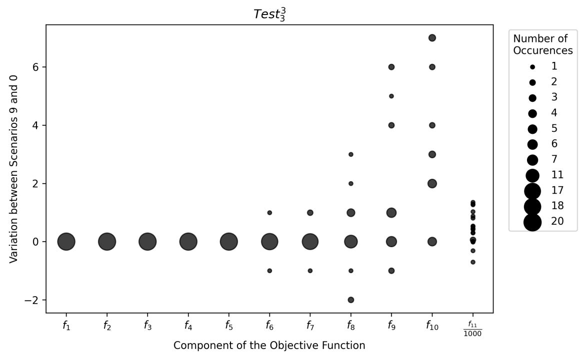
(c)



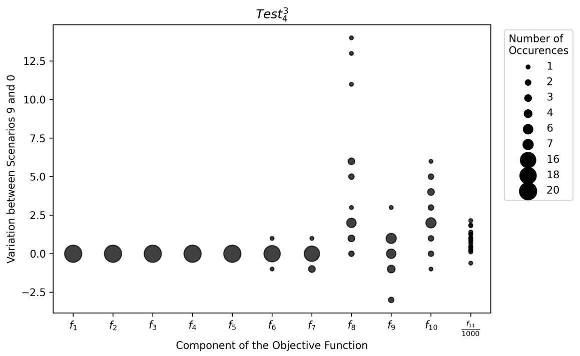
(d)



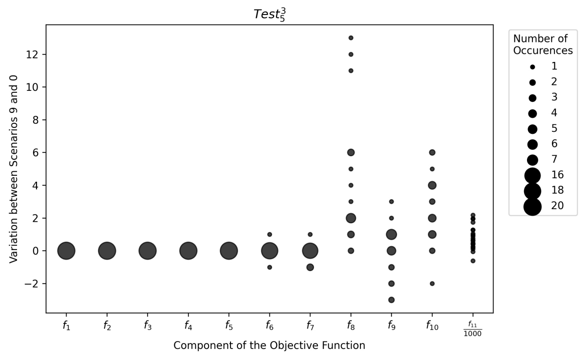
(e)



(f)



(g)



(h)

Source: Own authorship.

Table 19 – Result variations between Scenarios 9 and 0 for instance I45C.

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

Source: Own authorship.

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 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.

Chapter 6

Helicopter recovery in an oil and gas industry considering multiple aerodromes and crew workday

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

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

In such manner, the present research extends the content of Chapter 5 and De La Vega et al. (2022a), where the ARP does not consider a set of aerodromes and possible flight transfers between these aerodromes. The solution approaches of these studies (mathematical models and heuristic method) were developed for the particular case in which there is a single aerodrome, isolated or independent from the others, and all flights must depart from and return to this same aerodrome.

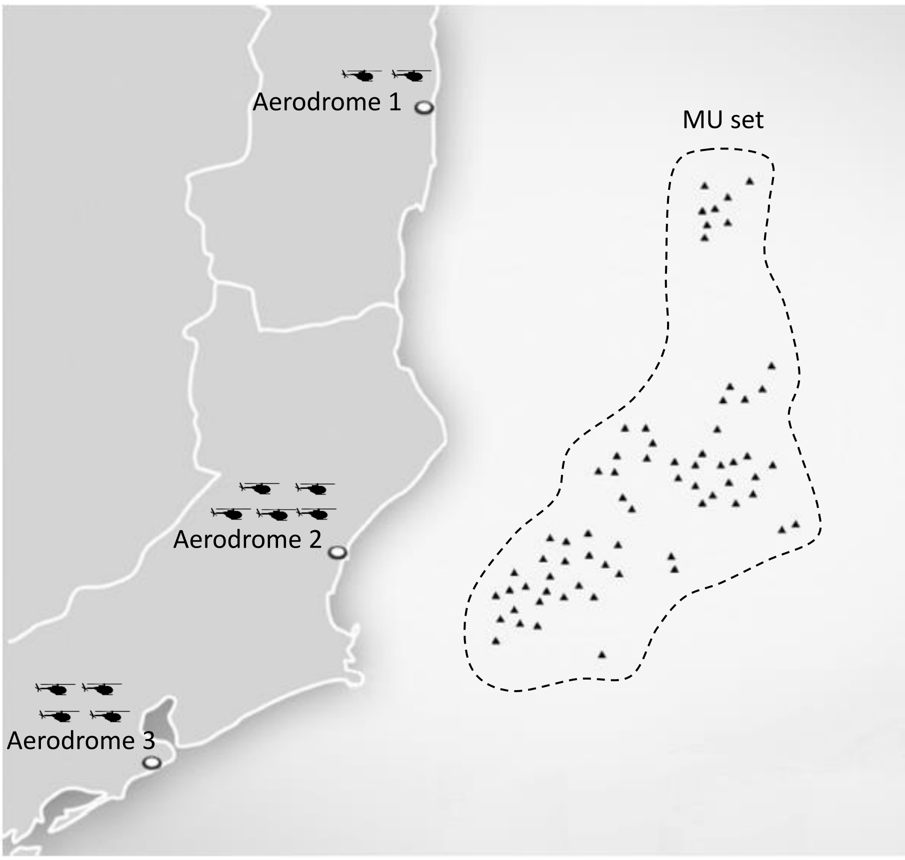
In order to tackle this new problem, two MIP models contemplating multiple aerodromes is proposed, where the first is based on the single aerodrome continuous-time network-flow model, presented in Chapter 5, and the second is inspirited on the single aerodrome discrete-time assignment model, available in De La Vega et al. (2022a). Both closely describe the present company's ARP. Due to its complexity and the usual need of a quick response in practical settings, we also propose a two-phase heuristic (an extension of the previous chapter too), capable of coping with larger problem instances within acceptable computer runtimes. The performance of solving the models with general-purpose optimization software and the heuristic is assessed by means of realistic problem instances with data collected in a case study conducted at the company. Our aim in this work is to contribute to the practice of aerial passenger transportation and improve the company's flight recovery plans, highlighting the potential of proposed approaches. Notwithstanding the specificities of our models, we believe that this study can also contribute to deal with problems of other oil and gas companies that routinely transport work teams to maritime units, such as offshore production platforms, drilling rigs, service units and special vessels. Similar situations arise in oil and gas companies operating in the North sea, Gulf of Mexico, West of Africa and Australia, for example.

The chapter was structured in this manner. Section 6.1 presents the problem description. The MIP formulations and the heuristics are described in Sections 6.2 and 6.3, respectively. Section 6.4 discusses and analyses the computational experiments and results from applying the solution approaches based on the models and the heuristics in realistic problem instances.

6.1 Problem description

As stated before, in this study we consider the recovery of helicopter flights from/(to) aerodromes to/(from) maritime units, motivated by the ARP faced by an oil and gas company. As can be seen in Figure 39, each aerodrome has its own available fleet of helicopters and the helicopters are heterogeneous in terms of travel cruise speed, capacity of passengers, flying range, etc. The company programmers previously schedule several daily flights (timetables) for each aerodrome and for each helicopter at the beginning of the week and, in the absence of unexpected events, these timetables should be followed. Each flight has its scheduled departure time in the day and it is basically a round-trip between an aerodrome and a given maritime unit (i.e., aerodrome - maritime unit - aerodrome). There may be more than one flight departing from the same aerodrome (or from different aerodromes) to the same maritime unit in a day. Each aerodrome helicopter can perform several round-trip flights per day.

Figure 39 – Spatial representation of the studied problem with three aerodromes and a set of maritime units - MU.



Source: Own authorship.

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

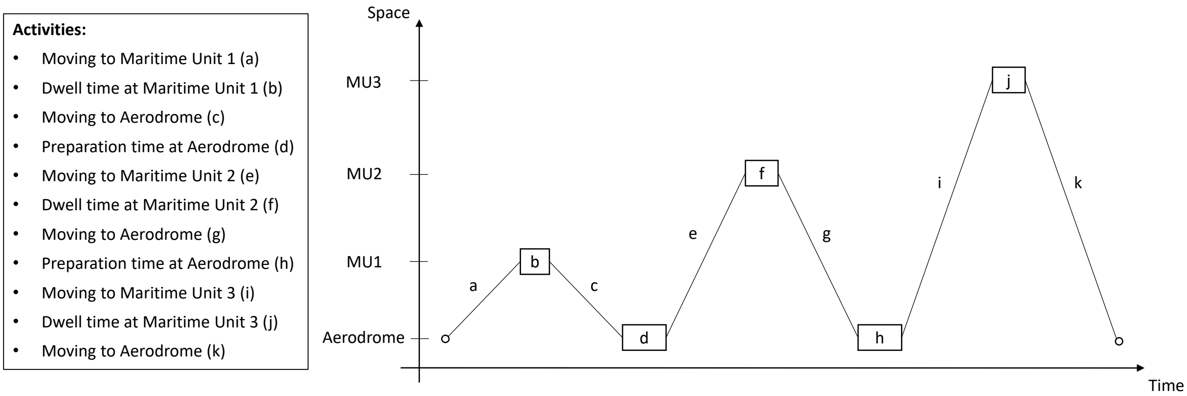
Each aerodrome has a single runway for helicopters taking off and landing, and each maritime unit can bear a single helicopter at a time as it has a single heliport. The ARP consists of determining joint daily flight reschedules for all aerodromes that satisfy operational constraints and recovers all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

There are two main categories of flights: *table* flights, that is, the ones originally scheduled in the timetables of the aerodromes, and the aforementioned *day-transferred* flights, which are the pending unrealized flights transferred from the last days. There are two other classes of high priority and less frequent flights, called *mandatory* and *entourage* flights. Mandatory flights are those that must be performed according to the original timetable, that is, they must not be changed to another aerodrome or transferred to the next day, whereas entourage flights are the ones used to transport managers and other representatives of the company for visits to the maritime units. These are special flight reserved to senior managements, shareholders or even political positions, such as governors and president. In practical terms, the differentiation of entourage flights from the others is justified by their length of time spent in a MU (which is on the order of hours) for holding meetings, visitation, among other activities. Compared to Chapter 5, they no longer block marine units for the rest of the day, which benefits in building more efficient flight rescheduling. As the mandatory flights, the entourage flights must not be changed to another aerodrome, but differently from the mandatory flights, they can be transferred to the next day. We note that the rescheduling of table flights, entourage flights and mandatory flights can only delay the departures, but never anticipate them. Each flight departs from an aerodrome to a destination unit with a group of previously booked passengers and returns to the aerodrome from the unit with another group of previously booked passengers, or the same group in case of entourage flights. Therefore, any flight comprises a set of passengers, a planned departure time, the time required to reach the destination (or to go from the destination back to the aerodrome) and the service time required by the maritime unit to transfer passengers and prepare the aircraft to return. The recovery plan is not allowed to split or join flights, or change the set of passengers previously assigned to the flights in the original timetable. All aerodrome operations must occur during sunlight, forcing an operational time window for each helicopter. A table or entourage flight is transferred to the next day if it either cannot respect the operational time window, or its rescheduled departure time is more than 4 hours from its original departure time.

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

There are also minimum time intervals between consecutive flight departures at the runway of each aerodrome (the departures must be spaced by, at least, 5 minutes) and minimum time intervals for service times of the flights at the maritime units (typically of 15 minutes), as well as minimum time intervals for ground preparation times between consecutive flights of the same helicopter at the aerodrome, called turnaround time - tat (e.g., 45 minutes). A simple representation of the elements of a single aircraft plan is depicted in Figure 40. In this example, three flights departing from the same aerodrome with different destinations are assigned to a single aircraft.

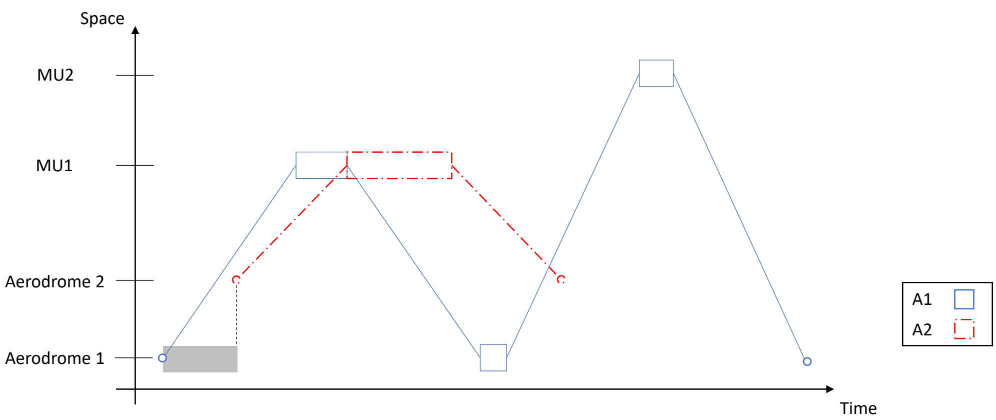
Figure 40 – A representation of a 3-departure flight plan of a single aircraft.



Source: Own authorship.

As an aircraft is not allowed to spend the night in a maritime unit, all helicopters must return to the aerodromes before the end of the day. Moreover, the single heliport and other operational constraints forbid multiple helicopters to land on the maritime unit at the same time. Note that these constrains may affect the departure of different helicopters in different aerodromes, as presented in Figure 41. In this example, aircraft A1 is based on aerodrome 1 and has two scheduled flights: one to maritime unit MU1 and the other to MU2. Aircraft A2 is based on aerodrome 2 and has one flight assigned to MU1. The gray area represents the delay inserted for the departure of aircraft A2 due to flight landing conflicts at maritime unit MU1. Without this delay, A2 would arrive at MU1 when the heliport of MU1 were occupied by aircraft A1.

Figure 41 – A representation of the flight plans of two aircraft A1 and A2 - the gray rectangle represents the delay on the departure of A2 to avoid conflicts at maritime unit MU1.



Source: Own authorship.

During the requirement development phase of this research, the company programmers indicate some guidelines to identify better recovery plans. For example, flights transferred from the last days have higher priorities than table flights of the daily timetable, and helicopters of the normal fleet should be preferred for use rather than helicopters of the pool fleet. These and other guidelines were translated into the following objectives, here depicted in lexicographical order, in accordance with the recommendation of the company's programmers: *i*) minimum number of unrealized entourage flights during the day, *ii*) minimum number of unrealized flights during the day, which were transferred from the last two or more days, *iii*) minimum number of unrealized flights during the day, which were transferred from the last day, *iv*) minimum number of unrealized table flights during the day, *v*) minimum number of table and day-transferred flights changed to a different aerodrome during the same day (mandatory and entourage flights cannot change the aerodrome), *vi*) minimum number of pool helicopters used during the day, *vii*) minimum number of normal helicopters used during the day, *viii*) minimum overall delay of mandatory, entourage and table flights during the day, *ix*) minimum overall delay of the other flights during the day. Formally, a set of performance indicators translates the quality of a daily recovery plan, as detailed in the weighted sum objective function of the models from next section, as well as a formal description of this problem, including the necessary symbols and the mathematical formulations.

6.2 Mathematical formulations

Before presenting the two optimization models to represent this ARP, we introduce the following notation that is common between them. Let I, H, P and K be the total number of flights, helicopters, maritime units and aerodromes in the problem instance, respectively. Regarding the parameters, we have:

Parameters:

- r_i: scheduled departure time of flight i;
- s_{ih}: travel time of flight i from the aerodrome of helicopter h to the destination maritime unit of this flight;
- tu_i: dwell time of flight i at its destination maritime unit;
- $tf_{ih} = 2.s_{ih} + tu_i$: duration of flight i using helicopter h, including the dwell time at the destination maritime unit of this flight;
- tat: minimum time interval between a landing and a takeoff of the same helicopter at an aerodrome (the turnaround time);
- sb: minimum time interval between consecutive takeoffs of helicopters at the same aerodrome (the safety briefing);

- $d^{\max}$: maximum allowed delay for table, entourage and mandatory flights;
- tr_{kl}: land transfer time (by road transportation) of flight passengers between aerodromes k and l;
- $t^{\max}$: maximum travel time (by road transportation) used to allow local-transfer of flights between pairs of different aerodromes;
- wd_h: workday of helicopter h (crew requirement);
- $[tw_k^A, tw_k^B]$: time window of aerodrome k;
- u_i: destination maritime unit of flight i;
- $\hat{a}_h$: aerodrome of helicopter h;
- $\check{a}_i$: aerodrome base of flight i (i.e., the aerodrome where flight i was scheduled in the original timetable);
- pr_{ij}: $= 1$, if the departure of flight i from any aerodrome must be scheduled before the departure of flight j from any aerodrome, and 0, otherwise;
- c_{ih}: $= 1$, if helicopter h can be assigned (i.e., it is compatible in terms of capacity, flying range, etc.) to flight i, and 0, otherwise;
- $n_i = \begin{cases} 0, & \text{if flight } i \text{ is a table flight;} \\ 1, & \text{if flight } i \text{ is a transferred flight delayed by 1 day;} \\ 2, & \text{if flight } i \text{ is a transferred flight delayed by 2 or more days;} \\ 3, & \text{if flight } i \text{ is an entourage flight;} \\ 4, & \text{if flight } i \text{ is a mandatory flight.} \end{cases}$

It is noteworthy to mention that some previously defined parameters are made available by the company in different units of time. However, before running the models, we convert all time parameters to minutes.

6.2.1 Continuous-time model

This continuous-time model is derived from the extension of the network-flow formulation proposed in Chapter 5. We created two dummy flights 0 and $I + 1$ and impose that any sequence of flights from a helicopter starts with dummy flight 0 and ends with dummy flight $I + 1$. As each aerodrome has its own fleet of helicopters that cannot be shared/transferred among them, the following decision variables are well-defined without the explicit use of an aerodrome index k (i.e., this information can be obtained from the helicopter index h and parameter $\hat{a}_h$):

- X_{ijh}: 1, if and only if flight i is performed immediately before flight j by helicopter h;
- Y_{ih}: 1, if and only if helicopter h is assigned to flight i;
- Z_{ij}: 1, if and only if the departure of flight i from any aerodrome is before the departure of flight j from any aerodrome;
- V_h: 1, if and only if helicopter h is used;
- $DT_i \geq 0$: departure time of flight i;
- $AT_i \geq 0$: arrival time of flight i;
- $D_i \geq 0$: delay of flight i.

The continuous-time model for the present ARP is formulated as follows:

Objective function. As the present ARP involves different objectives with different priorities to be optimized, the objective function (285) uses a weighted sum method to minimize the sum of the penalties associated to the following terms: (f_1) flight transfers for the next day (except for mandatory flights); (f_2) flight transfers to aerodromes different from the flight base aerodrome; (f_3) use of helicopters; (f_4) flight delays. The vectors w^1, w^2, w^3 and w^4 establish the corresponding weights of each these terms and their relative importance in the present case study is such that: $w^1 > w^2 > w^3 > w^4$, as indicated by the company's programmers. The penalty of $w_i^1 \in w^1$ depends on the class of flight i (table, transferred from the last day, transferred from the last two or more days, entourage), the penalty of $w_{ik}^2 \in w^2$ depends on the distance between the original base aerodrome of flight i and aerodrome k, the penalty of $w_h^3 \in w^3$ depends on the fleet and the model of helicopter h (normal, pool), and the penalty of $w_i^4 \in w^4$ depends on the class of flight i (table, transferred from the last day, transferred from the last two or more days, entourage, mandatory). Each of these penalty values were carefully defined by the company's programmers in order to reflect their relative priorities.

$$\min \quad f = \sum_{s=1}^{4} f_s;$$

where

$$f_1 = \sum_{\substack{i=1:\\ n_i \neq 4}}^{I} w_i^1 . \left(1 - \sum_{h=1}^{H} Y_{ih}\right); \quad f_2 = \sum_{i=1}^{I} \sum_{\substack{k=1:\\ k \neq \check{a}_i}}^{K} \sum_{\substack{h=1:\\ k=\hat{a}_h}}^{H} w_{ik}^2 . Y_{ih}; \tag{285}$$

$$f_3 = \sum_{h=1}^{H} w_h^3 . V_h; \quad f_4 = \sum_{i=1}^{I} w_i^4 . D_i.$$

Flow constraints. Constraints (286)-(288) define the flow constraints. If flight i is assigned to helicopter h, i.e., if $Y_{ih} = 1$, in a given solution, then constraints (286) and

(287) ensure that there is an immediate predecessor flight (which can be the dummy flight 0) and another immediate successor flight (which can be the dummy flight $I+1$) in the helicopter h schedule, respectively. Constraints (288) ensure the outflow and inflow of dummy flights 0 and $I+1$ if helicopter h is used in the solution (i.e., if $V_h = 1$).

$$\sum_{\substack{i=0: \\ i \neq j}}^{I} X_{ijh} = Y_{jh}; \ \forall \ j = 1, \ldots, I; h = 1, \ldots, H; \tag{286}$$

$$\sum_{\substack{j=1: \\ j \neq i}}^{I+1} X_{ijh} = Y_{ih}; \ \forall \ i = 1, \ldots, I; h = 1, \ldots, H; \tag{287}$$

$$\sum_{i=1}^{I} X_{0ih} = \sum_{j=1}^{I} X_{j(I+1)h} = V_h; \ \forall \ h = 1, \ldots, H. \tag{288}$$

Assignment constraints. The assignment constraints are defined by constraints (289)-(292). Constraints (289) are related to the scheduling or not of the day-transferred and table flights, either at their base aerodrome or at a different one. Constraints (290) and (291) ensure that the entourage and mandatory flights, if scheduled in a solution, must depart from their base aerodromes, respectively. Note that the equality of constraints (291) forces the scheduling of mandatory flights in the solution – this is why there is no need for penalties associated with their transfer to the next day in the objective function. Finally, constraints (292) are used to avoid assigning flights to a helicopter h if this helicopter is not used in the solution (i.e., if $V_h = 0$).

$$\sum_{h=1}^{H} Y_{ih} \leq 1; \ \forall \ i = 1, \ldots, I \mid n_i = 0, 1, 2; \tag{289}$$

$$\sum_{\substack{h=1: \\ \hat{a}_h = \check{a}_i}}^{H} Y_{ih} \leq 1; \ \forall \ i = 1, \ldots, I \mid n_i = 3; \tag{290}$$

$$\sum_{\substack{h=1: \\ \hat{a}_h = \check{a}_i}}^{H} Y_{ih} = 1; \ \forall \ i = 1, \ldots, I \mid n_i = 4; \tag{291}$$

$$Y_{ih} \leq V_h; \ \forall \ i = 1, \ldots, I; h = 1, \ldots, H. \tag{292}$$

Helicopter synchronization constraints. The synchronization of the flight departures made by the same helicopter is guaranteed by the set of constraints (293) and (294), which also act as sub-tour elimination constraints for the flow variables X_{ijh}. Constraints (293) impose a minimum time of tat minutes between the arrival of flight i and the departure of flight j if both flights are performed consecutively by the same helicopter. This time is mainly associated with the helicopter inspection. Constraints (294) determine the arrival times of the flights in accordance with their departure times and assigned helicopters. Note that these constraints and constraints (295)-(296) (described later) ensure that the continuous variables DT_i, AT_i, and D_i assume the value 0 if flight i is not scheduled in a solution, i.e., if $\sum_{h=1}^{H} Y_{ih} = 0$. Therefore, the equality sign of constraints (294) is without loss of generality. $BigM$ is used in the model as a sufficiently large number (in our implementation, we set $BigM = \max_{k=1,\ldots,K}\{tw_k^B\}$).

$$DT_j \geq AT_i + tat.X_{ijh} - BigM.\left(1 - X_{ijh}\right); \quad \forall\ i, j = 1, \ldots, I \mid i \neq j; h = 1, \ldots, H; \tag{293}$$

$$AT_i = DT_i + \sum_{h=1}^{H} tf_{ih}.Y_{ih};\ \ \forall\ \ i = 1, \ldots, I. \tag{294}$$

Scheduled departure time and maximum delay constraints. Constraints (295) impose the minimum departure times for flights and also determine the flight delays. Note that these constraints indicate that the flight departures can be delayed, but not advanced. Constraints (296) impose a maximum delay of $d^{\max}$ hours on the table, entourage and mandatory flights.

$$r_i.\sum_{h=1}^{H} Y_{ih} \leq DT_i \leq r_i.\sum_{h=1}^{H} Y_{ih} + D_i;\ \ \forall\ \ i = 1, \ldots, I; \tag{295}$$

$$D_i \leq d^{\max}.\sum_{h=1}^{H} Y_{ih};\ \ \forall\ \ i = 1, \ldots, I \mid n_i = 0, 3, 4. \tag{296}$$

Time window constraints. The imposition of aerodrome time windows is given by constraints (297) and (298). These constraints require that the takeoff and landing of the flights scheduled in the solution, respectively, satisfy the minimum and maximum operating timetables of the aerodrome associated with the helicopter that performs them.

$$DT_i \geq \sum_{h=1}^{H} tw^A_{\hat{a}_h} Y_{ih};\ \ \forall\ \ i = 1, \ldots, I; \tag{297}$$

$$AT_i \leq \sum_{h=1}^{H} tw^B_{\hat{a}_h} Y_{ih};\ \ \forall\ \ i = 1, \ldots, I. \tag{298}$$

Helicopter workday constraints. The workday of the helicopters is imposed by constraints (299). For a given helicopter h, these constraints determine the elapsed time between its first takeoff and its last landing and also ensure that this time does not exceed its workday of wd_h hours.

$$AT_j - DT_i \leq wd_h + BigM.\left(2 - X_{0ih} - X_{j(I+1)h}\right); \quad \forall\ i, j = 1, \ldots, I \mid i \neq j; h = 1, \ldots, H. \tag{299}$$

Precedence constraints. Constraints (300)-(302) activate the variables Z_{ij} used to enforce the synchronization of the departures from the aerodrome of the flights, as well as their arrivals at the maritime units. Constraints (300) ensure at most an order of precedence between flights i and j, regardless of the aerodromes where they are scheduled in a solution. Constraints (301) guarantee that if flights i and j are scheduled in a solution, then there must be at least one order of precedence between them, i.e., the departure of flight i precedes the departure of flight j (in this case $Z_{ij} = 1$) or the departure of flight j precedes the departure of i ($Z_{ji} = 1$). Constraints (302) are complementary to constraints (301) and ensure the scheduling of flights i and j if there is an order of precedence between them, i.e., if $Z_{ij} = 1$ or $Z_{ji} = 1$. It is important to note that constraints (300)-(302) only guarantee the existence of one order of precedence between flights i and j if they are scheduled in the solution, but they do not define the order. Constraints (303) and (304) (described below) are responsible for defining this order, which is the one whose departure times from the aerodrome of flights i and j incur the least penalty of the total delay.

$$Z_{ij} + Z_{ji} \leq 1; \;\; \forall \; i, j = 1, \ldots, I \mid i \neq j;. \tag{300}$$

$$Z_{ij} + Z_{ji} \geq \sum_{h=1}^{H} Y_{ih} + \sum_{h=1}^{H} Y_{jh} - 1; \;\; \forall \; i, j = 1, \ldots, I \mid i \neq j; \tag{301}$$

$$2. (Z_{ij} + Z_{ji}) \leq \sum_{h=1}^{H} Y_{ih} + \sum_{h=1}^{H} Y_{jh}; \;\; \forall \; i, j = 1, \ldots, I \mid i \neq j. \tag{302}$$

Aerodrome synchronization constraints. The synchronization of the departure times of scheduled flights at the same aerodrome is guaranteed by constraints (303). In these constraints, if $Z_{ij} = 1$ and both flights i and j are scheduled at the same aerodrome k, then their departure times must be at least sb minutes apart.

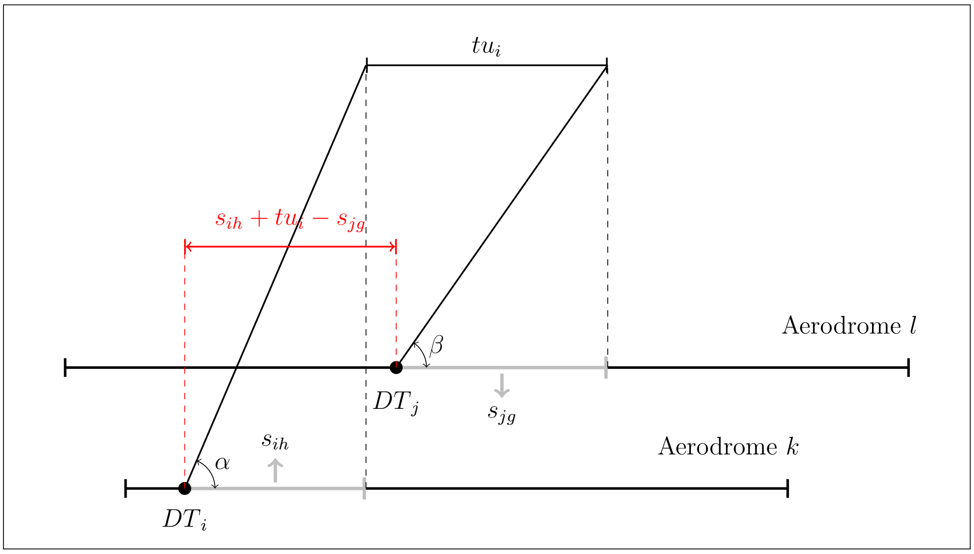
$$DT_j - DT_i \geq sb.Z_{ij} - BigM. \left(3 - Z_{ij} - \sum_{\substack{h=1: \\ \hat{a}_h = k}}^{H} Y_{ih} - \sum_{\substack{h=1: \\ \hat{a}_h = k}}^{H} Y_{jh} \right); \quad \forall \; k = 1, \ldots, K; i, j = 1, \ldots, I \mid i \neq j. \tag{303}$$

Unit maritime synchronization constraints. Constraints (304) ensure the synchronization of the arrival times of the flights at a maritime unit p. In Figure 42, we illustrate how the imposition of the time interval $(s_{ih} + tu_i - s_{jg})$ ensure this synchronization. In this figure, we assume that flight i is assigned to helicopter h at aerodrome k, while flight j is assigned to helicopter g at aerodrome l (we note that helicopters h and g can be associated to the same aerodrome; in this case aerodrome k coincides with aerodrome l). Moreover, we also assume that the departure of flight i from aerodrome k occurs before the departure of flight j from aerodrome l. Observe that the $(s_{ih} + tu_i - s_{jg})$ interval in Figure 42 prevents flight j from landing on the maritime unit while its heliport is occupied by flight i. It is worthy of mention that the synchronization of the arrival times of the flights in the maritime unit can be done only using the departure times of the flights from their aerodromes because the flight travel times are considered deterministic.

$$DT_j - DT_i \geq (s_{ih} + tu_i - s_{jg}) \, Z_{ij} - BigM. (3 - Z_{ij} - Y_{ih} - Y_{jg}); \quad \forall \; i, j = 1, \ldots, I; \mid i \neq j \wedge u_i = u_j; h, g = 1, \ldots, H \mid h \neq g. \tag{304}$$

Other practical constraints: Constraints (305)-(308) are necessary to consider other practical characteristics of the company problem. The set of constraints (305) and (306)

Figure 42 – Illustration of the synchronization of flights at a maritime unit.



Source: Own authorship.

ensures that flight i (if scheduled in a solution) is not carried out by a helicopter incompatible with it. Constraints (307) impose an order of precedence between flights i and j. Thus, in a given solution and regardless of the aerodromes where these flights are scheduled, the departure of flight j will never occur before the departure of flight i if $pr_{ij} = 1$. The set of constraints (308) prevents local-transfers between different aerodromes for entourage and mandatory flights or if the travel time by bus between these aerodromes exceed $t^{\max}$ units of time for the other type of flights

$$X_{ijh} = 0; \;\; \forall \; i, j = 1, \dots, I \mid i \neq j; h = 1, \dots, H \mid c_{ih} + c_{jh} < 2; \tag{305}$$
$$Y_{ih} \leq c_{ih}; \;\; \forall \; i = 1, \dots, I; h = 1, \dots, H; \tag{306}$$
$$Z_{ji} = 0; \;\; \forall \; i, j = 1, \dots, I \mid i \neq j \wedge pr_{ij} = 1; \tag{307}$$
$$Y_{ih} = 0; \;\; \forall \; i = 1, \dots, I; h = 1, \dots, H \mid \breve{a}_i \neq \hat{a}_h \wedge (tr_{\breve{a}_i, \hat{a}_h} > t^{\max} \vee n_i = 3, 4). \tag{308}$$

Domain of the decision variables. Constraints (309)-(313) indicate the type and domain of the decision variables. From these constraints, we observe that variables X_{ijh}, Y_{ih}, Z_{ij} and V_h are binaries, while variables DT_i, AT_i and D_i are non-negative continuous.

$$X_{ijh} \in \{0, 1\}; \;\; \forall \; i = 0, \dots, I; j = 1, \dots, I+1 \mid i \neq j; h = 1, \dots, H; \tag{309}$$
$$Y_{ih} \in \{0, 1\}; \;\; \forall \; i = 1, \dots, I; h = 1, \dots, H; \tag{310}$$
$$Z_{ij} \in \{0, 1\}; \;\; \forall \; i, j = 1, \dots, I \mid i \neq j; \tag{311}$$
$$V_h \in \{0, 1\}; \;\; \forall \; h = 1, \dots, H; \tag{312}$$
$$DT_i \geq 0, AT_i \geq 0, D_i \geq 0; \;\; \forall \; i = 1, \dots, I. \tag{313}$$

6.2.2 Discrete-time model

An alternative formulation for this ARP is to treat the time as a discrete measure, which refers to assigning a temporal index (for example, t) into the decision variables. Evidently, the ideal total number of periods (T^*) to be used for this modeling corresponds to the number of minutes contained within the largest time window of the aerodromes considered in a given instance, which makes the discretization equivalent to the schedules generated by the continuous-time model. However, representing each index t as one minute may be impractical, since T^* typically have 660 periods/minutes (in general, the time windows are from 7:00 am to 6:00 pm), thus drastically impacting memory consumption and computing time. A way to overcome this is by reducing T^*, making each t represent a time greater than one minute. We chose to express t as 5 minutes, once the total number of periods dropped to $660/5 = 132$, the solution quality is not much affected (assessed by computational tests), and it already respects the requirement of placing the safety briefing in the takeoff instants.

This subsection extends the recent discrete-time formulation elaborated in De La Vega et al. (2022a), who also proposed leaving the time as a multiple of 5 minutes. Unlike this work, the authors did not consider the presence of multiple aerodromes and the possibility of transferring the flight locations.

Next, we introduce the parameters converted by the discretization and the decision variables:

Parameters:

- $factor = sb$: number to be used as multiple of the time-related data (in this case, 5 minutes);
- $\overline{r}_i = \lceil r_i/factor \rceil - \min_{k=1,\dots,K}\{\lceil tw_k^A/factor \rceil\} + 1$: scheduled departure of flight i, discretized in 5-minute periods;
- $\overline{s}_{ih} = \lceil s_{ih}/factor \rceil$: travel time of flight i from helicopter h to its destination maritime unit, discretized in 5-minute periods;
- $\overline{tu}_i = \lceil tu_i/factor \rceil$: dwell time of flight i at its destination maritime unit, discretized in 5-minute periods;
- $\overline{tf}_{ih} = \lceil tf_{ih}/factor \rceil$: duration of flight i using helicopter h, discretized in 5-minute periods;
- $\overline{tat} = \lceil tat/factor \rceil$: turnaround time discretized in 5-minute periods;
- $\overline{sb} = \lceil sb/factor \rceil$: safety briefing discretized in 5-minute periods;
- $\overline{d^{\max}} = \lfloor d^{\max}/factor \rfloor$: maximum allowed delay for pre-scheduled flights, discretized in 5-minute periods;
- $\overline{wd}_h = \lfloor wd_h/factor \rfloor$: workday of crew on board helicopter h, discretized in 5-minute periods;

- $T = \max_{k=1,\ldots,K}\{\lfloor tw_k^B/factor\rfloor\} - \min_{k=1,\ldots,K}\{\lceil tw_k^A/factor\rceil\} + \overline{tat}$: planning horizon (added with $\overline{tat}$, which will be discussed later) discretized in 5-minute periods.

Decision variables:

- X_{ith}: 1, if flight i is started by helicopter h in period t; 0, otherwise;
- Y_{th}: 1, if helicopter h is performing a flight in period t; 0, otherwise;
- Z_{tk}: 1, if a takeoff from aerodrome k occurs in period t; 0, otherwise;
- Q_{tp}: 1, if a landing to maritime unit p occurs in period t; 0, otherwise;
- V_h: 1, if helicopter h is used; 0, otherwise;
- out_i: 1, if flight i is not scheduled on the current day; 0, otherwise (except for mandatory flights).

Even with the simplification of setting time-related data in multiples of 5 minutes, the assembly of discrete-time model can still impact the computing time for larger realistic instances. Consequently, we created the parameters below in order to improve the formulation by reducing the ranges of variables and constraints, hence decreasing the solver's runtime.

Enhancements:

- $tFirst_k^Z = \lceil tw_k^A/factor\rceil - \min_{l=1,\ldots,K}\{\lceil tw_l^A/factor\rceil\} + 1$;
- $tLast_k^Z = T - (\max_{l=1,\ldots,K}\{\lceil tw_l^B/factor\rceil\} - \lceil tw_k^B/factor\rceil)$;
- $tFirst_{ih}^X = \begin{cases} \overline{r}_i, \text{if } r_i > tw_{\hat{a}_h}^A; \\ tFirst_{\hat{a}_h}^Z, \text{otherwise.} \end{cases}$
- $tLast_{ih}^X = \begin{cases} \min(tFirst_{ih}^X + \overline{d^{\max}}, tLast_{\hat{a}_h}^Z - \overline{tf}_{ih} - \overline{tat} + 1), n_i = 0, 3, 4; \\ tLast_{\hat{a}_h}^Z - \overline{tf}_{ih} - \overline{tat} + 1, n_i = 1, 2. \end{cases}$
- $tLast_i^{X2} = \min_{h=1,\ldots,H:c_{ih}=1}\{tLast_{ih}^X\}$;
- $tFirst_h^Y = \min_{i=1,\ldots,I:c_{ih}=1}\{tFirst_{ih}^X\}$;
- $tLast_h^Y = \max_{i=1,\ldots,I:c_{ih}=1}\{tLast_{ih}^X + \overline{tf}_{ih}\} + \overline{tat} - 1$;
- $tFirst_p^Q = \min_{i=1,\ldots,I;h=1,\ldots,H:u_i=p\wedge c_{ih}=1}\{tFirst_{ih}^X + \overline{s}_{ih}\}$;
- $tLast_p^Q = \max_{i=1,\ldots,I;h=1,\ldots,H:u_i=p\wedge c_{ih}=1}\{tLast_{ih}^X + \overline{s}_{ih} + \overline{tu}_i\}$;
- $leftover_i = \lceil r_i/factor\rceil.factor - r_i$.

Thereby, the enhanced discrete-time model for the problem is proposed in that way.

Objective function. Similar the optimization of (285), (314) minimizes the weighted sum of unassigned flights transferred to the next day (f_1), the weighted sum of flights transferred to other aerodromes (f_2), the helicopter utilization (f_3), and the total delay of the flights (f_4). Some slight differences are in the use of variable out_i to quantify the flights transferred from the current day in f_1, the presence of variable X_{ith} to compare the designated aerodromes with the pre-defined ones in f_3, and the usage of index t and parameter $leftover_i$ to quantify the delay obtained in f_4. An addendum to $leftover_i$ is that it computes the delay obtained by rounding r_i in the discretization to correct the existing lag between $t = 1$ and $\overline{r}_i$.

$$\min \quad f = \sum_{s=1}^{4} f_s;$$

where

$$f_1 = \sum_{\substack{i=1: \\ n_i \neq 4}}^{I} w_i^1.out_i; \quad f_2 = \sum_{i=1}^{I} \sum_{\substack{k=1: \\ k \neq \breve{a}_i}}^{K} \sum_{\substack{h=1: \\ c_{ih}=1 \\ k=\hat{a}_h}}^{H} w_{ik}^2. \left(\sum_{t=tFirst_{ih}^X}^{tLast_{ih}^X} X_{ith} \right); \tag{314}$$

$$f_3 = \sum_{h=1}^{H} w_h^3.V_h; \quad f_4 = \sum_{i=1}^{I} \sum_{\substack{h=1: \\ c_{ih}=1}}^{H} \sum_{t=tFirst_{ih}^X}^{tLast_{ih}^X} w_i^4. \left[(t - \overline{r}_i) .factor + leftover_i \right] .X_{ith}.$$

Assignment constraints. Constraints (315)-(317) are responsible for associating a flight to an helicopter and period time. Specifically, (315) ensure that each table or day-transferred flight i is performed once at most, independent from the designated aerodrome, (316) guarantee that entourage flights can be assigned only their base aerodromes, while (317) obliges that all mandatory flights are included in the original planning (only permitting delay).

$$\sum_{\substack{h=1: \\ c_{ih}=1}}^{H} \sum_{t=tFirst_{ih}^X}^{tLast_{ih}^X} X_{ith} + out_i = 1; \ \forall \ i = 1, \dots, I \mid n_i = 0, 1, 2; \tag{315}$$

$$\sum_{\substack{h=1: \\ \hat{a}_h=\breve{a}_i \\ c_{ih}=1}}^{H} \sum_{t=tFirst_{ih}^X}^{tLast_{ih}^X} X_{ith} + out_i = 1; \ \forall \ i = 1, \dots, I \mid n_i = 3; \tag{316}$$

$$\sum_{\substack{h=1: \\ \hat{a}_h=\breve{a}_i \\ c_{ih}=1}}^{H} \sum_{t=tFirst_{ih}^X}^{tLast_{ih}^X} X_{ith} = 1; \ \forall \ i = 1, \dots, I \mid n_i = 4. \tag{317}$$

Logical occupancy constraints. They express the level of occupancy/usage for the resources, aerodromes, aircraft and maritime units, in the time horizon. Constraints (318) cause flight i to be allocated (if applicable) to at most one aerodrome k and time period t by activating binary variable Z_{tk}, hence respecting safety briefing sb (given the $factor = 5$, as mentioned above). Constraints (319) allow the duration of flight $\overline{tf}_{ih}$ and the turnaround time $\overline{tat}$ to be fulfilled. Given the activation of X_{ith} in period t, these constraints make binary variable Y_{th} to be activated from period t to $t + \overline{tf}_{ih} + \overline{tat}$ for helicopter h (i.e., $Y_{th} = 1, \ldots, Y_{t+\overline{tf}_{ih}+\overline{tat},h} = 1$). This eliminates overlap between the landing and takeoff for a same aircraft from its aerodrome. Finally, constraints (320) impose dwell time $\overline{tu}_i$ for each flight lading at its destination maritime unit (i.e., landing period $t + \overline{s}_{ih}$ at MU u_i). Note that (320) has the similar operating principle as (319), although differing in relation to the sign "$\geq$", since the "for all" with index h is necessary, given the presence of $\overline{s}_{ih}$ in the variable $Q_{t+\overline{s}_{ih},u_i}$. It is also important to comment on the placement of variables Z_{tk}, Y_{th} and Q_{tp} in constraints (318), (319) and (320) instead of simply using the summations with "≤ 1". We observed that the solver's B&C performance improved considerably by enabling branching on the bounds of these constraints.

$$Z_{tk} = \sum_{i=1}^{I} \sum_{\substack{h=1: \\ c_{ih}=1 \\ k=\hat{a}_h}}^{H} X_{ith}; \;\; \forall \;\; k = 1, \ldots, K; t = tFirst_k^Z, \ldots, tLast_k^Z; \tag{318}$$

$$\begin{aligned} Y_{th} = \sum_{\substack{i=1: \\ c_{ih}=1}}^{I} \; \sum_{t'=t,t-1,\ldots,\max\left\{tFirst_{ih}^X, t-\overline{tf}_{ih}-\overline{tat}+1\right\}} X_{it'h}; \\ \forall \; h = 1, \ldots, H; t = tFirst_h^Y, \ldots, tLast_h^Y; \end{aligned} \tag{319}$$

$$\begin{aligned} Q_{t+\overline{s}_{ih},u_i} \geq \sum_{t'=t,t-1,\ldots,\max\left\{tFirst_{ih}^X, t-\overline{tu}_i+1\right\}} X_{it'h}; \\ \forall \; i = 1 \ldots, I; h = 1 \ldots, H \mid c_{ih} = 1; t = tFirst_{ih}^X, \ldots, tLast_{ih}^X. \end{aligned} \tag{320}$$

Helicopter capacity and synchronization constraints. Constraints (321) oblige $V_h = 1$ when some $Y_{th} > 0$, in this way, an allocation in the time horizon only occurs if this helicopter is used. Suppose t is the first takeoff and t' is the last landing (with $\overline{tat}$) of helicopter h. Therefore, the maximum workday tolerated for crew on board this aircraft ($\overline{wd}_h$) on the present day is respected when satisfying $(t' - \overline{tat} + 1) - t \leq \overline{wd}_h$. With that in mind, (322) guarantee this rule by preventing $Y_{t'h}$ and Y_{th} from being activated concurrently when $(t' - \overline{tat} + 1) - t > \overline{wd}_h$. Note that these constraints only verify the extremities of planning horizon, which makes them much more efficient than others that compare, for example, all $t' > t$ using "$BigM$" parameters. To make sure that the last $Yt'h = 1$ always consider $\overline{tat}$, so that it can then be discounted, we add $\overline{tat}$ to T.

$$Y_{th} \leq V_h; \;\; \forall \;\; h = 1, \ldots, H; t = tFirst_h^Y, \ldots, tLast_h^Y; \tag{321}$$

$$\begin{aligned} Y_{t'h} + Y_{th} \leq 1; \;\; \forall \;\; h = 1, \ldots, H; \\ t, t' = tFirst_h^Y, \ldots, tLast_h^Y \mid (t' - \overline{tat} + 1) - t > \overline{wd}_h; \end{aligned} \tag{322}$$

Practical precedence constraints. The flight precedence required by the problem through parameter pr_{ij} is enforced by constraints (323). Basically, they prevent flight j from being performed in periods after flight i when $pr_{ij} = 1$.

$$\sum_{\substack{h=1:\\ c_{jh}=1}}^{H} \sum_{t'=tFirst^X_{jh}}^{\min\{tLast^X_{jh},t-1\}} X_{jt'h} \leq 1 - \sum_{\substack{h=1:\\ c_{ih}=1}}^{H} X_{ith}; \quad \forall\ i,j=1,\ldots,I \mid i \neq j \wedge pr_{ij}=1; t=\overline{r}_i,\ldots,tLast^{X2}_i; \tag{323}$$

Variable pre-fixing. The next three constraints ensure the last rules of problem by means of fixing variables. Constraints (324) check if $d^{\max}$ will not be exceeded, considering the addition of $leftover_i$ in the delay. Lastly, constraints (325) avoid local transfers for entourage/mandatory flights and do not permit this type of transfer for road routes that exceed $t^{\max}$.

$$X_{ith} = 0;\ \ \forall\ i=1,\ldots,I \mid n_i = 0,3,4; h=1,\ldots,H \mid c_{ih}=1; \quad t = tFirst^X_{ih},\ldots,tLast^X_{ih} \mid (t-\overline{r}_i).factor + leftover_i > d^{\max}; \tag{324}$$

$$X_{ith} = 0;\ \ \forall\ i=1,\ldots,I; h=1,\ldots,H \mid c_{ih}=1 \wedge \quad \check{a}_i \neq \hat{a}_h \wedge (tr_{\check{a}_i,\hat{a}_h} > t^{\max} \vee n_i = 3,4); t = tFirst^X_{ih},\ldots,tLast^X_{ih}; \tag{325}$$

Domain of the decision variables. The discrete-time model is finalized by the type and domain of decision variables in (326)–(331), which in turn we have a binary integer programming (BIP). We can see that the intervals of period t were limited by the enhancements, causing the model to reduce drastically when compared with an interval $1,\ldots,T$.

$$X_{ith} \in \{0,1\};\ \ \forall\ i=1,\ldots,I; t=tFirst^X_{ih},\ldots,tLast^X_{ih}; h=1,\ldots,H \mid c_{ih}=1; \tag{326}$$
$$out_i \in \{0,1\};\ \ \forall\ i=1,\ldots,I \mid n_i = 0,1,2,3; \tag{327}$$
$$Y_{th} \in \{0,1\};\ \ \forall\ t=tFirst^Y_h,\ldots,tLast^Y_h; h=1,\ldots,H; \tag{328}$$
$$Z_{tk} \in \{0,1\};\ \ \forall\ t=tFirst^Z_k,\ldots,tLast^Z_k; k=1,\ldots,K; \tag{329}$$
$$Q_{tp} \in \{0,1\};\ \ \forall\ t=tFirst^Q_p,\ldots,tLast^Q_p; p=1,\ldots,P; \tag{330}$$
$$V_h \in \{0,1\};\ \ \forall\ h=1,\ldots,H. \tag{331}$$

6.3 Heuristic approach

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

6.3.1 Decompose-Construct-and-Improve Phase

This method shares the same theoretical heuristic framework presented in Chapter 5. In that work, a single aerodrome is considered, the crew's duty are disregarded and flight times are independent of the helicopter type. In addition to extending the problem to the case of multiple aerodromes, our heuristic does not apply the strategy of relaxing the fleet heterogeneity as seen in Section 5.3 since the current configuration of the operation requires prior knowledge of the helicopters to determine their schedules (there are non-scalar parameters that depend on identifying the helicopter to be used). Next, we describe the two parts that form *Decompose-Construct-and-Improve Phase*, *Construction Part* and *Improvement Part*.

The Construction Part

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

$$H_k^{min} = \left\lceil \frac{\sum_{i \in \mathcal{I}_k} \left[tat + min_{h \in \mathcal{H}_k} \left(tf_{ih} \right) \right]}{min \left\{ tw_k^B - tw_k^A, \; max_{h \in \mathcal{H}_k} \left(wd_h \right) \right\}} \right\rceil ; \tag{332}$$

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

$$Cn = \left(\begin{array}{c} |\mathcal{H}_k| \\ H_k^{min} \end{array} \right) = \frac{|\mathcal{H}_k|!}{H_k^{min}! \, . (|\mathcal{H}_k| - H_k^{min})!}. \tag{333}$$

In order to choose an appropriate subset of $\mathcal{H}_k$, each helicopter h is evaluated according to function $gw_h = C1. \left(I - \hat{I}_h \right) + C2. \left(I - \check{I}_h \right) + C3. \left(\hat{tf}_h / wd_h \right) + C4. w_h^3$, where $\hat{I}_h$ is the number of flights compatible with h, $\check{I}_h$ is the number of flights to which h is assigned in the original schedule, $\hat{tf}_h$ is h's average flight time considering its compatible flights, and $C1, \ldots, C4$ are coefficients to be calibrated. The score of a given combination of helicopters $Comb_l \subseteq \mathcal{H}_k$ is given by $sc_l = \sum_{h \in Comb_l} gw_h, \forall \; l = 1, \ldots, Cn$. Through the non-decreasing ordering of sc_l, the G first subsets of helicopters are selected to be checked in function of the best performance that will be observed by the allocation of flights during the rest of the *Construction Part*. We defined these subsets as $\mathcal{H}_g^*, \forall \; g = 1, \ldots, G$ (note that $\left| \mathcal{H}_g^* \right| = H_k^{min}$).

A two-flight sequence Pr_h is then constructed for each helicopter $h \in \mathcal{H}_g^*$. The start of the assembly of each sequence is done in relation to the due time $pz_i = r_i + \tilde{tf}_i$, where $\tilde{tf}_i$ is the average time of flight i. Specifically, the first and the last flight of Pr_h is chosen by $i' = \underset{i \in I_k : i \notin Pr_v, v \in H_g^*, v \neq h}{argmin} \{ pz_i \mid c_{ih} = 1\}$ and $j' = \underset{j \in I_k : j \notin Pr_v, v \in H_g^*, v \neq h}{argmax} \{ pz_j \mid c_{jh} = 1\}$, respectively. This corresponds to following an EDT rule ordering. Regarding the remaining helicopter $(|\mathcal{H}_k| - H_k^{min})$, we set $Pr_h = \emptyset, \forall\ h \in \mathcal{H}_k \backslash \bigcup_{g=1}^{G} \mathcal{H}_g^*$.

As a way of verifying feasibility, carrying out the schedule, as well as quantifying the delay of this first flight association, the heuristic executes backward programming, called *GetSchedule*, responsible for making all scheduling from flight sequences. In general, *GetSchedule* delays the flights of each Pr_h only when necessary, in order to meet the problem's constraints aimed at building schedules. For this, the routine starts the departure and arrival times of each flight as if they were planned (without delays). Based on this, there is a looping designed to "push" flights that violate the conditions related to the overlapping of helicopter in a MU (tu_i), safety briefing (sb), turnaround time of the same consecutive helicopter (tat), and calculating the arrival time of a flight to the aerodrome ($AT_i = DT_i + tf_{ih}$). The closing of this looping occurs when all these requirements are guaranteed, no matter how big the values of DT_i and AT_i may be in this first moment.

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

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

$$(i^*, j^*, h^*) = \underset{(i,j) \in I_k : i \notin Pr_h, j \in Pr_h, h \in H_g^*}{argmin} \left\{ Type_i + Prec_i + \tfrac{\tilde{H}_i}{H} + min\left(TD_{ijh}, TD_{jih}\right) \right\}\ ; \quad (334)$$

where:

- $Type_i = \begin{cases} 1, n_i \in \{1, 2\}\,; \\ 2, n_i \in \{0, 3, 4\}\,. \end{cases}$

- $Prec_i = \begin{cases} 2.I, \tilde{p}_i = 0; \\ I/\tilde{p}_i, \tilde{p}_i > 0. \end{cases}$

- $\tilde{p}_i$: number of flights that succeeds (not necessarily immediately) flight i (pr_{ij});
- $\tilde{H}_i$: number of helicopters compatible with flight i;
- TD_{ijh}: total delay obtained if flight i is inserted immediately before j (TD_{jih}, if the insertion is after).

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

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

$$OF_{it,k} = \min_{g=1,\dots,G} f_{it,g,k}; \tag{335}$$

where:

$$f_{it,g,k} = \sum_{i \in \mathcal{R}_{gk}} w_i^1 + \sum_{h \in \mathcal{H}_g^*} w_h^3 + \sum_{h \in \mathcal{H}_g^*} \sum_{i \in Pr_h} w_i^4 . D_i.$$

If $OF_{it-1,k} > OF_{it,k}$ and $H_k^{min} < |\mathcal{H}_k|$, the iteration is increased and the values of $OF_{it,k}$ and solution $sol_{it,k}\,(X, DT, AT, D)$ are stored (for iteration $it = 1$, we initialize $OF_{0,k} = +\infty$). The heuristic proceeds by doing $H_k^{min} = H_k^{min} + 1$ and repeating the steps that determine $\mathcal{H}_g^*$. It makes the first flight insertions using the EDT ordering and tries to designate the remaining flights using the $InsertFlights$ routine by the criterion (334) and the calculation of $OF_{it,k}$. This cycle is performed until the comparison between the objective functions and the availability of the fleet is considered false, which implies a convergence to the best result found ($OF^*_{it-1,k}$), thus finishing the *Constructive Part*.

The Improvement Part

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

- $N1$: Helicopter itinerary rescheduling;
- $N2$: Exchange of scheduled flights by unscheduled flights;
- $N3$: Inter-helicopter flight insertion;
- $N4$: Inter-helicopter flight exchange;
- $N5$: Intra-helicopter flight exchange.

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

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

6.3.2 Integration Phase

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

In case the procedure fails in incorporating one or more flights, we apply a second local search whose neighborhood is similar to N1 described in the *Improvement Part*. A scheduled flight i is replaced by an unscheduled $i^{'}$ in the same order of visitation, followed by the reinsertion of i in a different order of visitation or in the itinerary of a distinct helicopter. The move that results in the lowest delay is selected to produce the new incumbent solution, and the process is repeated until no improving insertion is found. If this attempt fails, a third local search is applied, whose move consists of assigning an unused compatible helicopter (if there is) to $i^{'}$.

The general scheme of the heuristic is presented by Algorithm 7. The heuristic starts with the *Decompose-Construct-and-Improve Phase*, which comprises Steps 4 to 22. For each aerodrome, the *Construction Part* and *Improvement Part* are executed in sequence.

Basically, the *Construction Part* is responsible for quantifying and determining the aircraft to be used through the best score $f_{it,g,k}$ found, and also defining the sequences and schedules of flights for each aircraft, thus storing the results $OF_{it,k}$ and $sol_{it,k}$ (X, DT, AT, D). As mentioned earlier, after obtaining a feasible solution for the first phase, the *Improvement Part* is triggered, which can be seen in Steps 20 to 22. As long as there is improvement in the incumbent solution, the five local searches are performed. Next, the *Integration Phase* tries to allocate flights that were not scheduled at their original aerodromes, according to Steps 24 to 28.

Algorithm 7: *General algorithm of the two-phase heuristic method*

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

6.4 Computational experiments

This section presents the results of numerical experiments to evaluate the computational performance of three solution approaches: (i) solve continuous-time model (285)-(313)

via CPLEX 12.10 with time limit of one hour (called M1); (*ii*) solve discrete-time model (314)-(331) through the same CPLEX version and duration limit (M2); and (*iii*) apply the two-phase heuristic (H) of Section 6.3. All these solution approaches were implemented in C++ language with the same hardware configuration used in Section 5.4.

Table 20 resumes the main characteristics of ten realistic problem instances considered in the computational experiments, ranging from 23 to 90 flights, from 7 to 24 helicopters, from 17 to 59 maritime units and from 2 to 3 aerodromes. In this table, column "Instance" indicates the instance name, while the next four columns portray the number of flights (I), helicopters (H), maritime units (P) and aerodromes (K), respectively. The remaining five columns show the number of table flights, transferred flights from the last day, transferred flights from the last two or more days, entourage flights and mandatory flights, respectively. The first eight instances used real-data provided by the company and the last two (I82_2 and I90_2) were generated based on instances I82_3 and I90_3. These instances simulate the important unexpected event that one of aerodromes (in the case, the one with the fewest flights) is closed for the whole day and its respective demand is transferred to the other aerodromes. To do this, helicopters from the closed aerodrome are disregarded and their types of booked flights, table and transfer delayed by one day flights, become, respectively, transfer delayed by one day and transfer delayed by two or more days flights. Table 20 highlights the differences between I82_2 and I82_3, and I90_2 with I90_3. From now on, the first eight instances of Table 20 are called *real* instances, whereas the last two ones are called *simulated* instances.

Table 20 – Characteristics of the realistic problem instances.

Instance	I	H	P	K	$\hat{n}_0$	$\hat{n}_1$	$\hat{n}_2$	$\hat{n}_3$	$\hat{n}_4$
I23_2	23	7	17	2	15	6	2	0	0
I35_2	35	12	22	2	23	9	0	3	0
I48_3	48	18	32	3	35	9	2	1	1
I56_3	56	18	40	3	27	21	3	2	3
I67_3	67	21	46	3	35	22	3	2	5
I71_3	71	21	50	3	40	25	2	0	4
I82_3	82	24	56	3	46	27	3	0	6
I90_3	90	24	52	3	49	31	4	1	5
I82_2	82	21	56	2	37	35	4	0	6
I90_2	90	18	52	2	39	39	6	1	5

Source: Own authorship.

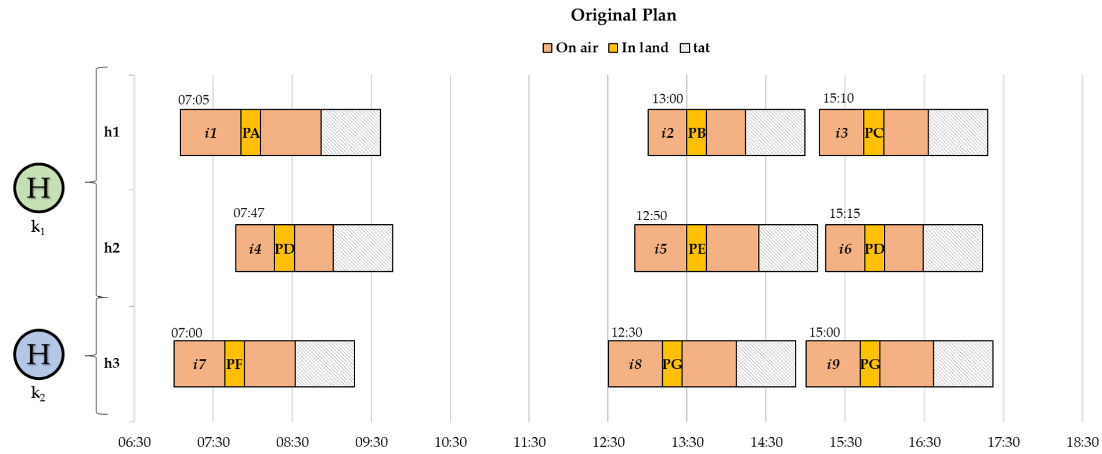
All parameter values correspond to those used by the company. As said before, tat was set to 45 minutes, sb to 5 minutes, and $d^{\max}$ and $t^{\max}$ to 4 hours (240 minutes). The remaining data of all instances, such as the scheduled departure times of flights (r_i), the duration of flight i using helicopter h (tf_{ih}), the service times of flights at maritime units (tu_i), the workdays of helicopters (wd_h), the time windows of aerodromes ($[tw_k^A, tw_k^B]$), the land transfer times (by road transportation) of flight passengers between aerodromes k and l (tr_{kl}), among others, can be obtained upon request to us. These parameters are in "hours:minutes" format but we convert them to minutes before using any solution approach. Each of weight values for the optimization were carefully defined by the company's programmers in order to reflect their relative priorities. These chosen penalties for $w^1 > w^2 > w^3 > w^4$ used in the experiments were: w_i^1 equals 320 if flight i is an entourage flight, 240 if it is a transferred flight from the last two or more days, 180 if it is a transferred flight from the last day and 80 if it is a table flight; w_{ik}^2 equals 40 if flight i is transferred to aerodrome k when it was originally scheduled in another aerodrome; w_h^3 equals 20 if helicopter h is of the pool fleet and 10 if it is of the normal fleet; w_i^4 equals 0.001 if flight i is a table, entourage or mandatory flight and 0.005 if it is a day-transferred flight. Regarding the two-phase heuristic, through computational tests we set parameter $G = 3$ (the number of aircraft subsets) and adopted parameters $C1 = 1$, $C2 = 0.1$, $C3 = 100$, and $C4 = 1$ (coefficients used to determine gw_h), all described in *Constructive Part* of *Decompose-Construct-and-Improve Phase*.

6.4.1 Toy problem

Before showing the results of real problem instances, let us consider an illustrative example with a daily timetable of only 9 table flights ($\mathcal{I} = \{i_1, i_2, ..., i_9\}$ with $n_1 = n_2 = ... = n_9 = 0$) and 2 aerodromes ($\mathcal{K} = \{k_1, k_2\}$). Figure 43 depicts the corresponding Gantt diagram for the scheduled timetable, where the flights are represented by colored bars: the orange ones represent the travel times to and from each maritime unit, s_{ih}; the yellow ones represent the time that each helicopter remains on the ground at each maritime unit, tu_i; the gray ones represent the turnaround time of each helicopter just after each flight at the aerodrome, tat. The departure times of each flight, r_i, are shown over these bars. The flights depart from the two aerodromes, represented on the left-side of figure, which have different time windows, i.e., tw_1^A = 06:55, tw_2^A = 07:00 and tw_1^B = 19:00, tw_2^B = 17:30, respectively. The flight labels are indicated inside the orange bars of figure. Three helicopters are used ($\mathcal{H} = h_1, h_2, h_3$) in this timetable, represented along the ordinate axis of figure, with a workday of 10 hours ($wd_1 = wd_2 = wd_3 = 10$). Note that helicopters h_1 and h_2 belong to aerodrome k_1 while helicopter h_3 to aerodrome k_2. There are 7 maritime units ($\mathcal{P} = \{PA, PB, ..., PG\}$) and note that there are flights with same unit destination (e.g., flights i_4 and i_6 goes to unit PD, i.e., $PD = PF$). The maritime unit labels are indicated inside the yellow bars of figure.

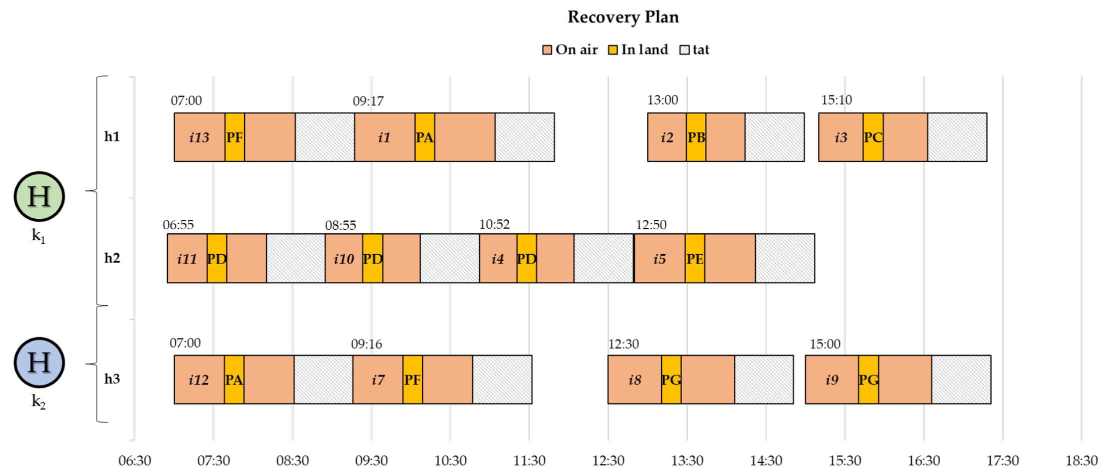
Now, let us consider that four unrealized flights (i_{10}, i_{11}, i_{12}, i_{13}) were transferred from the last days to aerodrome k_1, two of them were 2-day delay flights (i_{11}, i_{12} with $n_{11} = n_{12} = 2$), and the other two were 1-day delay flights (i_{10}, i_{13} with $n_{10} = n_{13} = 1$). The unit destination of flights i_{10}, i_{11} coincides with the destination of flights i_4, i_6 (i.e., PD); the destination of flight i_{12} coincides with the destination of flight i_1 (PA); the destination of flight i_{13} coincides with the destination of flight i_7 (PF). Therefore, these four day-transferred flights should be scheduled before the table flights because of the precedence constraints (i.e., $pr_{10,4} = pr_{10,6} = pr_{11,4} = pr_{11,6} = pr_{12,1} = pr_{13,7} = 1$). Figure 44 depicts a recovery plan after the inclusion of these day-transferred flights in the previous daily timetable.

Figure 43 – Original timetable of the toy problem.



Source: Own authorship.

Figure 44 – Recovery plan of the toy problem.



Source: Own authorship.

As shown in Figure 44, the four day-transferred flights were successfully rescheduled in the recovery plan, which required some decisions to be taken. First, the precedence constraints were met for the flights with the same unit destination - note that flights i_{11} and i_{10} were rescheduled before flight i_4 (in the same aerodrome), and flight i_{12} was scheduled before flight i_1 (in this case, in different aerodromes). For this, the departure times of flights i_1, i_4, i_7 had to be postponed in a few hours, implying in some delays $(D_1, D_4, D_7 > 0)$. Moreover, flight i_{12}, which was previously scheduled in aerodrome k_1, had to change to aerodrome k_2. Given that the departure of flight i_{12} must occur before the departure of flight i_1, the scheduling of flight i_{12} in helicopter h_1 of aerodrome k_1 would imply in delaying flight i_1 in, at least, $tf_{1,1} + tat$ minutes, which is greater than the time limit for a delay ($d^{\max}$). Thus, flight i_1 would be transferred to the next day, which would be highly penalized. Similarly, given that the departure of flight i_{12} must occur before the departure of flight i_4, the scheduling of flight i_{12} in helicopter h_2 at aerodrome k_1 would imply in delaying flight i_4 for more than the time limit of a delay and, hence, flight i_4 would be transferred to the next day. A better alternative is to move flight i_{12} from aerodrome k_1 to aerodrome k_2 and use helicopter h_3. Despite this recovery plan implying a delay for flight i_7, it is able to include more flights in the current day.

It should be observed that flight i_6 was unrealized in the recovery plan of Figure 44 and had to be transferred to the next day. Note that this flight could not be maintained as the last flight of helicopter h_2, because of the workday restriction of 10 hours of this helicopter. Similarly, if flight i_6 were assigned to helicopters h_1 or h_3, the workday limit of these helicopters would be violated as well and hence, transferring flight i_6 to the next day seems to result in a good solution. In fact, the recovery plan of Figure 44 is indeed an optimal solution for this problem, which is proven by solving to optimality model M1.

Table 21 shows in detail the solutions obtained by the approaches M1, M2 and H. Column "Solution Approach" indicates the proposed methods; column f the total solution penalty; columns f_1, f_2, f_3, f_4 the penalties regarding four terms of objective function, respectively; column "Time" the computer runtime (in seconds); column nR the number of flights transferred to the next day; column nT the number of flights that change the aerodrome; column nH the number of helicopters used. As previously specified, the solution of M1 is optimal. When comparing it with the solution of M2, we notice that the delay reached by M2 is greater (see f_4). This was due to the rounding of time as a multiple of 5 minutes. Both approaches presented exactly the same assignment and sequencing flights in the recovery plan, however, the parameters r_4 = 07:47 and tf_5 = 01:32, and the schedule DT_1 = 09:17, DT_4 = 10:52 and DT_7 = 09:16, belonging to the solution of M1 had to be increased by r_4 = 07:50, tf_5 = 01:35, DT_1 = 09:20, DT_4 = 10:55 and DT_7 = 09:20 in the solution of M2. Respecting to the solution of H, the recovery plan was different from the optimal solution. Basically four flights were assigned and sequenced differently. Despite this change, H provides a slightly higher overall flight delay (column f_4). Now about the runtime, the heuristic solution is obtained within a much shorter computational time, and this advantage of H over M1 and M2 will become more evident when solving the realist problem instances of the next section.

6.4.2 Real problem instances

Table 22 presents the results obtained by the three solution approaches for each of the real problem instances. Some of the table columns are the same as Table 21, except for the column "Instance" that indicates the instance name; column Gap[1], the optimality gap

Table 21 – Summary of the results of the toy problem.

Solution approach	f	f_1	f_2	f_3	f_4	Time (sec)	nR	nT	nH
M1	151.05	80	40	30	1.05	51.24	1	1	3
M2	151.07	80	40	30	1.07	59.36	1	1	3
H	151.93	80	40	30	1.93	0.09	1	1	3

Source: Own authorship.

(in percentage) with respect to the lower bound LB^{M1} for each instance found by solution approach M1 after one hour of execution time; column Gap^2, the gap (in percentage) with respect to the best upper bound UB^{IN} of the all f got for each instance. The values of columns Gap^1 and Gap^2 were determined as:

$$Gap^1 = (1 - LB^{\mathrm{M1}}/f).100\%; \quad (336)$$

$$Gap^2 = (1 - UB^{\mathrm{IN}}/f).100\%. \quad (337)$$

Some of the main results of this table are:

- The solution approach M1 was able to find feasible solutions without transferring flights to the next day until problem instance I48_3. However, for the remaining larger instances, the obtained recovery plans are not able to include all flights within the time limit of one hour, which is highly undesirable by the company. The high values of column Gap^1 show that the solver cannot produce tight lower bounds using the flow model under this time limit;

- Regarding the instances with three airports, the solution approach M2 is the one that presented the best upper bounds, managing to recover all flights with low utilization of the available fleet, which shows its potential in generating high quality reschedule plans. Nevertheless, in these same instances, the time limit of one hour was reached, which can be a negative point when short response times are required in practice;

- The solution approach H was the fastest to generate feasible solutions, less than 170 seconds on average. Moreover, these solutions are of good quality as the obtained recovery plans were able to reschedule all pending flights of the previous days, as well as all flights of the day, which is the most important criterion of the company.

Next, we compare the solutions in more detail obtained by approaches M1, M2 and H in each of the real instances. In instance I23_2, approach M1 achieves optimal solution just before completing one hour of execution. With a slightly higher total flight delay and with the same number of aircraft used, M2 and H are competitive at f. However, there is a big difference in computational time among the two methods, M2 with 223.53 s and H with only 0.56 s. These deviations between quality and runtime towards the approaches are kept on instance I35_2. All manage to save two aircraft from the available fleet.

Table 22 – Summary of the results of the real problem instances.

Instance	Solution approach	f	f_1	f_2	f_3	f_4	Time (sec)	Gap[1] (%)	Gap[2] (%)	nR	nT	nH
I23_2	M1	82.44	0	0	80	2.44	3,600.02	37.10	0.00	0	0	7
	M2	82.48	0	0	80	2.48	223.53	37.13	0.05	0	0	7
	H	82.52	0	0	80	2.52	0.56	37.16	0.09	0	0	7
I35_2	M1	141.32	0	0	140	1.32	3,600.18	49.89	0.00	0	0	10
	M2	141.33	0	0	140	1.33	693.01	49.90	0.01	0	0	10
	H	141.92	0	0	140	1.92	7.35	50.11	0.42	0	0	10
I48_3	M1	173.58	0	0	170	3.58	3,600.33	42.11	5.60	0	0	15
	M2	163.86	0	0	160	3.86	3,600.20	38.68	0.00	0	0	14
	H	165.58	0	0	160	5.58	16.51	39.32	1.04	0	0	14
I56_3	M1	942.39	640	80	210	12.39	3,600.12	90.40	78.51	4	2	17
	M2	202.52	0	0	180	22.52	3,600.17	55.34	0.00	0	0	15
	H	263.36	0	40	200	23.36	52.29	65.65	23.10	0	1	16
I67_3	M1	1,238.71	880	120	230	8.71	3,600.18	93.50	82.18	9	3	19
	M2	220.73	0	0	200	20.73	3,600.45	63.54	0.00	0	0	17
	H	250.39	0	0	230	20.39	148.46	67.86	11.85	0	0	19
I71_3	M1	1,504.30	1,120	120	250	14.30	3,600.05	93.33	82.62	9	3	20
	M2	261.38	0	0	240	21.38	3,600.60	61.60	0.00	0	0	20
	H	267.20	0	0	240	27.20	80.79	62.44	2.18	0	0	20
I82_3	M1	2,031.30	1,680	80	260	11.30	3,600.23	95.06	86.18	17	2	21
	M2	280.72	0	0	260	20.72	3,600.50	64.24	0.00	0	0	22
	H	311.00	0	0	290	21.00	425.83	67.72	9.74	0	0	23
I90_3	M1	3,588.34	3,360	0	210	18.34	3,600.47	96.92	91.29	29	0	18
	M2	312.58	0	0	280	32.58	3,600.51	64.66	0.00	0	0	21
	H	329.07	0	0	300	29.07	589.22	66.43	5.01	0	0	23

Source: Own authorship.

Leaving the scope of two airports and going to the first instance with three, I48_3, we discover a more apparent gap in the solution quality of M1 in view of M2 and H (yielding a $Gap^2 = 5.6\%$, given the use of one more helicopter) and in the time demanded by H among the others (16.51 s against 1 h).

Concerning instance I56_3, whilst M1 transfers four flights to the next days, changes two aerodrome bases and uses a total of 17 helicopters; M2 and H are able to include all flights with less local transfers and fleet utilization. It is worth highlighting the quality of solution obtained by M2, which did not transfer flights to other aerodromes and retained one more helicopter than H. When observing I67_3 and I71_3, again M2 stands out in terms of best solution, and H continues to have the lowest running time. Unlike M1, which rejects nine flights with three local transfers, M2 and H are capable to allocate all flights without having to change the default aerodromes.

Comparing the approaches in relation to the remaining larger instances (I82_3 and I90_3), the potential of M2 and H to allocate flights is perceptible. The results indicate that M2 and H can be suitable approaches for use in practical settings of the present ARP. On the other hand, M1 become very ineffective as the problem instances increase, involving transference of several flights to the next day, as seen in instances I82_3 and I90_3. In general, we recommend M2 in situations that do not require very quick responses, such as an interval of at least one hour, otherwise, the best choice is H.

6.4.3 Simulated problem instances

As previously mentioned, we also consider a simulated pessimistic scenario where the aerodrome with the fewest flights closes for the whole day due to an unexpected event, for example, very bad weather. In this case, all scheduled flights in the timetable of the closed aerodrome have to be either changed to the other aerodromes, or transferred to the next day. For this simulation, we consider instances I82_2 and I90_2 with only two aerodromes. These are supposed to be difficult problem instances as the recovery plans have to deal with very congested situations. Table 23 reports the results obtained with the tree solution approaches for these simulated instances. The columns of this table are the same as Table 22.

Table 23 – Summary of the results of the simulated problem instances.

Instance	Solution approach	f	f_1	f_2	f_3	f_4	Time (sec)	Gap^1 (%)	Gap^2 (%)	nR	nT	nH
I82_2	M1	5,368.93	5,200	0	150	18.93	3,600.36	98.50	87.11	45	0	14
	M2	694.25	320	80	260	34.25	3,600.47	88.39	0.31	4	2	21
	H	692.10	400	0	260	32.09	376.94	88.36	0.00	5	0	21
I90_2	M1	6,888.94	6,720	0	160	8.94	3,600.15	97.82	82.10	51	0	14
	M2	1,233.11	960	0	230	43.11	3,600.46	87.81	0.00	11	0	18
	H	1,233.25	960	0	230	43.25	504.68	87.82	0.01	11	0	18

Source: Own authorship.

Note in Table 23 that, as expected, the recovery plans of M2 and H did not reschedule all flights, even using all available fleets of the two aerodromes. In the solutions of these approaches, all mandatory, entourage and day-transferred flights from the last days were included in the recovery plans, as they are priority flights, and the flights transferred to the next day are only table flights. The required runtimes of H are around 500 seconds, which are tolerable for supporting decisions in this ARP. In particular, on instance I82_2, despite the approaches M2 and H having close results, the solutions of both are very different. While M2 rejected four flights ($f_1 = 320$) and carry out two local transfers ($f_2 = 80$), H rejected five ($f_1 = 400$) without transferring flights among aerodromes, which is practically equivalent in terms of f. Similarly to the results with real instances, approach M1 was, by far, the one that provides the largest values.

6.4.4 Sensitivity analysis for approach H

In addition to the experiments with the real-life instances of Subsection 6.4.2 (called here scenario "Real"), we perform a sensitivity analysis on the objective function terms of the problem to evaluate the robustness of the solutions obtained from heuristic method H (scenario "SA"). Thereby, in scenario SA, we perform different computational tests making changes on the objective function weights in accordance with the weight-family groupings: $\mathcal{F}_1 = \{0, 1, 2, 3\}$ (types of flights only alluding to $n_i = 0, 1, 2, 3$, i.e., without mandatory flights $n_i = 4$); $\mathcal{F}_2 = \{\text{local-transfer}\}$ (transference of flights between different aerodromes); $\mathcal{F}_3 = \{\text{normal}, \text{pool}\}$ (types of helicopters); and $\mathcal{F}_4 = \{\text{pre-scheduled}, \text{unscheduled}\}$ (referring to the types of flights, where the pre-scheduled flights are in set $\{0, 3, 4\}$ and the unscheduled flights are in $\{1, 2\}$). These groupings are based on the penalties f_1 to f_4 and they are used to distinguish the values that are different among the weights w_i^1, w_{ik}^2, w_h^3 and w_i^4 defined at the beginning of Section 6.4.

Given these groupings, the first set of tests consists of setting each penalty value to 0. Therefore, we define $\text{Test}^1(f_p)$: $w^p = 0$ for each $p = 1, \ldots, 4$. The second set of tests is in charge of leveling the values for each weight-family grouping. We choose by the average values acquired from each $\mathcal{F}_p$ with $p \neq 2$ (since all $w_{ik}^2 = 40$ for $k \neq \check{a}_i$, which results in scenario Real). Let $\tilde{w}^1 = 200$, $\tilde{w}^3 = 15$ and $\tilde{w}^4 = 0.003$ be the average values of $\mathcal{F}_1$, $\mathcal{F}_3$ and $\mathcal{F}_4$, respectively; hence, we have $\text{Test}^2(\mathcal{F}_p)$: $w^p = \tilde{w}^p$ for each $p \in \{1, 3, 4\}$. The third set of tests cancels (i.e., makes null) the weights in an accumulated way, starting with the elements in $\mathcal{F}_4$ and going up to those in $\mathcal{F}_1$, in order to maintain the stability of the objective function. The details of the third set of tests are given below (we discard $\text{Test}_2^3(\mathcal{F}_4)$ since $\text{Test}_2^3(\mathcal{F}_4) = \text{Test}^1(f_4)$):

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

For each instance, we perform the sensitivity analysis using all these tests and heuristic H. The results from these experiments are presented in the tables of Appendix D.

They are shown in the form of deviations (variations), i.e., differences between the values of scenarios SA and Real. Among the results, we select the ones of instance I90_3 to detail here (Table 24) because this is the largest instance representing the most complex operating situation. The columns of Table 24 are: Test description (name of the proposed test); ΔTime (sec) (variation of the computational time in seconds); ΔR_M (variation of the total number of mandatory flights moved or transferred to the next day); ΔR_E (variation of the total number of entourage flights moved to the next day); ΔR_{D-2} (variation of the total number of 2 day-transfer or more flights moved to the next day); ΔR_{D-1} (variation of the total number of 1 day-transfer flights moved to the next day); ΔR_{D0} (variation of the total number of table flights moved to the next day); ΔT_{A1} (variation of the total number of flights that left aerodrome A1 and went to another using a local-transfer); ΔT_{A2} (variation of the total number of flights that left aerodrome A2 and went to another using a local-transfer); ΔT_{A3} (variation of the total number of flights that left aerodrome A3 and went to another using a local-transfer); ΔH_P (variation of the total number of type-pool helicopters used); ΔH_N (variation of the total number of type-normal helicopters used); ΔDL_2 (variation of the total delay of unscheduled flights, in minutes); ΔDL_1 (variation of the total delay of pre-scheduled flights, in minutes); ΔnR (variation of the total number of all flights moved to the next day); ΔnT (variation of the total number of all flights that changed the aerodrome); ΔnH (variation of the total number of all used helicopters); and, ΔnDL (variation of the total delay of all flights, in minutes).

Table 24 – Result variations between scenarios SA and Real for instance I90_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-140.35	0	0	2	6	1	0	0	0	-3	0	-1,130	-744	9	0	-3	-1,874
$\text{Test}^1(f_2)$	142.02	0	0	0	0	0	0	0	0	0	0	-409	291	0	0	0	-118
$\text{Test}^1(f_3)$	134.8	0	0	0	0	0	0	0	0	1	0	-401	-145	0	0	1	-546
$\text{Test}^1(f_4)$	-119.79	0	0	0	0	0	0	0	0	0	0	1,561	-761	0	0	0	800
$\text{Test}^2(\mathcal{F}_1)$	126.17	0	0	0	0	0	0	0	0	0	0	-409	291	0	0	0	-118
$\text{Test}^2(\mathcal{F}_3)$	138.43	0	0	0	0	0	0	0	0	1	-1	-406	308	0	0	0	-98
$\text{Test}^2(\mathcal{F}_4)$	-46.39	0	0	0	0	0	0	0	0	0	0	1,518	-1,564	0	0	0	-46
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-24.65	0	0	0	0	0	0	0	0	0	0	-324	645	0	0	0	321
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-117.5	0	0	0	0	0	0	0	0	0	0	1,561	-761	0	0	0	800
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-112.6	0	0	0	0	0	0	0	0	1	-1	1,562	-823	0	0	0	739
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-126.51	0	0	0	0	0	0	0	0	1	-1	1,562	-823	0	0	0	739
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-124.74	0	0	0	0	1	0	0	0	1	-2	1,429	-516	1	0	-1	913
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-93.91	0	0	0	2	1	0	0	0	1	-3	1,088	-321	3	0	-2	767
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-93.61	0	0	0	2	1	0	0	0	1	-3	1,088	-321	3	0	-2	767
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-104.13	0	0	0	2	1	0	0	0	1	-3	1,088	-321	3	0	-2	767

Source: Own authorship.

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

Going to the second set of tests, leveling the weights of family $\mathcal{F}_1$ in $\text{Test}^2(\mathcal{F}_1)$ implies changing the solutions that have rejected or canceled flights, i.e., variations related to the types of flights. Since the heuristic solutions do not reject flights, one would expect no changes in day-transfers and fleet usage. $\text{Test}^2(\mathcal{F}_2)$ allowed the heuristic to change the used helicopters, achieving a slight decrease in the total delay. Defining the same value for the weights w^3 varies the delays between pre-scheduled flights and those to be scheduled. The heuristic increased the delay for unscheduled flights and decreased it for pre-scheduled flights, being able to improve the total delay.

In the third set of tests, which consists of gradually turning off the weights, starting with the lowest value and going up to the highest value, one would expect that the solution method to be used would have a standard behavior of worsening the metric corresponding to the weight-off to try to improve others that are on. As the best solution for scenario Real does not contain rejected flights and no local-transfer, we conclude that the heuristic generated coherent solutions, only worsening the metrics with weight-off, as shown in the corresponding rows from $\text{Test}^3_1(\{\text{pre-scheduled}\})$ to $\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$.

Chapter 7

Concluding remarks and future research

In this dissertation, we study integrated aircraft routing problems considering crew pairing in the context of non-scheduled air transportation industry. Differently from traditional (scheduled) airlines, in which several operational requirements are already pre-established over a long period of time, the non-scheduled mode encompasses complex decisions in a more dynamic and unpredictable environment, defined by events that occur on short-term horizons and differ from period to period, based upon the state of resources that are provided at a given moment.

Taking this context as a basis, the dissertation addresses real cases of two companies that offer non-scheduled air transportation services, categorized in the literature as a dial-a-flight problem (DAFP) and an aircraft recovery problem (ARP), respectively. We seek to contribute scientifically in terms of the development of optimization models, which are intended to adequately represent the evaluated problems, as well as the elaboration of exact and heuristic solution methods that are effective in practice, providing quality solutions in acceptable computing times.

Specifically, the first problem is related to on-demand air transportation with a fractional ownership contract, involving the integration of aircraft routing process with crew assignment. For this problem, we proposed a compact MIP model that allows us to define efficiently, what routes each aircraft must execute, considering features of the fleet and its different costs and qualities, maintenance schedules, the possibility of outsourcing for customer trips, and also the legislation related to crew members, imposed by the incorporation of rest periods, breaks (split duties), maximum flight time in a single duty, overtime payments and the crew availability (pilots' time windows), in order to meet some of the labor rights considered fundamental. Despite its practical relevance, this theme has been barely explored in the literature. This mathematical model is an extension of the formulation of Munari and Alvarez (2019), intended for aircraft routing without considering crew assignment to minimize operating costs arising from positioning flights and aircraft upgrade services. As solution method, we apply in the proposed model an approach that combines two classical MIP-based heuristics: the relax-and-fix (R&F) and fix-and-optimize (F&O). In our adaptation, R&F-F&O heuristic iteratively solves relaxed MIP subproblems partitioned by requests. From sorting the starting time with anticipation of requests, we can opt for the forward and backward temporal strategies. The fixation of variables is aimed at two artifices: fix the allocation ($X1$) or fix the sequencing of flights ($X2$). We also customize the R&F to be able to choose how many partitions of variables will be considered in the linear relaxation (qtR). To obtain new bounds and optimal results, especially from larger instances, we proposed a branch-and-price (B&P) algorithm together with a set partitioning formulation for this DAFP. To achieve a good performance in practice, our B&P relies on stabilized column generation processes based on well-centered solutions provided by an interior-point approach, known as primal-dual column generation method (PDCGM). The subproblem were solved by a tailored labeling algorithm, responsible for complying with all the crew rules during aircraft routes. Additional features include different types of branching rules (two-step and strong branching strategies) and a primal MIP-based heuristic to find faster incumbent solutions.

To assess the adequacy of our approaches, we performed computational experiments with real-life data provided by the company that motivated this study. The results obtained by proposed model revealed that the commercial solver CPLEX was able to achieve optimal results in 52 out of 65 instances (an 80% yield), within the maximum runtime of one hour. From the experiments, we can conclude that the proposed model with CPLEX can optimally solve instances with up to 200,000 constraints and 700,000 variables, which corresponds to the journey logs typically used by the airline. Related to R&F-F&O approach, we compared eight variants that differ in terms of temporal strategy, artifice, number of R&F partitions (I), qtR and number of F&O partitions (J). The results showed that the R&F-F&O[$Backward$, $X2$, $I = 4$, $qtR = 1$, $J = 3$] variant stood out from the others, capable of getting solutions in all instances. When comparing it with the proposed model, the heuristic variant proved to be better, being able to achieve greater cost reductions in all instances with open gap, and having results very close to those with closed gap. Regarding the B&P algorithm, we first tested the impact of switching the branching strategies and turning on and off the primal heuristic. The B&P variant using the two-step branching rule and with the primal heuristic turned on had the best computational performance. Compared to the previous approaches, the best B&P variant was far superior, being able to close the optimality gaps in all 65 instances in less than 38 min (in total). Although B&P excelled in relation to the R&F-F&O heuristic, it is worth remembering that its construction/implementation is time-consuming and complex, unlike MIP-based heuristics, which in a short development time, we were capable of finding quality solutions.

The second problem is alluded to a real-life short-term rescheduling problem of helicopter flights from one onshore airport to several maritime units (e.g., offshore oil rigs, gas-producing platforms, etc.). Due to unexpected circumstances, such as bad weather or aircraft mechanical failures, the original timetable very often cannot be fully met, resulting in flight delays on the same day or even postponements to the following days, interrupting many of essential activities and services on a current day. In general aspects, this study 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 began the study of this ARP by considering the characteristics of a single runway aerodrome, heterogeneous fleet, airport time windows, briefing of safety, minimum time on the ground between two consecutive flights of an aircraft, mandatory flight precedence, minimum time intervals between takeoffs from the airport and same maritime units and maximum allowed flight delays, among many others. We propose two mixed-integer programming models to formulate the problem with all relevant features, one based on the extension of traditional network flow models and other that relies on a novel takeoff event-based representation. Additionally, we have brought an effective tailored heuristic to the studied problem. In summary, the method is composed of two parts, constructive and improvement. The construction part is a relax-and-repair type procedure that alternates between a relaxed construction stage, responsible for finding a base solution without considering the heterogeneity of fleet, at first, and a repairing stage, in charge of adjusting the base solution with the fleet heterogeneity constraints, if necessary, to achieve a feasible solution, in fact for the problem. Finally, the heuristic runs six local search methods on the improvement part, aiming to improving the incumbent solution obtained from the construction part.

The performance of each approach was analyzed through computational experiments with instances created from real-life data provided by a Brazilian oil company, which gave us all the requirements and specificities of their operations. For each model and instance, we ran the general-purpose B&C method of CPLEX using four configurations: default settings, with the local branching heuristic turned on, with the RINS heuristic turned on, and with both heuristics turned on. The results revealed that in both models, CPLEX, together with the tested configurations, was able to solve well only instances related to airport A, which are in the order of 8 to 14 flights. On the other hand, the computational experiments show the potential of the proposed 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 proposed heuristic 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 heuristic method, conversely, include all flights and hence, no flights need to be transferred to the next day. As the runtime is less than three minutes, the heuristic 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 nine different scenarios, which confirmed these results and showed the robustness of proposed method.

We extended the ARP to involve workday duration of crew members and the case of multiple aerodromes. Along with the restrictions from the previous work, in this one, we need to determine joint daily flight reschedules for all aerodromes, while trying to recover all pending flights and reduce flight transfers among these airports, usage of different helicopters (with crew on board) and overall flight delays. In this panorama, we proposed a detailed continuous-time network-flow MIP model that appropriately represents this ARP with all the specific characteristics of its real settings, but which is unable to provide good recovery plans for realistic problem instances within acceptable runtimes using the CPLEX solver. Therefore, we also suggest a discrete-time approximation BIP model together with a two-phase heuristic to cope with larger problem instances, ultimately aiming at producing reschedules with full flight recovery and within acceptable computer runtimes for the flight operators.

According to the results, the continuous-time model is effective in producing full flight recovery only in small-sized problem instances (until two aerodromes). On the other hand, the discrete-time model provided the best solutions with no transferred flights for all instances, proving to perform much better on the CPLEX's B&C than the continuous-time model, however, taking the time limit of one hour for solving instances from 48 flights. The two-phase heuristics was able to reschedule all flights at relatively short computing times (167.83 s on average), which may be suitable for successful application in practice. Finally, the robustness of heuristic method was also evaluated by a sensitivity analysis performed in eight different scenarios, varying the objective function terms in each of them.

When comparing ARP with DAFP, we can see as similarities, obviously, in addition to non-scheduled condition, the dynamism with demand appears in short-term planning, which is guided by a reference (customer request or flight timetables), and based on that, there is an entire effort directed towards to servicing it. Moreover, both are concerned with reducing operating costs and increasing the service level, and respecting certain aviation rules, such as time windows, delay tolerance, turnaround times, among others. Nevertheless, there are differences that stand out because one problem is of a commercial nature and the other not, and in which the specificities that each operation has. For example, in the ARP, the delay, not only results in daily and overtime payments, but also it harms productivity (interconnection activities, underwater and deck maintenance, and the shift works cannot occur at the same time with the aircraft trips) and negatively affect the work environment (employees who are on board at maritime units are somewhat confined, with great physical and psychological strain). As a matter of fact, each landing and takeoff is considered a high-risk activity, and work schedules should be minimized as much as possible. In the first problem, it is not achievable to make flight connections. For reasons of safety and physical space, it is not feasible to leave passengers waiting at any aerodrome (not authorized by ANAC, for example). Another dissimilarity is seen in relation to the possibility of postponing flights to other schedules, since in DAFP it is not allowed, and in ARP is.

The next step in this research should be to extend the DAFP. Something desirable for the airline and very relevant for the sector would be the consideration of time zones and the circadian low effect on the crew journey. Business aviation operational demands require 24-hour-a-day activities that can include shift, night, irregular and unpredictable work schedules, and time zone changes. These factors challenge human physiology and can result in performance impairing fatigue and an increased risk to safety. Scientific information and practical experience with fatigue, human sleep and circadian physiology can improve aviation safety by providing guidance in mitigating and managing factors that contribute to fatigue in operational settings. Because of this, we suggest adding the so-called window of circadian low (WCL) in our DAFP, which is a period between 02:00 hours and 05:59 hours, and when a crew works in it, the maximum allowed time without rest is reduced. Another interest may be to separate the crew from aircraft, in order to generate the schedule for both.

In particular, we verified that one of six months provided from the airline's journey logs presented disparity regarding the computational difficulty when solving by a solution method, something curious since all the instances of experimentation were built by grouping three days of company operation. To understand the reason for this difference, we applied statistical techniques from multivariate data analysis. Firstly, we checked whether the data had normality condition (which has been proven) and then, parametric tests such as multivariate analysis of variance (MANOVA) were carried out to seek statistical evidence that the measurements of location and spread differ in terms of months. The *p-value* is significant (probability less than 0.05). In addition, we employed in our experimentation the causality technique for prediction and classification, called discriminant function analysis, and the interdependence technique that allows grouping objects (cases or variables), named as clustering method. Both technique showed in their univariate tests that the counters related to, number of maintenance requests, number of overlapping windows (of requests), number of constraints and variables generated from proposed model, have statistical significance to discriminate the groups (i.e., the months of operation). The insertion of this topic in the dissertation was not considered, because applications of these techniques would demand a long description on the text, and also fall outside the doctorate's concentration area.

As for the ARP, the next steps could be to comprise decisions on assigning passengers to flights, aircraft capacity varying on route (fuel consumption), merging and splitting flights, guarantee the occurrence of the lunch break for the crew, and trips that visit more than one maritime unit on the same route.

Something interesting that we have noticed in the discrete-time model of the ARP is that formulation also has the capacity to provide dual bounds. To this, we just invert the way in which the parameters are discretized in 5 minutes, i.e., what it is rounded up becomes rounded down, and vice-versa. This modification allows us to apply the Benders decomposition (BD). Our idea was to put the discrete-time model into the master problem, whilst the continuous variables (responsible for the schedule times) from the continuous-time model would be placed in the subproblem. Note that, in this configuration, BD could generate optimal results for the original problem, even using a formulation that was reduced by dividing the number of periods from the time horizon by 5 minutes. However, although we arrived at a lean subproblem (variables with only two indices, constraints with reduced intervals, existence of artifices to strengthen linear relaxation), CPLEX was unable to solve the required subproblems for larger instances in the time limit of one hour. A way to face this obstacle might be to decompose the subproblem into one more level, which would allow us to adopt a B&P algorithm (where the routes of each aircraft are solved by column generation), or even another Benders (to generate a subproblem with only linearly relaxed variables). Both second-level decompositions have the potential to provide high-density Pareto cuts for the first-level BD. The proposed subproblem for BD is in Appendix E.

Respecting the B&P algorithm proposed for the DAFP, we can find several methods in the literature that could improve our B&P's overall performance. For example, the mono-directional backward extension could inserted to the labeling algorithm, which would give us the possibility of applying the bidirectional extension, a technique that merges forward and backward labels at a so-called half-way point (the middle of planning horizon), accelerating the convergence. Similar to a branch-and-cut method, valid inequalities can be added in the master problem to strengthen bound values, decreasing the number of nodes in the enumeration branch-and-bound (B&B) tree. One of the most successful valid inequalities for set partitioning formulations is the subset row cuts (SRCs) (JEPSEN et al., 2008). Typically, these inequalities bring a stronger linear relaxation to the restricted master problem (RMP). To improve the application of our strong branching in the B&B tree, we could solve each RMP using a simplex-type algorithm with limited pivoting (or with a higher optimality tolerance), which is ideal for quickly estimating perturbations in the objective function. As the strong branching rule solves several LPs to evaluate the most promising variable to branch, a simplex method from a good MIP solver (e.g., Gurobi, CPLEX, etc.) might be suitable to get out of this ordinary computational overburden. Another advance on this rule would be to store in memory the branching evaluations already made during the search (in order to use them later as pseudo-costs), calculating for the only new variables, those not yet computed (LINDEROTH; SAVELSBERGH, 1999). Correlated to the node selection strategy employed on the B&B algorithm, something that could improve the best-first rule would be the execution of a diving heuristic (BIXBY et al., 1999; ACHTERBERG et al., 2008), a procedure that goes down some branch of the B&B tree until it hits a pre-specified tree depth, reaches infeasibility, or finds a feasible solution. As our DAFP is large (given the real-life instances) and B&P's focus to obtain optimal results, a diving heuristic might provide incumbents early, which makes it possible to decrease the size of the B&B tree by optimality pruning. Following the same idea, another node selection scheme could be a technique called best estimate criterion (BÉNICHOU et al., 1971; FORREST; HIRST; TOMLIN, 1974), capable of determining an estimate of the best feasible integer solution obtainable from a given node, taking into account pseudo-costs from variables that are fractional.

Ultimately, concerning heuristic methods, we identified that some metaheuristics have shown effectiveness in VRP's applications, such as iterated local search (ILS) (LOURENÇO; MARTIN; STÜTZLE, 2003), and adaptive large neighborhood search (ALNS) (ROPKE; PISINGER, 2006). ILS repeatedly applies a local search algorithm to modified versions of a good solution found previously. In this way, it is like a clever version of the stochastic hill climbing with random restarts algorithm. The intuition behind the algorithm is that random restarts can help to locate many local optima in a problem and that better local optima are often close to other local optima. Therefore modest perturbations to existing local optima may locate better or even the best solutions to an optimization problem. Essentially, an ILS algorithm uses four basic components: (*i*) an initial solution; (*ii*) a local search procedure; (*iii*) a perturbation mechanism; and (*iv*) an acceptance criterion. ALNS is an extension of large neighborhood search (SHAW, 1998), in which a pool of simple heuristics depend on past success compete to modify the incumbent solution. This is done iteratively through alternating destroy and repair operators. The choices of heuristics are given by an adaptive search engine, based on Russian roulette of the previously obtained scores. In addition to being employed individually as solution approaches, these heuristics can also be combined with exact methods, as is the case of our B&P. In this hybridization, the heuristic is triggered to solve subproblems whenever convenient, whose purpose is to obtain new incumbents and dual bounds of the problem at a more competitive computational time, still guaranteeing the optimality of the method (BOSCHETTI; MANIEZZO, 2010; ALVAREZ; MUNARI, 2017).

References

ACHTERBERG, T. et al. Constraint integer programming: A new approach to integrate cp and mip. In: SPRINGER. **International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming**. [S.l.], 2008. p. 6–20.

ACHTERBERG, T.; KOCH, T.; MARTIN, A. Branching rules revisited. **Operations Research Letters**, Elsevier, v. 33, n. 1, p. 42–54, 2005.

AKTURK, M. S.; ATAMTURK, A.; GUREL, S. Aircraft rescheduling with cruise speed control. **Operations Research**, v. 62, n. 4, p. 829–845, 2014.

AL-THANI, N. A.; AHMED, M. B.; HAOUARI, M. A model and optimization-based heuristic for the operational aircraft maintenance routing problem. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 72, p. 29–44, 2016.

ALVAREZ, A.; MUNARI, P. An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. **Computers & Operations Research**, Elsevier, v. 83, p. 1–12, 2017.

AMOR, H. M. B.; DESROSIERS, J.; FRANGIONI, A. On the choice of explicit stabilizing terms in column generation. **Discrete Applied Mathematics**, Elsevier, v. 157, n. 6, p. 1167–1184, 2009.

ANDERSSON, T. Solving the flight perturbation problem with meta heuristics. **Journal of Heuristics**, Springer, v. 12, n. 1, p. 37–53, 2006.

ANDERSSON, T.; VÄRBRAND, P. The flight perturbation problem. **Transportation planning and technology**, Taylor & Francis, v. 27, n. 2, p. 91–117, 2004.

APPLEGATE, D. et al. **Finding cuts in the TSP (A preliminary report)**. [S.l.]: Citeseer, 1995. v. 95.

ARCHETTI, C.; SPERANZA, M. G. A survey on matheuristics for routing problems. **EURO Journal on Computational Optimization**, Elsevier, v. 2, n. 4, p. 223–246, 2014.

ARGUELLO, M. F. **Framework for exact solutions and heuristics for approximate solutions to airlines' irregular operations control aircraft routing problem**. [S.l.]: The University of Texas at Austin, 1997.

ARGÜELLO, M. F.; BARD, J. F.; YU, G. A grasp for aircraft routing in response to groundings and delays. **Journal of Combinatorial Optimization**, Springer, v. 1, n. 3, p. 211–228, 1997.

ARIAS, P. et al. A methodology combining optimization and simulation for real applications of the stochastic aircraft recovery problem. In: IEEE. **2013 8th EUROSIM Congress on Modelling and Simulation**. [S.l.], 2013. p. 265–270.

ARIKAN, U.; GÜREL, S.; AKTÜRK, M. S. Flight network-based approach for integrated airline recovery with cruise speed control. **Transportation Science**, INFORMS, v. 51, n. 4, p. 1259–1287, 2017.

ARMACOST, A. P. et al. Ups optimizes its air network. **Interfaces**, INFORMS, v. 34, n. 1, p. 15–25, 2004.

BAKER, K. R. **Introduction to sequencing and scheduling**. [S.l.]: John Wiley & Sons, 1974.

BALDACCI, R.; MINGOZZI, A.; ROBERTI, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. **European Journal of Operational Research**, Elsevier, v. 218, n. 1, p. 1–6, 2012.

BARD, J. F.; YU, G.; ARGUELLO, M. F. Optimizing aircraft routings in response to groundings and delays. **IIE Transactions**, Taylor & Francis, v. 33, n. 10, p. 931–947, 2001.

BASTOS, Y.; FLECK, J.; MARTINELLI, R. A stochastic programming approach for offshore flight scheduling. In: **15th IFAC Workshop on Discrete Event Systems**. [S.l.: s.n.], 2020. p. 1–7.

BEASLEY, J. E.; CHRISTOFIDES, N. An algorithm for the resource constrained shortest path problem. **Networks**, Wiley Online Library, v. 19, n. 4, p. 379–394, 1989.

BELOBABA, P.; ODONI, A.; BARNHART, C. **The Global Airline Industry**. [S.l.]: John Wiley & Sons, Ltd, 2015. 494 p. ISBN 9780470740774.

BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. **Numerisch Mathematik**, Springer-Verlag, v. 4, p. 238–252, 1962.

BÉNICHOU, M. et al. Experiments in mixed-integer linear programming. **Mathematical programming**, Springer, v. 1, p. 76–94, 1971.

BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, MCB UP Ltd, 2002.

BIOLCHINI, J. et al. **Systematic review in software engineering**. Rio de Janeiro, 2005. 45 p.

BIXBY, E. R. et al. Mip: Theory and practice—closing the gap. In: SPRINGER. **IFIP Conference on System Modeling and Optimization**. [S.l.], 1999. p. 19–49.

BOSCHETTI, M. A.; MANIEZZO, V. Combining exact methods and heuristics. **Wiley Encyclopedia of Operations Research and Management Science**, Wiley Online Library, 2010.

BRIANT, O. et al. Comparison of bundle and classical column generation. **Mathematical programming**, Springer, v. 113, p. 299–344, 2008.

BRUNNER, J. O. Rescheduling of flights during ground delay programs with consideration of passenger and crew connections. **Transportation Research Part E: Logistics and Transportation Review**, v. 72, p. 236–252, 2014. ISSN 1366-5545.

BUSACKER, R.; GOWEN, P. A procedure for determining a family of minimum-cost flow patterns. **Operations Research Office Technical Report**, v. 15, 1961.

CADARSO, L.; VAZE, V. Passenger-centric integrated airline schedule and aircraft recovery. **Transportation Science**, INFORMS, v. 57, n. 3, p. 813–837, 2023.

CAMPBELL, I.; ALI, M. M.; SILVERWOOD, M. Solving a dial-a-flight problem using composite variables. **Top**, Springer, v. 28, n. 1, p. 123–153, 2020.

CAO, J.-M.; KANAFANI, A. Real-time decision support for integration of airline flight cancellations and delays part i: mathematical formulation. **Transportation Planning and Technology**, Routledge, v. 20, n. 3, p. 183–199, 1997.

______. Real-time decision support for integration of airline flight cancellations and delays part ii: algorithm and computational experiments. **Transportation Planning and Technology**, Routledge, v. 20, n. 3, p. 201–217, 1997.

CLARKE, M. D. D. **Development of heuristic procedures for flight rescheduling in the aftermath of irregular airline operations**. PhD Thesis (PhD Thesis) — Massachusetts Institute of Technology, 1997.

CLAUSEN, J. et al. Disruption management in the airline industry—concepts, models and methods. **Computers & Operations Research**, Elsevier, v. 37, n. 5, p. 809–821, 2010.

CORDEAU, J.-F. et al. Minimizing user inconvenience and operational costs in a dial-a-flight problem for flying safaris. **INFOR: Information Systems and Operational Research**, Taylor & Francis, v. 61, n. 1, p. 104–140, 2023.

______. Transportation on demand. **Handbooks in operations research and management science**, Elsevier, v. 14, p. 429–466, 2007.

COSTA, L.; CONTARDO, C.; DESAULNIERS, G. Exact branch-price-and-cut algorithms for vehicle routing. **Transportation Science**, INFORMS, v. 53, n. 4, p. 946–985, 2019.

DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations research**, INFORMS, v. 8, n. 1, p. 101–111, 1960.

D'ARIANO, A.; PISTELLI, M.; PACCIARELLI, D. Aircraft retiming and rerouting in vicinity of airports. **IET Intelligent Transport Systems**, Institution of Engineering and Technology, v. 6, p. 433–443(10), December 2012. ISSN 1751-956X. Available at: <https://digital-library.theiet.org/content/journals/10.1049/iet-its.2011.0182>.

de Campos, R. A.; VIEIRA, T.; MUNARI, P. A branch-and-cut algorithm for aircraft routing with crew assignment for on-demand air transportation. In: SPRINGER. **Computational Logistics: 12th International Conference, ICCL 2021, Enschede, The Netherlands, September 27–29, 2021, Proceedings**. [S.l.], 2021. p. 611–626.

De La Vega, J. et al. Model-based solution approach for a short-term flight rescheduling problem in aerial passenger transportation to maritime units. **International Transactions in Operational Research**, Wiley Online Library, v. 29, n. 6, p. 3400–3434, 2022.

_____. Helicopter recovery in an oil and gas industry: Model and solution approaches. **EURO Journal on Transportation and Logistics**, Elsevier, v. 11, p. 100084, 2022.

DESAULNIERS, G. et al. Daily aircraft routing and scheduling. **Management Science**, INFORMS, v. 43, n. 6, p. 841–855, 1997.

DESAULNIERS, G.; LESSARD, F.; HADJAR, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. **Transportation Science**, INFORMS, v. 42, n. 3, p. 387–404, 2008.

DESAULNIERS, G.; MADSEN, O. B.; ROPKE, S. The vehicle routing problem with time windows. In: TOTH, P.; VIGO, D. (Ed.). **Vehicle routing: Problems, methods, and applications**. [S.l.]: Eds, 2014, (MOS/SIAM Ser Optim). p. 119–159.

DESROCHERS, M. **An algorithm for the shortest path problem with resource constraints**. [S.l.]: Université de Montréal, Centre de recherche sur les transports, 1986.

DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. **Operations Research Letters**, Elsevier, v. 10, n. 1, p. 27–36, 1991.

DEY, S. S. et al. A theoretical and computational analysis of full strong-branching. **Mathematical Programming**, Springer, p. 1–34, 2023.

DING, Y. et al. Towards efficient airline disruption recovery with reinforcement learning. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier, v. 179, p. 103295, 2023.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Mathematical programming**, Springer, v. 91, p. 201–213, 2002.

DROR, M. Note on the complexity of the shortest path models for column generation in vrptw. **Operations Research**, INFORMS, v. 42, n. 5, p. 977–978, 1994.

DUNBAR, M.; FROYLAND, G.; WU, C.-L. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. **Comput. Oper. Res.**, Elsevier Science Ltd., GBR, v. 45, p. 68–86, May 2014. ISSN 0305-0548. Available at: <https://doi.org/10.1016/j.cor.2013.12.003>.

EGGENBERG, N.; SALANI, M.; BIERLAIRE, M. Constraint-specific recovery network for solving airline recovery problems. **Computers & Operations Research**, v. 37, n. 6, p. 1014 – 1026, 2010. ISSN 0305-0548.

ENGINEER, F. G.; NEMHAUSER, G. L.; SAVELSBERGH, M. W. Dynamic programming-based column generation on time-expanded networks: Application to the dial-a-flight problem. **INFORMS Journal on Computing**, INFORMS, v. 23, n. 1, p. 105–119, 2011.

ERDMANN, A. et al. Modeling and solving an airline schedule generation problem. **Annals of Operations Research**, Springer, v. 107, n. 1, p. 117–142, 2001.

ERKAN, H.; ERKIP, N. K.; SAFAK Özge. Collaborative decision making for air traffic management: A generic mathematical program for the rescheduling problem. **Computers & Industrial Engineering**, v. 137, p. 106016, 2019.

ESHKEVARI, M. L.; KOMIJAN, A. R.; BARADARAN, V. A bi-objective model for aircraft and crew recovery considering individual flight legs and maintenance requirements. **Engineering Optimization**, Taylor & Francis, p. 1–27, 2023.

ESPINOZA, D. et al. Per-seat, on-demand air transportation part i: Problem description and an integer multicommodity flow model. **Transportation Science**, INFORMS, v. 42, n. 3, p. 263–278, 2008.

______. Per-seat, on-demand air transportation part ii: Parallel local search. **Transportation Science**, INFORMS, v. 42, n. 3, p. 279–291, 2008.

EVLER, J. et al. Integration of turnaround and aircraft recovery to mitigate delay propagation in airline networks. **Computers & Operations Research**, Elsevier, v. 138, p. 105602, 2022.

FAGERHOLT, K.; FOSS, B. A.; HORGEN, O. J. A decision support model for establishing an air taxi service: a case study. **Journal of the Operational Research Society**, v. 60, n. 9, p. 1173–1182, sep 2009. ISSN 0160-5682.

FANTAZZINI, T. et al. Hierarchical goal programming approaches to solve a discrete-time formulation for the aircraft recovery problem of a brazilian oil and gas company. Submitted. 2024.

FEILLET, D. A tutorial on column generation and branch-and-price for vehicle routing problems. **4or**, Springer, v. 8, n. 4, p. 407–424, 2010.

FERREIRA, D.; MORABITO, R.; RANGEL, S. Um modelo de otimização inteira mista e heurísticas relax and fix para a programação da produção de fábricas de refrigerantes de pequeno porte. **Production**, SciELO Brasil, v. 18, p. 76–88, 2008.

FILAR, J. A. et al. A model for adaptive rescheduling of flights in emergencies (marfe). **Journal of Industrial & Management Optimization**, American Institute of Mathematical Sciences, v. 3, n. 2, p. 335, 2007.

FORREST, J.; HIRST, J.; TOMLIN, J. A. Practical solution of large mixed integer programming problems with umpire. **Management Science**, INFORMS, v. 20, n. 5, p. 736–773, 1974.

FOY, T. L. L. **On-demand air transportation flight scheduling**. PhD Thesis (PhD Thesis) — School of Computational and Applied Mathematics - University of the Witwatersrand, Johannesburg, South Africa, 2013.

FUKASAWA, R. et al. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. **Mathematical programming**, Springer, v. 106, n. 3, p. 491–511, 2006.

GALVÃO, R. D.; 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**, v. 49, n. 2, p. 266–270, 1990.

GAO, M. et al. Flight rescheduling responding to large-area flight delays. **Kybernetes**, v. 41, n. 10, p. 1483–1496, 2012. ISSN 0368492X.

GHIANI, G.; LAPORTE, G.; MUSMANNO, R. **Introduction to logistics systems planning and control**. [S.l.]: John Wiley & Sons, 2004.

GOFFIN, J.-L.; VIAL, J.-P. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. **Optimization methods and software**, Taylor & Francis, v. 17, n. 5, p. 805–867, 2002.

GONDZIO, J. Interior point methods 25 years later. **European Journal of Operational Research**, Elsevier, v. 218, n. 3, p. 587–601, 2012.

GONDZIO, J.; GONZÁLEZ-BREVIS, P.; MUNARI, P. New developments in the primal–dual column generation technique. **European Journal of Operational Research**, Elsevier, v. 224, n. 1, p. 41–51, 2013.

______. Large-scale optimization with the primal-dual column generation method. **Mathematical Programming Computation**, Springer, v. 8, n. 1, p. 47–82, 2016.

GRIBKOVSKAIA, I.; HALSKAU, O. S.; KOVALYOV, M. Y. Minimizing takeoff and landing risk in helicopter pickup and delivery operations. **Omega**, v. 55, p. 73 – 80, 2015. ISSN 0305-0483.

HAOUARI, M.; MANSOUR, F. Z.; SHERALI, H. D. A new compact formulation for the daily crew pairing problem. **Transportation Science**, v. 53, n. 3, p. 811–828, 2019. Available at: <https://doi.org/10.1287/trsc.2018.0860>.

HASSAN, L.; SANTOS, B. F.; VINK, J. Airline disruption management: A literature review and practical challenges. **Computers & Operations Research**, Elsevier, p. 105137, 2020.

HERMETO, T. S.; HERMETO, N. D. S. S.; HAWSON, P. N. São tomé aircraft parking area redesign: An international infrastructure regulation study case. In: **23nd ATRS World Conference Amsterdam, Netherland**. [S.l.: s.n.], 2019.

HICKS, R. et al. Bombardier flexjet significantly improves its fractional aircraft ownership operations. **Interfaces**, INFORMS, v. 35, n. 1, p. 49–60, 2005.

HU, Y. et al. Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft. **Expert Systems with Applications**, v. 83, p. 283–299, 2017. ISSN 0957-4174.

______. Integrated recovery of aircraft and passengers with passengers' willingness under various itinerary disruption situations. **Computers & Industrial Engineering**, Elsevier, v. 161, p. 107664, 2021.

HUANG, Z. et al. An iterative cost-driven copy generation approach for aircraft recovery problem. **European Journal of Operational Research**, Elsevier, v. 301, n. 1, p. 334–348, 2022.

IAC-1223. **Instrução de Aviação Civil – Normas para confecção e aprovação de HOTRAN**. [S.l.: s.n.], 2000.

ICAO. **Review of the Classification and Definitions used for Civil Aviation Activities**. Sta/10-wp/7. Montréal, Canada, 2009.

IRNICH, S.; DESAULNIERS, G. Shortest path problems with resource constraints. In: **Column generation**. [S.l.]: Springer, 2005. p. 33–65.

JAFARI, N.; ZEGORDI, S. H. The airline perturbation problem: considering disrupted passengers. **Transportation Planning and Technology**, Taylor & Francis, v. 33, n. 2, p. 203–220, 2010.

______. Simultaneous recovery model for aircraft and passengers. **Journal of the Franklin Institute**, v. 348, n. 7, p. 1638–1655, 2011. ISSN 0016-0032.

JARRAH, A. I. Z. et al. A decision support framework for airline flight cancellations and delays. **Transportation Science**, v. 27, n. 3, p. 266–280, 1993.

JEPSEN, M. et al. Subset-row inequalities applied to the vehicle-routing problem with time windows. **Operations Research**, INFORMS, v. 56, n. 2, p. 497–511, 2008.

JI, C. et al. A novel rescheduling algorithm for the airline recovery with flight priorities and airport capacity constraints. **Asia-Pacific Journal of Operational Research**, World Scientific, v. 38, n. 05, p. 2140025, 2021.

JOHNSON, L. A.; MONTGOMERY, D. C. **Operations research in production planning, scheduling, and inventory control**. [S.l.]: John Wiley & Sons, 1974.

KARAESMEN, I. et al. Scheduling multiple types of time shared aircraft: Models and methods for practice. In: **Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, Stern School of Business, New York University**. [S.l.: s.n.], 2005. p. 19–38.

KESKINOCAK, P.; TAYUR, S. Scheduling of time-shared jet aircraft. **Transportation Science**, INFORMS, v. 32, n. 3, p. 277–294, 1998.

KHALED, O. et al. A compact optimization model for the tail assignment problem. **European Journal of Operational Research**, v. 264, p. 548–557, 2018.

KHIABANI, A. et al. A mathematical model for integrated aircraft and crew recovery after a disruption: a benders' decomposition approach. **Journal of Modelling in Management**, Emerald Publishing Limited, v. 18, n. 6, p. 1740–1761, 2023.

KLABJAN, D. et al. Solving large airline crew scheduling problems: Random pairing generation and strong branching. **Computational Optimization and Applications**, Springer, v. 20, p. 73–91, 2001.

LAND, A.; DOIG, A. An automatic method of solving discrete programming problems. **Econometrica**, v. 28, n. 3, p. 497–520, 1960.

LE, M.; GAO, J.; ZHAN, C. Solving the airline recovery problem based on vehicle routing problem with time window modeling and genetic algorithm. In: IEEE. **2013 Ninth International Conference on Natural Computation (ICNC)**. [S.l.], 2013. p. 822–828.

LE, M.-l.; WU, C.-c. Solving airlines disruption by considering aircraft and crew recovery simultaneously. **Journal of Shanghai Jiaotong University (Science)**, Springer, v. 18, p. 243–252, 2013.

LEE, D. W. et al. A traffic engineering model for air taxi services. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier, v. 44, n. 6, p. 1139–1161, 2008.

LEE, J.; LEE, K.; MOON, I. A reinforcement learning approach for multi-fleet aircraft recovery under airline disruption. **Applied Soft Computing**, Elsevier, v. 129, p. 109556, 2022.

LEE, J.; MARLA, L.; JACQUILLAT, A. Dynamic disruption management in airline networks under airport operating uncertainty. **Transportation Science**, INFORMS, v. 54, n. 4, p. 973–997, 2020.

LETCHFORD, A. N.; SALAZAR-GONZÁLEZ, J.-J. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 244, n. 3, p. 730–738, 2015.

LIANG, Z. et al. On a new rotation tour network model for aircraft maintenance routing problem. **Transportation Science**, INFORMS, v. 45, n. 1, p. 109–120, 2011.

_____. Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. **Transportation Research Part B: Methodological**, v. 78, p. 238–259, 2015. ISSN 0191-2615.

_____. A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. **Transportation Research Part B: Methodological**, Elsevier, v. 113, p. 70–90, 2018.

LIMA, M. G. de; BELDERRAIN, M. C. N. O fluxo logístico da carga aérea internacional brasileira. In: SIMPǾSIO DE PESQUISA OPERACIONAL E LOGÍSTICA MARINHA (SPOLM). Rio de Janeiro/RJ, 2008.

LIN, H.; WANG, Z. Flight scheduling for airport closure based on sequential decision. In: IEEE. **2018 4th International Conference on Information Management (ICIM)**. [S.l.], 2018. p. 241–245.

LINDEROTH, J. T.; SAVELSBERGH, M. W. A computational study of search strategies for mixed integer programming. **INFORMS Journal on Computing**, INFORMS, v. 11, n. 2, p. 173–187, 1999.

LIU, H.; SUN, F.; ZHANG, Y. An air-rail inter-modal strategy for aircraft recovery. **Chinese Journal of Aeronautics**, Elsevier, v. 35, n. 6, p. 240–249, 2022.

LIU, T.-K.; CHEN, C.-H.; CHOU, J.-H. Optimization of short-haul aircraft schedule recovery problems using a hybrid multiobjective genetic algorithm. **Expert Systems with Applications**, Elsevier, v. 37, n. 3, p. 2307–2315, 2010.

LIU, T.-K.; JENG, C.-R.; CHANG, Y.-H. Disruption management of an inequality-based multi-fleet airline schedule by a multi-objective genetic algorithm. **Transportation Planning and Technology**, Taylor & Francis, v. 31, n. 6, p. 613–639, 2008.

LIU, T.-K. et al. Applications of multi-objective evolutionary algorithm to airline disruption management. In: IEEE. **2006 IEEE international conference on systems, man and cybernetics**. [S.l.], 2006. v. 5, p. 4130–4135.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2003. p. 320–353.

LØVE, M. et al. Disruption management for an airline—rescheduling of aircraft. In: SPRINGER. **Applications of Evolutionary Computing: EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN Kinsale, Ireland, April 3–4, 2002 Proceedings**. [S.l.], 2002. p. 315–324.

______. Using heuristics to solve the dedicated aircraft recovery problem. **Central European Journal of Operations Research**, v. 13, n. 2, 2005.

LÜBBECKE, M. E.; DESROSIERS, J. Selected topics in column generation. **Operations research**, INFORMS, v. 53, n. 6, p. 1007–1023, 2005.

LUO, S.; YU, G. On the airline schedule perturbation problem caused by the ground delay program. **Transportation Science**, v. 31, n. 4, p. 298–311, 1997.

MAHER, S. J. Solving the integrated airline recovery problem using column-and-row generation. **Transportation Science**, INFORMS, v. 50, n. 1, p. 216–239, 2016.

MAHER, S. J.; DESAULNIERS, G.; SOUMIS, F. Recoverable robust single day aircraft maintenance routing problem. **Computers & Operations Research**, Elsevier, v. 51, p. 130–145, 2014.

MAHESHWARI, A.; DELAURENTIS, D. Novel hierarchical markov decision process framework to enable ridesharing in on-demand air service operations. In: **AIAA AVIATION 2021 FORUM**. [S.l.: s.n.], 2021. p. 3173.

MANE, M.; CROSSLEY, W. A. Allocation and design of aircraft for on-demand air transportation with uncertain operations. **Journal of aircraft**, v. 49, n. 1, p. 141–150, 2012.

MANIEZZO, V.; STÜTZLE, T.; VOSS, S. **Matheuristics: Algorithms and Implementations**. [S.l.]: Springer, 2021.

MARSTEN, R. E.; HOGAN, W. W.; BLANKENSHIP, J. W. The boxstep method for large-scale optimization. **Operations Research**, INFORMS, v. 23, n. 3, p. 389–405, 1975.

MARTIN, C.; JONES, D.; KESKINOCAK, P. Optimizing on-demand aircraft schedules for fractional aircraft operators. **Interfaces**, INFORMS, v. 33, n. 5, p. 22–35, 2003.

MATHAISEL, D. F. Decision support for airline system operations control and irregular operations. **Computers & Operations Research**, v. 23, n. 11, p. 1083 – 1098, 1996. ISSN 0305-0548.

MCCARTNEY, S. How airlines spend your airfare. **The Wall Street Journal**, 6 2012. Available at: <http://online.wsj.com/news/articles/SB10001424052702303296604577450581396602106>.

MENEZES, F. et al. Optimizing helicopter transport of oil rig crews at petrobras. **INFORMS Journal on Applied Analytics**, v. 40, n. 5, p. 408–416, 2010.

MERCIER, A.; SOUMIS, F. An integrated aircraft routing, crew scheduling and flight retiming model. **Computers & Operations Research**, v. 34, p. 2251–2265, 08 2007.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the ACM (JACM)**, ACM New York, NY, USA, v. 7, n. 4, p. 326–329, 1960.

MITROFF, I. I. et al. On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon. **Interfaces**, INFORMS, v. 4, n. 3, p. 46–58, 1974.

MORABITO, R.; PUREZA, V. Modelagem e simulação. **Cauchick Miguel, PAC et al. Metodologia de pesquisa em engenharia de produção e gestão de operações. Rio de Janeiro: Elsevier**, p. 165–192, 2010.

MORENO, L.; ARAGÃO, M. P. de; UCHOA, E. Column generation based heuristic for a helicopter routing problem. In: ÀLVAREZ, C.; SERNA, M. (Ed.). **Experimental Algorithms**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 219–230.

MOTA, M. M.; MOTA, I. F. D. L.; SERRANO, D. G. **Applied Simulation and Optimization: In Logistics, Industrial and Aeronautical Practice**. [S.l.]: Springer, 2015.

MUNARI, P.; ALVAREZ, A. Aircraft routing for on-demand air transportation with service upgrade and maintenance events: Compact model and case study. **Journal of Air Transport Management**, v. 75, p. 75 – 84, 2019. ISSN 0969-6997. Available at: <http://www.sciencedirect.com/science/article/pii/S0969699717303538>.

MUNARI, P.; GONDZIO, J. Using the primal-dual interior point algorithm within the branch-price-and-cut method. **Computers & Operations Research**, Elsevier, v. 40, n. 8, p. 2026–2036, 2013.

MUNARI, P.; MORABITO, R. A branch-price-and-cut algorithm for the vehicle routing problem with time windows and multiple deliverymen. **Top**, Springer, v. 26, n. 3, p. 437–464, 2018.

NAKANO, D. Métodos de pesquisa adotados na engenharia de produção e gestão de operações. **Metodologia de pesquisa em engenharia de produção e gestão de operações**, Campus Rio de Janeiro, v. 2, p. 65–74, 2010.

NAWAZ, M.; ENSCORE, E. E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega**, v. 11, n. 1, p. 91 – 95, 1983. ISSN 0305-0483. Available at: <http://www.sciencedirect.com/science/article/pii/0305048383900889>.

OLIVEIRA, B. B. et al. A relax-and-fix-based algorithm for the vehicle-reservation assignment problem in a car rental company. **European Journal of Operational Research**, Elsevier, v. 237, n. 2, p. 729–737, 2014.

ÖNCAN, T.; ALTINEL, I. K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. **Computers & Operations Research**, Elsevier, v. 36, n. 3, p. 637–654, 2009.

PECIN, D. et al. Improved branch-cut-and-price for capacitated vehicle routing. **Mathematical Programming Computation**, Springer, v. 9, p. 61–100, 2017.

PESSOA, A. et al. In-out separation and column generation stabilization by dual price smoothing. In: SPRINGER. **Experimental Algorithms: 12th International Symposium, SEA 2013, Rome, Italy, June 5-7, 2013. Proceedings 12**. [S.l.], 2013. p. 354–365.

_____. A generic exact solver for vehicle routing and related problems. **Mathematical Programming**, Springer, v. 183, p. 483–523, 2020.

POCHET, Y.; WOLSEY, L. A. **Production planning by mixed integer programming**. [S.l.]: Springer, 2006. v. 149.

QIAN, F.; GRIBKOVSKAIA, I.; HALSKAU, O. S. Helicopter routing in the norwegian oil industry: Including safety concerns for passenger transport. **International Journal of Physical Distribution & Logistics Management**, v. 41, n. 4, p. 401–415, 2011. ISSN 0960-0035.

QIAN, F. et al. Passenger and pilot risk minimization in offshore helicopter transportation. **Omega**, v. 40, n. 5, p. 584–593, 2012. ISSN 0305-0483.

_____. Minimization of passenger takeoff and landing risk in offshore helicopter transportation: Models, approaches and analysis. **Omega**, v. 51, p. 93–106, 2015. ISSN 0305-0483.

QIANG, G.; XIAO-WEI, T.; JIN-FU, Z. Research on greedy simulated annealing algorithm for irregular flight schedule recovery model. In: IEEE. **2009 IEEE International Conference on Grey Systems and Intelligent Services (GSIS 2009)**. [S.l.], 2009. p. 1469–1475.

RAKSHIT, A.; KRISHNAMURTHY, N.; YU, G. System operations advisor: A real-time decision support system for managing airline operations at united airlines. **INFORMS Journal on Applied Analytics**, v. 26, n. 2, p. 50–58, 1996.

REXING, B. et al. Airline fleet assignment with time windows. **Transportation Science**, v. 34, p. 1–20, 02 2000.

RHODES-LEADER, L. et al. A multi-fidelity modelling approach for airline disruption management using simulation. **Journal of the Operational Research Society**, Taylor & Francis, v. 73, n. 10, p. 2228–2241, 2022.

_____. Airline disruption recovery using symbiotic simulation and multi-fidelity modelling. Operational Research Society Simulation Workshop, 2018.

_____. Multi-fidelity simulation optimisation for airline disruption management. In: IEEE. **2018 Winter Simulation Conference (WSC)**. [S.l.], 2018. p. 2179–2190.

RONEN, D. Scheduling charter aircraft. **Journal of the Operational Research Society**, v. 51, n. 3, p. 258–262, mar 2000. ISSN 0160-5682.

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. **Transportation science**, Informs, v. 40, n. 4, p. 455–472, 2006.

ROSENBERGER, J. M.; JOHNSON, E. L.; NEMHAUSER, G. L. Rerouting aircraft for airline recovery. **Transportation Science**, v. 37, n. 4, p. 408–421, 2003.

ROUSSEAU, L.-M.; GENDREAU, M.; FEILLET, D. Interior point stabilization for column generation. **Operations Research Letters**, Elsevier, v. 35, n. 5, p. 660–668, 2007.

SAHLING, F. et al. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. **Computers & Operations Research**, Elsevier, v. 36, n. 9, p. 2546–2553, 2009.

SANTANA, M. et al. The aircraft recovery problem: A systematic literature review. **EURO Journal on Transportation and Logistics**, Elsevier, p. 100117, 2023.

SANTOS, L. M. et al. A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes. **European Journal of Operational Research**, Elsevier, v. 245, n. 2, p. 581–590, 2015.

ŠARČEVIĆ, T.; ROCHA, A. P.; CASTRO, A. J. Artificial bee colony algorithm for solving the flight disruption problem. In: SPRINGER. **International Conference on Practical Applications of Agents and Multi-Agent Systems**. [S.l.], 2018. p. 72–81.

SHAO, Q. et al. Flight recovery method of regional multiairport based on risk control model. **Mathematical Problems in Engineering**, Hindawi, v. 2020, 2020.

SHAPIRO, J. F. **Modeling the supply chain**. [S.l.]: Duxbury Resource Center, 2001.

SHAW, P. Using constraint programming and local search methods to solve vehicle routing problems. In: SPRINGER. **International conference on principles and practice of constraint programming**. [S.l.], 1998. p. 417–431.

SHEBALOV, S.; KLABJAN, D. Robust airline crew pairing: Move-up crews. **Transportation Science**, v. 40, n. 3, p. 300–312, 2006. Available at: <https://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0131>.

SOUSA, H. et al. Airline disruption management: dynamic aircraft scheduling with ant colony optimization. In: **7th International Conference on Agents and Artificial Intelligence (ICAART 2015)**. [S.l.: s.n.], 2015.

SUBRAMANIAN, A.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for a class of vehicle routing problems. **Computers & Operations Research**, Elsevier, v. 40, n. 10, p. 2519–2531, 2013.

SUMARTI, N.; BRAHMANDITA, I. B.; AQSHA, M. A. A simulation of optimal model on fractional aircraft ownership (fao) management. In: **MENDEL**. [S.l.: s.n.], 2022. v. 28, n. 2, p. 1–8.

SUN, F.; LIU, H.; ZHANG, Y. Integrated aircraft and passenger recovery with enhancements in modeling, solution algorithm, and intermodalism. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 23, n. 7, p. 9046–9061, 2021.

TANG, X. et al. Optimization model and algorithm of unbalanced aircraft recovery. In: **International Conference on Transportation Engineering 2009**. [S.l.: s.n.], 2009. p. 3008–3013.

TEODOROVIĆ, D.; GUBERINIĆ, S. Optimal dispatching strategy on an airline network after a schedule perturbation. **European Journal of Operational Research**, v. 15, n. 2, p. 178–182, 1984. ISSN 0377-2217.

TEODOROVIĆ, D.; STOJKOVIĆ, G. Model for operational daily airline scheduling. **Transportation Planning and Technology**, Routledge, v. 14, n. 4, p. 273–285, 1990.

______. Model to reduce airline schedule disturbances. **Journal of Transportation Engineering**, v. 121, n. 4, p. 324–331, 1995.

THENGVALL, B. G.; BARD, J. F.; YU, G. Balancing user preferences for aircraft schedule recovery during irregular operations. **IIE Transactions**, v. 32, n. 3, p. 181–193, Mar 2000. ISSN 1573-9724.

______. A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. **Transportation Science**, INFORMS, v. 37, n. 4, p. 392–407, 2003.

THENGVALL, B. G.; YU, G.; BARD, J. F. Multiple fleet aircraft schedule recovery following hub closures. **Transportation Research Part A: Policy and Practice**, v. 35, n. 4, p. 289 – 308, 2001. ISSN 0965-8564.

TILK, C.; GOEL, A. Bidirectional labeling for solving vehicle routing and truck driver scheduling problems. **European Journal of Operational Research**, Elsevier, v. 283, n. 1, p. 108–124, 2020.

TOTH, P.; VIGO, D. **Vehicle routing: problems, methods, and applications**. [S.l.]: SIAM, 2014.

UHLMANN, I. R.; FRAZZON, E. M. Production rescheduling review: Opportunities for industrial integration and practical applications. **Journal of manufacturing systems**, Elsevier, v. 49, p. 186–193, 2018.

VANDERBECK, F. Implementing mixed integer column generation. In: **Column generation**. [S.l.]: Springer, 2005. p. 331–358.

VIEIRA, T. et al. Exact and heuristic approaches to reschedule helicopter flights for personnel transportation in the oil industry. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier, v. 151, p. 102322, 2021.

VINK, J. et al. Dynamic aircraft recovery problem-an operational decision support framework. **Computers & Operations Research**, Elsevier, v. 117, p. 104892, 2020.

VOS, H.-W. M.; SANTOS, B. F.; OMONDI, T. Aircraft schedule recovery problem–a dynamic modeling framework for daily operations. **Transportation Research Procedia**, Elsevier, v. 10, p. 931–940, 2015.

WANG, D. et al. Flight schedule recovery: a simulation-based approach. **Asia-Pacific Journal of Operational Research**, World Scientific, v. 36, n. 06, p. 1940010, 2019.

WANG, N. et al. A data-driven heuristic method for irregular flight recovery. **Mathematics**, MDPI, v. 11, n. 11, p. 2577, 2023.

WENTGES, P. Weighted dantzig-wolfe decomposition for linear mixed-integer programming. **International Transactions in Operational Research**, Elsevier, v. 4, n. 2, p. 151–162, 1997.

WOLSEY, L. A. **Integer programming**. [S.l.]: John Wiley & Sons, 1998. v. 52.

WU, C.; LE, M. A new approach to solve aircraft recovery problem. In: **The Second Int. Conf. on Advanced Communications and Computation (INFOCOMP 2012)**. [S.l.: s.n.], 2012. p. 148–154.

WU, Z. et al. Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming. **Neurocomputing**, Elsevier, v. 269, p. 232–255, 2017.

XIULI, Z.; JINFU, Z. Grey programming for irregular flight scheduling. In: IEEE. **2007 IEEE international conference on grey systems and intelligent services**. [S.l.], 2007. p. 1607–1611.

XIULI, Z.; YANCHI, G. Study on graps-aco algorithm for irregular flight rescheduling. In: IEEE. **2012 International Conference on Computer Science and Service System**. [S.l.], 2012. p. 266–269.

XU, H.; HAN, S. Weighted time-band approximation model for flight operations recovery considering simplex group cycle approaches in china. **Mathematical Problems in Engineering**, Hindawi, v. 2016, 2016.

XU, H. et al. The time-band approximation model on flight operations recovery model considering random flight flying time in china. In: IEEE. **2015 IEEE International Conference on Systems, Man, and Cybernetics**. [S.l.], 2015. p. 695–700.

XU, Y.; WANDELT, S.; SUN, X. A distributionally robust optimization approach for airline integrated recovery under in-flight pandemic transmission risks. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 152, p. 104188, 2023.

YAN, S.; CHEN, Y.-C. Flight rescheduling, fleet rerouting and passenger reassignment for typhoon disruption events. **Transportation Letters**, Taylor & Francis, v. 14, n. 8, p. 818–837, 2022.

YAN, S.; TU, Y.-P. Multifleet routing and multistop flight scheduling for schedule perturbation. **European Journal of Operational Research**, Elsevier, v. 103, n. 1, p. 155–169, 1997.

YAN, S.; YANG, D.-H. A decision support framework for handling schedule perturbation. **Transportation Research Part B: Methodological**, v. 30, n. 6, p. 405 – 419, 1996. ISSN 0191-2615.

YANG, W. et al. Aircraft and crew scheduling for fractional ownership programs. **Annals of Operations Research**, Springer, v. 159, n. 1, p. 415–431, 2008.

YAO, Y. et al. Strategic planning in fractional aircraft ownership programs. **European Journal of Operational Research**, Elsevier, v. 189, n. 2, p. 526–539, 2008.

YETIMOĞLU, Y. N.; AKTÜRK, M. S. Aircraft and passenger recovery during an aircraft's unexpected unavailability. **Journal of Air Transport Management**, Elsevier, v. 91, p. 101991, 2021.

ZEGHAL, F. M. et al. Flexible aircraft fleeting and routing at tunisair. **Journal of the Operational Research Society**, Taylor & Francis, v. 62, n. 2, p. 368–380, 2011.

ZHANG, C. Two-Stage Heuristic Algorithm for Aircraft Recovery Problem. **Discrete Dynamics in Nature and Society**, v. 2017, p. 12, 2017. ISSN 1026-0226.

ZHANG, D.; LAU, H. H.; YU, C. A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. **Computers & Industrial Engineering**, Elsevier, v. 87, p. 436–453, 2015.

ZHANG, D. et al. A math-heuristic algorithm for the integrated air service recovery. **Transportation Research Part B: Methodological**, v. 84, p. 211–236, 2016. ISSN 0191-2615.

ZHAO, A.; BARD, J. F.; BICKEL, J. E. A two-stage approach to aircraft recovery under uncertainty. **Journal of Air Transport Management**, Elsevier, v. 111, p. 102421, 2023.

ZHAO, T.; CHEN, X. A weight-table based heuristic algorithm for aircraft recovery problem. In: IEEE. **2018 37th Chinese Control Conference (CCC)**. [S.l.], 2018. p. 2242–2246.

ZHU, B.; ZHU, J.-f.; GAO, Q. A stochastic programming approach on aircraft recovery problem. **Mathematical Problems in Engineering**, Hindawi, v. 2015, 2015.

ZWAN, F. Van der; WILS, K.; GHIJS, S. Development of an aircraft routing system for an air taxi operator. In: **Aeronautics and Astronautics**. [S.l.]: IntechOpen, 2011.

Appendices

APPENDIX A

Supplementary content of Chapter 3

The purpose of this appendix is to support and supplement the content presented in Chapter 3. In A.1, we show again the notation already defined in Section 3.1, in a way that makes it easier to query the sets, parameters, and decision variables. In A.2, we exhibit the computational results of each R&F-F&O variant (suggested in Subsection 3.4.3).

A.1 Notation defined for the proposed model

From the real-world data provided by the company and the DAFP description made in Section 3.1, we re-present here all the required notation for a better understanding of the proposed model found in Subsection 3.2.2. Sets, parameters and decision variables were divided/organized into the inheritance from the base model and the six operational topics of aviation labor rights (Table 25 to 32), according to what was shown in Section 3.1 and Subsections 3.2.2.1-3.2.2.6. Once the proposed model has an extensive explanation, we believe that this section can function as a effective glossary.

Table 25 – Notation of the base model.

Notation	Definition
$\mathcal{K} = \{1, \ldots, K\}$	set of airports in which the company operates
$\mathcal{V} = \{1, \ldots, V\}$	set of available aircraft
$\mathcal{P} = \{1, \ldots, P\}$	set of aircraft types (to simplify the notation, we assume that $\mathcal{P}$ follows a non-descending order regarding the quality of aircraft types)
$\mathcal{L}$	set of customer (or live) requests
$\mathcal{M}$	set of maintenance requests (hence, $\mathcal{M} \cap \mathcal{L} = \emptyset$)
$\mathcal{R} = \{0\} \cup \mathcal{L} \cup \mathcal{M}$	set of requests, where 0 is a dummy request used as the first and last request serviced by any aircraft
$\mathcal{V}^p$	subset of aircraft corresponding to type p
k^v	initial (pre-designated) airport of aircraft v
$\check{p}^v$	type of aircraft v
$\hat{p}^r$	required type of aircraft in request r
i^r, j^r	origin and destination airports, respectively, of the request r
c_p	travel cost per time unit of an aircraft of type p, in \$/min
TF_{ij}^p	travel time between airports i and j for an aircraft of type p, in minutes
av_v	exact time at which aircraft v becomes available to fly for the first time in the planning horizon, in minutes
tat_k^r	turnaround time required for an aircraft at airport k before servicing request r, in minutes ($tat_k^r = 0;\ \forall\ r \in \mathcal{M}, k \in \mathcal{K}$)
st_r	planned starting time of request r, in minutes
$\Delta_{\mathcal{L}}$	maximum delay allowed to start servicing any customer request, in minutes
v^r	index of the aircraft that must undergo the maintenance request r
TL_r	duration of maintenance request r, in minutes
$\Delta_{\mathcal{M}}$	maximum tolerance of the anticipation/postponement of a maintenance event, in minutes
Cf_{rs}^v	positioning cost when aircraft v flies without customers between requests r and s
Cup_r^v	upgrade cost by choosing an aircraft v with type better (superior) than the one contracted in request r
$y_{rs}^v \in \{0, 1\}$	binary variable that assumes 1, if only if, aircraft v services requests r and s, consecutively
$W_r \geq 0$	continuous variable that represents the exact time that request r can be serviced, in minutes

Source: Own authorship.

Table 26 – Notation for the crew rest rules.

Notation	Definition
$maxDuty$	maximum crew work time allowed without rest in a single duty
PRE, POS	crew presentation time applied to the beginning/end of each duty
$minRest$	minimum uninterrupted crew rest time among two consecutive duties
$U_r \geq 0$	continuous variable that measures the accumulation of work time up to a certain request r since the inclusion of last rest
$E1_{rs} \in \{0, 1\}$	binary variable that is 1 if only if a rest for the crew members exists before a ferry leg between requests r and s
$E2_{rs} \in \{0, 1\}$	binary variable that is 1 if only if a rest for the crew members exists after a ferry leg between requests r and s
$E12_{rs} \in \{0, 1\}$	binary variable that is 1 if only if a rest for the crew members exists before and after a ferry leg between requests r and s
$E_{rs} \in \{0, 1\}$	binary variable that is 1 if only if a rest for the crew members exists (regardless of a positioning flight) between requests r and s
$E0_{vr} \in \{0, 1\}$	binary variable that is 1 if only if a rest for the crew members exists after a ferry leg between the initial airport of aircraft v and the origin airport of request $r \in \mathcal{L}$
$lengthCurrR1_r \geq 0$	continuous variable that quantifies the time without ferry leg added at the end of a current duty by request r
$lengthCurrR2_r \geq 0$	continuous variable that quantifies the time with ferry leg added at the end of a current duty by request r
$lengthNextR1_r \geq 0$	continuous variable that quantifies the time with ferry leg to be added at the beginning of the next duty by request r
$lengthNextR2_r \geq 0$	continuous variable that quantifies the time without ferry leg to be added at the beginning of the next duty by request r
$lengthAmongR12_r \geq 0$	continuous variable that quantifies the occurred ferry time between the end and start of consecutive duties by request r
$RestM_r \geq 0$	continuous variable that counts the rest time given to a crew in request r, taking into account the use of subsequent maintenance events

Source: Own authorship.

Table 27 – Notation for the crew break rules.

Notation	Definition
DL_f, DU_f	ground time's lower and upper bound belonging to range f
$B^f_{rs} \in \{0, 1\}$	binary variable that takes 1 if and only if the ground time between requests r and s is classified at time range f
$GT_{rs} \in \mathbb{R}$	continuous variable that quantifies the ground time value between requests r and s
$acumGT_r \geq 0$	continuous variable that accumulates the potential ground time until request r, to be used on the split duty proceeding
$Duty_{rs} \in \mathbb{R}$	continuous variable that quantifies the work time between requests r and s, if there is no ground time accumulation among them
$DA_{rs} \in \{0, 1\}$	binary variable that assumes value 1 if and only if there is ground time accumulation between requests r and s given the use of maintenance events

Source: Own authorship.

Table 28 – Notation for the maintenance utilization rules.

Notation	Definition
$firstM_r \in [0, 1]$	continuous variable that assumes value 1 if and only if a maintenance (or dummy) request precedes request $r \in \mathcal{M}$ without ferry leg among them
$acumRest_r \geq 0$	continuous variable that accumulates the potential ground time until request r, to be used as rest

Source: Own authorship.

Table 29 – Notation for the flying time rule.

Notation	Definition
$maxflying$	maximum total flight time allowed in a duty
$Q_r \geq 0$	continuous variable that accumulates live and ferry times until request r in a duty
$lengthCurrFD1_r \geq 0$	continuous variable that counts the duration of a live leg at the end of a current duty by request r
$lengthCurrFD2_r \geq 0$	continuous variable that counts the duration of a ferry leg at the end of a current duty by request r
$lengthCurrFD12_r \geq 0$	continuous variable used to account for the ferry time that may occur between the end and start of consecutive duties by request r

Source: Own authorship.

Table 30 – Notation for the pilot rostering rule.

Notation	Definition
nTW_v	number of pilot time windows belonging to aircraft v
$[PicTWa_{vt}, PicTWb_{vt}]$	time window t of a pilot assigned to aircraft v
$zl_r^{vt} \in \{0,1\}$	binary variable that assumes 1 if and only if customer request r is served inside time window t of a pilot allocated to aircraft v
$zf_r^{vt} \in \{0,1\}$	binary variable that assumes 1 if and only if a ferry leg exists after request r and it is in time window t of a pilot assigned to aircraft v
$stPicFerry_r \geq 0$, $edPicFerry_r \geq 0$	continuous variables related to the exact time to start/end of a ferry leg from request r on the time horizon
$stPicFerry0_v \geq 0$, $edPicFerry0_v \geq 0$	continuous variables that correspond to exact time to start/end of the first ferry leg executed by aircraft v

Source: Own authorship.

Table 31 – Notation of the overtime and outsourcing rules.

Notation	Definition
$overtPerc$	percentage used on the travel cost to pay overtime's crew members
$Cout_r$	outsourcing cost to carry out live request r
$overR_r \geq 0$	continuous variable that quantifies the overtime performed by the crew on live request r
$overF_r \geq 0$	continuous variable that quantifies the overtime performed by the crew on a ferry leg (if any) at request r
$over0_v \geq 0$	continuous variable that quantifies the overtime obtained by the first ferry leg (if any) of aircraft v
$out_r \in \{0,1\}$	binary variable that assumes 1 whether live request r is serviced by another company (a outsourcing event)

Source: Own authorship.

Table 32 – Notation of the sufficiently large numbers.

Expression
$M_r^1 = st_r + \Delta_{\mathcal{L}},\ \forall\ r \in \mathcal{L}$
$M_r^2 = st_r + \Delta_{\mathcal{M}},\ \forall\ r \in \mathcal{M}$
$M_r^3 = st_r + \max\{\Delta_{\mathcal{L}}, \Delta_{\mathcal{M}}\}, \forall\ r \in \mathcal{L} \cup \mathcal{M}$
$M_r^4 = \max_{v \in \mathcal{V}, s \in \mathcal{L} \cup \mathcal{M}: s \neq r}\{PRE + TL_r + TF^{\tilde{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\tilde{p}^v}_{j^r i^s} + tat^s_{i^s}\}, \forall\ r \in \mathcal{L} \cup \mathcal{M}$
$M_r^5 = M_r^4 + (POS + minRest + PRE), \forall\ r \in \mathcal{L} \cup \mathcal{M}$
$M^6 = \max_{r \in \mathcal{L} \cup \mathcal{M}}\{M_r^3\}$

Source: Own authorship.

A.2 Detailed results of the R&F-F&O variants

We report the results of computational experiments carried out with the seven heuristic variants ([$Forward$, $X1$, 3, 1, 2], [$Backward$, $X1$, 3, 1, 2], [$Forward$, $X2$, 3, 1, 2], [$Backward$, $X2$, 3, 1, 2], [$Forward$, $X1$, 4, 1, 3], [$Backward$, $X1$, 4, 1, 3] and [$Forward$, $X2$, 4, 1, 3]) by Tables 33-39. The intention is to present the originating data used by the graphs contained in Figures 18-20. The table structures of this section follow what was established on Table 9.

It is noteworthy to explain how we distributed the runtime so that the heuristics could execute within the time limit of one hour. Knowing that on a MIP-based heuristic, in general, there is a MIP model to be solved at each iteration, something common that happens is the solver gets stuck in one of them (being unable to close the B&C's gap), and may even consume all the remaining time that was dimensioned as a limit, which causes constructive heuristic methods to fail (inasmuch as it is not possible to find a feasible solution if the heuristic does not complete its entire routine). Obviously, if we parameterize that each model to be solved has one hour duration (the total time limit), at the end of procedure, the MIP-based heuristic will probably exceed one hour (mainly at larger instances). To counteract this problem, basically, we initially thought of setting the maximum running time parameter in the solver as: $tr_i = tl - \sum_{i'=0}^{i-1} st_{i'}$, where tr_i is the remaining runtime of iteration i, tl is the total time limit (in this case, one hour), and st_i is the solver's elapsed time for a MIP model at iteration i, being $st_0 = 0$. Nevertheless, some heuristic's iteration could consume all the stipulated time. Then we have arrived at a second alternative, which was to divide tl betwixt the iterations. First we distribute tl in 45% for relax-and-fix and 55% for fix-and-optimize, in such a manner that the average runtime of each iteration is: $tm = tl.perc/n$ ($perc$ is the percentage chosen for R&F or F&O heuristic and n represents the total partitions of R&F or F&O). Afterward, we continue doing: $tr_i = tr_{i-1} - st_{i-1} + tm$ (being $tr_0 = 0$). Note that in this way, tr_i can appropriate the unused time from previous iterations. As tr_i has to be provided at first as a solver parameter, we still have the situation in which R&F heuristic cannot find a feasible solution within tr_i. In this case, we have method failure. Tables 33-39 show this fact through solution status "No Sol. TL" .

Table 33 – Computational results obtained with R&F-F&O[$Forward$, $X1$, 3, 1, 2].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	-	-	-	-	-	-	-	-	-	-	1	1,088.76	-	-	-	-	-	1	1,088.76	No Sol. TL
M5_2to4	555,808.94	12,370.51	299,459.18	139,644.14	1	702.5	45	69	2,411,478.94	21.482%	3	1,635.25	1,007,282.77	0.003%	2	658.25	10.742%	5	2,293.50	Ending
M5_3to5	-	-	-	-	-	-	-	-	-	-	1	1,090.39	-	-	-	-	-	1	1,090.39	No Sol. TL
M5_4to6	-	-	-	-	-	-	-	-	-	-	2	1,633.43	-	-	-	-	-	2	1,633.43	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	1	1,091.95	-	-	-	-	-	1	1,091.95	No Sol. TL
M5_6to8	-	-	-	-	-	-	-	-	-	-	1	1,091.92	-	-	-	-	-	1	1,091.92	No Sol. TL
M5_7to9	605,407.86	24,208.52	0.00	105,797.36	0	529	50	64	1,718,955.48	32.089%	3	1,636.72	735,413.74	3.917%	2	1,981.31	18.003%	5	3,618.03	Time Limit
M5_8to10	638,355.33	36,922.62	0.00	10,399.74	0	52	50	52	691,818.79	0.000%	3	1,139.42	685,677.69	0.000%	2	328.48	0.000%	5	1,467.90	Ending
M5_9to11	655,241.30	22,395.53	0.00	177,293.32	0	897.5	53	65	1,391,956.32	21.407%	3	1,633.37	854,930.15	8.108%	2	1,981.40	14.757%	5	3,614.77	Time Limit
M5_10to12	-	-	-	-	-	-	-	-	-	-	1	1,089.74	-	-	-	-	-	1	1,089.74	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	1	1,090.62	-	-	-	-	-	1	1,090.62	No Sol. TL
M5_12to14	603,220.41	18,971.71	573,429.60	105,909.95	2	533	55	66	3,782,689.70	24.684%	3	1,633.95	1,301,531.67	0.005%	2	802.35	12.344%	5	2,436.30	Ending
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	36	1,696,928.63	0.003%	3	298.43	1,694,978.15	0.000%	2	95.94	0.002%	5	394.37	Ending
M5_14to16	423,780.53	9,665.52	276,889.68	101,797.45	1	509	35	39	812,133.18	0.003%	3	311.99	812,133.18	0.000%	2	117.32	0.002%	5	429.31	Ending
M5_15to17	415,204.19	20,917.05	0.00	115,447.21	0	581	32	47	1,926,240.72	0.360%	3	1,626.94	551,568.45	0.001%	2	250.84	0.180%	5	1,877.78	Ending
Avg M5	**539,095.94**	**19,423.40**	**301,184.03**	**95,736.12**	**1**	**481**	**44**	**55**	**1,804,025.22**	**12.503%**	**2.13**	**1,206.19**	**955,439.47**	**1.504%**	**2**	**776.99**	**7.004%**	**3.20**	**1,620.58**	**-**
M6_1to3	346,756.91	6,532.87	0.00	41,198.97	0	206	32	45	394,488.75	0.000%	3	225.65	394,488.75	0.000%	2	83.11	0.000%	5	308.76	Ending
M6_2to4	240,166.77	10,509.65	0.00	2,199.95	0	11	26	29	257,516.37	0.000%	3	124.50	252,876.37	0.000%	2	52.10	0.000%	5	176.60	Ending
M6_3to5	322,575.51	31,931.25	570,065.40	2,199.95	2	11	28	27	945,023.10	0.000%	3	118.32	926,772.11	0.000%	2	42.76	0.000%	5	161.08	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	23	332,115.23	0.000%	3	50.81	328,431.01	0.000%	2	21.36	0.000%	5	72.17	Ending
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	16	449,164.08	0.000%	3	45.19	449,164.08	0.000%	2	22.04	0.000%	5	67.23	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	12	137,257.46	0.000%	3	28.66	134,864.25	0.000%	2	10.50	0.000%	5	39.16	Ending
M6_7to9	197,684.64	3,973.06	0.00	33,399.16	0	167	12	16	236,523.80	0.000%	3	17.85	235,056.86	0.000%	2	8.18	0.000%	5	26.03	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	80,989.52	0.000%	3	9.54	80,989.52	0.000%	2	4.02	0.000%	5	13.56	Ending
M6_9to11	88,144.05	0.00	970,455.08	11,199.72	3	56	8	6	1,069,798.85	0.000%	3	8.43	1,069,798.85	0.000%	2	3.39	0.000%	5	11.82	Ending
M6_10to12	163,502.88	9,823.05	335,991.60	68,198.30	1	341	8	9	577,515.83	0.000%	3	8.81	577,515.83	0.000%	2	3.55	0.000%	5	12.36	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.15	1	171	3	9	283,953.15	0.000%	3	5.97	283,953.15	0.000%	2	2.63	0.000%	5	8.60	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	405,466.47	0.000%	3	11.12	405,466.47	0.000%	2	4.63	0.000%	5	15.75	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	9	76,158.33	0.000%	3	10.68	76,158.33	0.000%	2	4.87	0.000%	5	15.55	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	43,598.91	0.000%	3	9.24	43,598.91	0.000%	2	3.16	0.000%	5	12.40	Ending
Avg M6	**151,574.81**	**6,458.92**	**188,305.18**	**29,313.55**	**1**	**147**	**13**	**16**	**377,826.42**	**0.000%**	**3.00**	**48.20**	**375,652.46**	**0.000%**	**2**	**19.02**	**0.000%**	**5.00**	**67.22**	**-**

Source: Own authorship.

Table 34 – Computational results obtained with R&F-F&O[*Backward*, *X*1, 3, 1, 2].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	-	-	-	-	-	-	-	-	-	-	1	1,088.79	-	-	-	-	-	1	1,088.79	No Sol. TL
M5_2to4	-	-	-	-	-	-	-	-	-	-	1	1,089.62	-	-	-	-	-	1	1,089.62	No Sol. TL
M5_3to5	-	-	-	-	-	-	-	-	-	-	2	1,635.32	-	-	-	-	-	2	1,635.32	No Sol. TL
M5_4to6	-	-	-	-	-	-	-	-	-	-	1	1,089.55	-	-	-	-	-	1	1,089.55	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	0	546.09	-	-	-	-	-	0	546.09	No Sol. TL
M5_6to8	-	-	-	-	-	-	-	-	-	-	2	1,637.28	-	-	-	-	-	2	1,637.28	No Sol. TL
M5_7to9	-	-	-	-	-	-	-	-	-	-	1	1,091.52	-	-	-	-	-	1	1,091.52	No Sol. TL
M5_8to10	644,428.20	23,842.30	0.00	150,736.33	0	757	54	53	2,696,471.75	1.400%	3	1,408.49	819,006.83	8.865%	2	1,284.64	5.133%	5	2,693.13	Ending
M5_9to11	591,909.55	22,395.53	0.00	116,494.84	0	593.5	53	56	3,953,402.14	2.254%	3	1,209.99	730,799.92	0.004%	2	835.58	1.129%	5	2,045.57	Ending
M5_10to12	-	-	-	-	-	-	-	-	-	-	1	1,090.46	-	-	-	-	-	1	1,090.46	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	1	1,092.10	-	-	-	-	-	1	1,092.10	No Sol. TL
M5_12to14	620,772.68	8,110.88	573,429.60	104,397.39	2	522	55	56	5,199,638.42	0.008%	3	1,469.81	1,306,710.55	0.004%	2	725.27	0.006%	5	2,195.08	Ending
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	49	3,082,166.38	0.003%	3	344.36	1,694,978.15	0.000%	2	147.53	0.001%	5	491.89	Ending
M5_14to16	426,590.96	7,022.96	276,889.68	101,797.46	1	509	33	45	3,407,227.01	0.002%	3	410.02	812,301.06	0.000%	2	193.21	0.001%	5	603.23	Ending
M5_15to17	436,403.67	12,432.10	0.00	113,797.16	0	569	32	46	3,175,093.46	1.537%	3	880.49	562,632.93	0.005%	2	235.06	0.771%	5	1,115.55	Ending
Avg M5	**522,642.33**	**13,956.58**	**351,668.84**	**99,470.49**	**1**	**500**	**44**	**51**	**3,585,666.53**	**0.867%**	**1.87**	**1,072.26**	**987,738.24**	**1.480%**	**2**	**570.22**	**1.173%**	**2.67**	**1,300.35**	**-**
M6_1to3	368,703.10	9,412.67	0.00	41,198.97	0	206	34	42	3,240,499.89	0.203%	3	1,248.42	419,314.74	0.000%	2	138.80	0.101%	5	1,387.22	Ending
M6_2to4	269,439.30	10,815.41	0.00	26,999.32	0	135	27	30	2,374,612.53	0.003%	3	174.54	307,254.03	0.000%	2	59.94	0.002%	5	234.48	Ending
M6_3to5	322,575.51	31,931.25	570,065.40	2,199.95	2	11	28	24	1,036,314.72	0.001%	3	140.96	926,772.11	0.000%	2	49.19	0.001%	5	190.15	Ending
M6_4to6	266,159.05	14,206.49	0.00	51,798.71	0	259	24	22	371,229.93	0.000%	3	63.34	332,164.25	0.000%	2	21.40	0.000%	5	84.74	Ending
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	17	2,084,125.27	0.000%	3	48.87	449,164.08	0.000%	2	22.48	0.000%	5	71.35	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	12	157,527.74	0.000%	3	26.17	134,864.25	0.000%	2	11.50	0.000%	5	37.67	Ending
M6_7to9	225,790.73	0.00	0.00	42,398.94	0	212	13	16	356,923.29	0.000%	3	18.13	268,189.67	0.000%	2	8.01	0.000%	5	26.14	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	126,055.06	0.000%	3	9.02	80,989.52	0.000%	2	4.25	0.000%	5	13.27	Ending
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,041,799.55	0.000%	3	6.63	1,041,799.55	0.000%	2	3.01	0.000%	5	9.64	Ending
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	11	561,693.60	0.000%	3	7.78	561,693.60	0.000%	2	3.98	0.000%	5	11.76	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.14	1	171	3	9	381,582.74	0.000%	3	6.14	283,953.14	0.000%	2	3.04	0.000%	5	9.18	Ending
M6_12to14	31,442.81	653.52	312,838.32	77,798.05	1	389	1	8	426,465.95	0.000%	3	10.35	422,732.70	0.000%	2	5.23	0.000%	5	15.58	Ending
M6_13to15	35,919.42	0.00	0.00	43,598.91	0	218	4	9	79,518.33	0.000%	3	10.40	79,518.33	0.000%	2	4.86	0.000%	5	15.26	Ending
M6_14to16	28,560.00	0.00	0.00	43,598.91	0	218	3	4	72,158.91	0.000%	3	10.07	72,158.91	0.000%	2	4.06	0.000%	5	14.13	Ending
Avg M6	**157,549.49**	**6,402.67**	**188,305.18**	**32,069.01**	**1**	**167**	**14**	**16**	**879,321.96**	**0.015%**	**3.00**	**127.20**	**384,326.35**	**0.000%**	**2**	**24.27**	**0.007%**	**5.00**	**151.47**	**-**

Source: Own authorship.

Table 35 – Computational results obtained with R&F-F&O[$Forward$, $X2$, 3, 1, 2].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	-	-	-	-	-	-	-	-	-	-	2	1,635.04	-	-	-	-	-	2	1,635.04	No Sol. TL
M5_2to4	-	-	-	-	-	-	-	-	-	-	1	1,089.95	-	-	-	-	-	1	1,089.95	No Sol. TL
M5_3to5	593,533.73	31,212.40	0.00	29,399.26	0	147	48	70	29,626,914.74	30.697%	3	1,635.01	654,145.39	0.005%	2	750.22	15.351%	5	2,385.23	Ending
M5_4to6	-	-	-	-	-	-	-	-	-	-	1	1,088.99	-	-	-	-	-	1	1,088.99	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	1	1,093.75	-	-	-	-	-	1	1,093.75	No Sol. TL
M5_6to8	-	-	-	-	-	-	-	-	-	-	2	1,637.41	-	-	-	-	-	2	1,637.41	No Sol. TL
M5_7to9	-	-	-	-	-	-	-	-	-	-	2	1,635.98	-	-	-	-	-	2	1,635.98	No Sol. TL
M5_8to10	701,558.38	48,551.19	0.00	52,198.69	0	261	51	53	1,778,328.26	0.730%	3	1,632.87	802,308.26	0.000%	2	852.89	0.365%	5	2,485.76	Ending
M5_9to11	-	-	-	-	-	-	-	-	-	-	2	1,635.46	-	-	-	-	-	2	1,635.46	No Sol. TL
M5_10to12	-	-	-	-	-	-	-	-	-	-	2	1,635.55	-	-	-	-	-	2	1,635.55	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	1	1,091.27	-	-	-	-	-	1	1,091.27	No Sol. TL
M5_12to14	721,138.74	30,123.41	573,429.60	131,096.72	2	655.5	59	63	33,264,627.44	28.605%	3	1,635.80	1,455,788.47	0.508%	2	1,112.78	14.556%	5	2,748.58	Ending
M5_13to15	435,314.56	11,067.69	1,259,693.76	9,599.76	4	48	36	48	1,919,857.32	0.000%	3	271.56	1,715,675.77	0.004%	2	92.83	0.002%	5	364.39	Ending
M5_14to16	448,793.66	11,760.56	276,889.68	101,797.45	1	509	37	42	2,981,842.32	0.005%	3	713.60	839,241.35	0.000%	2	100.78	0.003%	5	814.38	Ending
M5_15to17	578,753.17	36,751.20	0.00	115,447.21	0	581	34	47	3,094,159.87	4.272%	3	1,628.69	730,951.58	0.009%	2	763.38	2.140%	5	2,392.07	Ending
Avg M5	**579,848.71**	**28,244.41**	**351,668.84**	**73,256.52**	**1**	**367**	**44**	**54**	**12,110,954.99**	**10.718%**	**2.13**	**1,337.40**	**1,033,018.47**	**0.088%**	**2**	**612.15**	**5.403%**	**2.93**	**1,582.25**	**-**
M6_1to3	445,927.66	6,532.87	0.00	41,198.97	0	206	36	42	1,186,144.45	0.002%	3	311.28	493,659.50	0.000%	2	149.54	0.001%	5	460.82	Ending
M6_2to4	247,046.57	7,629.85	0.00	2,199.95	0	11	27	37	443,705.05	0.000%	3	125.73	256,876.37	0.000%	2	72.76	0.000%	5	198.49	Ending
M6_3to5	343,551.08	24,848.88	570,065.40	2,199.95	2	11	29	25	1,235,498.36	0.003%	3	159.09	940,665.31	0.000%	2	68.12	0.001%	5	227.21	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	20	332,248.56	0.000%	3	50.44	328,431.01	0.000%	2	31.32	0.000%	5	81.76	Ending
M6_5to7	207,818.96	6,796.21	227,194.32	11,999.70	1	60	23	18	544,662.65	0.000%	3	50.97	453,809.19	0.000%	2	32.70	0.000%	5	83.67	Ending
M6_6to8	117,509.11	6,315.44	0.00	11,999.70	0	60	11	14	140,783.96	0.000%	3	29.67	135,824.25	0.000%	2	16.65	0.000%	5	46.32	Ending
M6_7to9	182,058.49	0.00	0.00	110,197.25	0	551	10	16	825,510.56	0.000%	3	22.05	292,255.74	0.000%	2	10.93	0.000%	5	32.98	Ending
M6_8to10	153,521.32	0.00	0.00	11,199.72	0	56	9	10	297,915.04	0.000%	3	11.14	164,721.04	0.000%	2	5.97	0.000%	5	17.11	Ending
M6_9to11	62,063.22	1,202.01	970,455.08	11,199.72	3	56	9	5	1,044,920.03	0.000%	3	6.65	1,044,920.03	0.000%	2	4.31	0.000%	5	10.96	Ending
M6_10to12	163,502.88	9,823.05	335,991.60	68,198.30	1	341	8	9	611,776.02	0.000%	3	10.33	577,515.83	0.000%	2	4.44	0.000%	5	14.77	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.15	1	171	3	10	416,763.48	0.000%	3	6.51	283,953.15	0.000%	2	3.92	0.000%	5	10.43	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	405,466.47	0.000%	3	10.83	405,466.47	0.000%	2	7.06	0.000%	5	17.89	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	9	247,020.72	0.000%	3	13.70	76,158.33	0.000%	2	8.26	0.000%	5	21.96	Ending
M6_14to16	11,680.00	0.00	0.00	43,598.91	0	218	1	5	105,116.29	0.000%	3	10.78	55,278.91	0.000%	2	5.53	0.000%	5	16.31	Ending
Avg M6	**164,861.89**	**5,572.02**	**188,305.18**	**34,799.13**	**1**	**174**	**14**	**16**	**559,823.69**	**0.000%**	**3.00**	**58.51**	**393,538.22**	**0.000%**	**2**	**30.11**	**0.000%**	**5.00**	**88.62**	**-**

Source: Own authorship.

Table 36 – Computational results obtained with R&F-F&O[*Backward*, *X*2, 3, 1, 2].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	481,088.00	15,405.12	0.00	149,196.27	0	746	42	61	2,641,643.66	2.435%	3	1,476.14	645,689.39	0.000%	2	979.71	1.217%	5	2,455.85	Ending
M5_2to4	-	-	-	-	-	-	-	-	-	-	1	1,091.27	-	-	-	-	-	1	1,091.27	No Sol. TL
M5_3to5	-	-	-	-	-	-	-	-	-	-	1	1,090.81	-	-	-	-	-	1	1,090.81	No Sol. TL
M5_4to6	535,279.03	30,925.67	320,791.98	60,398.49	1	302	48	59	2,165,014.96	14.791%	3	1,283.92	947,395.17	0.002%	2	788.00	7.397%	5	2,071.92	Ending
M5_5to7	905,253.83	46,503.28	0.00	392,790.18	0	1964	66	68	1,486,271.12	7.492%	3	1,561.12	1,344,547.29	3.440%	2	2,055.63	5.466%	5	3,616.75	Time Limit
M5_6to8	979,476.22	28,081.63	0.00	141,596.46	0	708	62	74	3,240,541.39	5.716%	3	1,202.09	1,149,154.31	0.003%	2	1,029.44	2.860%	5	2,231.53	Ending
M5_7to9	-	-	-	-	-	-	-	-	-	-	0	547.05	-	-	-	-	-	0	547.05	No Sol. TL
M5_8to10	653,081.02	13,426.53	0.00	14,999.63	0	75	53	52	734,440.90	5.973%	3	1,319.76	681,507.18	0.004%	2	196.81	2.988%	5	1,516.57	Ending
M5_9to11	582,843.11	22,395.53	0.00	116,494.84	0	593.5	52	60	775,868.01	1.325%	3	1,552.39	721,733.48	0.005%	2	766.17	0.665%	5	2,318.56	Ending
M5_10to12	-	-	-	-	-	-	-	-	-	-	1	1,090.98	-	-	-	-	-	1	1,090.98	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	1	1,090.89	-	-	-	-	-	1	1,090.89	No Sol. TL
M5_12to14	617,235.41	20,775.43	573,429.60	161,248.23	2	890	54	53	3,492,334.29	0.651%	3	1,635.12	1,372,688.67	0.001%	2	182.07	0.326%	5	1,817.19	Ending
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	42	2,685,327.42	0.000%	3	310.70	1,694,978.15	0.000%	2	90.46	0.000%	5	401.16	Ending
M5_14to16	428,300.69	13,287.50	276,889.68	101,797.45	1	509	37	34	866,215.04	0.000%	3	210.37	820,275.32	0.000%	2	91.94	0.000%	5	302.31	Ending
M5_15to17	535,064.09	17,733.57	0.00	86,397.84	0	432	39	48	1,117,270.88	0.003%	4	381.33	639,195.50	0.000%	2	292.50	0.002%	6	673.83	Ending
Avg M5	**613,337.03**	**21,847.00**	**243,080.50**	**123,451.91**	**1**	**627**	**49**	**55**	**1,920,492.77**	**3.839%**	**2.33**	**1,056.26**	**1,001,716.45**	**0.346%**	**2**	**647.27**	**2.092%**	**3.67**	**1,487.78**	**-**
M6_1to3	342,596.91	6,532.87	0.00	41,198.97	0	206	31	46	478,706.84	0.000%	3	595.22	390,328.75	0.000%	2	143.82	0.000%	5	739.04	Ending
M6_2to4	239,526.77	10,509.65	0.00	2,199.95	0	11	26	30	262,014.26	0.003%	3	145.82	252,236.37	0.000%	2	59.51	0.002%	5	205.33	Ending
M6_3to5	328,975.35	36,138.73	570,065.40	5,362.56	2	34	29	24	992,860.49	0.001%	3	199.71	940,542.04	0.000%	2	66.86	0.001%	5	266.57	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	21	335,497.50	0.000%	3	59.47	328,431.01	0.000%	2	31.96	0.000%	5	91.43	Ending
M6_5to7	204,242.47	7,506.50	227,194.32	11,999.70	1	60	22	17	450,942.99	0.000%	3	56.91	450,942.99	0.000%	2	32.88	0.000%	5	89.79	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	12	134,864.25	0.000%	3	27.80	134,864.25	0.000%	2	16.13	0.000%	5	43.93	Ending
M6_7to9	197,684.64	3,973.06	0.00	33,399.16	0	167	12	14	235,056.86	0.000%	3	19.73	235,056.86	0.000%	2	10.16	0.000%	5	29.89	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	80,989.52	0.000%	3	9.05	80,989.52	0.000%	2	5.39	0.000%	5	14.44	Ending
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,041,799.55	0.000%	3	6.87	1,041,799.55	0.000%	2	3.96	0.000%	5	10.83	Ending
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	10	561,693.60	0.000%	3	7.78	561,693.60	0.000%	2	5.40	0.000%	5	13.18	Ending
M6_11to13	62,202.33	968.61	219,727.84	34,199.14	1	171	5	9	317,097.92	0.000%	3	6.47	317,097.92	0.000%	2	4.00	0.000%	5	10.47	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	405,466.47	0.000%	3	10.72	405,466.47	0.000%	2	7.47	0.000%	5	18.19	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	8	76,158.33	0.000%	3	11.02	76,158.33	0.000%	2	6.36	0.000%	5	17.38	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	43,598.91	0.000%	3	10.29	43,598.91	0.000%	2	4.73	0.000%	5	15.02	Ending
Avg M6	**153,012.54**	**6,901.99**	**188,305.18**	**27,437.89**	**1**	**144**	**13**	**16**	**386,910.54**	**0.000%**	**3.00**	**83.35**	**375,657.61**	**0.000%**	**2**	**28.47**	**0.000%**	**5.00**	**111.82**	**-**

Source: Own authorship.

Table 37 – Computational results obtained with R&F-F&O[$Forward$, $X1$, 4, 1, 3].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	-	-	-	-	-	-	-	-	-	-	2	1,225.02	-	-	-	-	-	2	1,225.02	No Sol. TL
M5_2to4	-	-	-	-	-	-	-	-	-	-	2	1,225.28	-	-	-	-	-	2	1,225.28	No Sol. TL
M5_3to5	-	-	-	-	-	-	-	-	-	-	3	1,635.49	-	-	-	-	-	3	1,635.49	No Sol. TL
M5_4to6	-	-	-	-	-	-	-	-	-	-	2	1,226.47	-	-	-	-	-	2	1,226.47	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	2	1,228.63	-	-	-	-	-	2	1,228.63	No Sol. TL
M5_6to8	717,404.31	31,435.37	0.00	89,197.77	0	446	57	73	867,830.73	1.334%	4	1,573.84	838,037.45	1.412%	2	2,042.05	1.373%	6	3,615.89	Time Limit
M5_7to9	588,617.05	27,497.26	0.00	102,997.43	0	515	49	70	766,987.02	0.004%	4	1,471.93	719,111.74	0.000%	2	1,362.71	0.002%	6	2,834.64	Ending
M5_8to10	666,630.17	40,354.19	0.00	10,399.74	0	52	50	41	743,421.28	0.000%	4	781.30	717,384.10	5.617%	2	2,830.28	2.809%	6	3,611.58	Time Limit
M5_9to11	653,974.80	16,746.65	0.00	109,895.00	0	560.5	53	66	2,281,307.33	16.566%	4	1,632.76	780,616.45	8.143%	2	1,981.56	12.355%	6	3,614.32	Time Limit
M5_10to12	-	-	-	-	-	-	-	-	-	-	2	1,227.17	-	-	-	-	-	2	1,227.17	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	2	1,227.52	-	-	-	-	-	2	1,227.52	No Sol. TL
M5_12to14	615,482.08	9,885.56	573,429.60	104,397.39	2	522	56	70	5,539,526.81	14.283%	4	1,633.95	1,303,194.63	0.004%	2	1,233.37	7.143%	6	2,867.32	Ending
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	36	1,743,513.58	0.002%	4	377.01	1,694,978.15	0.000%	2	150.96	0.001%	6	527.97	Ending
M5_14to16	423,780.53	9,665.52	276,889.68	101,797.46	1	509	35	42	812,893.11	0.000%	4	295.82	812,133.19	0.000%	2	176.29	0.000%	6	472.11	Ending
M5_15to17	429,937.30	13,219.54	0.00	113,797.16	0	569	31	46	596,617.13	0.000%	4	531.77	556,953.99	0.005%	2	585.20	0.002%	6	1,116.97	Ending
Avg M5	**563,946.90**	**19,842.47**	**263,751.63**	**80,260.21**	**1**	**403**	**46**	**56**	**1,669,012.12**	**4.024%**	**3.13**	**1,152.93**	**927,801.21**	**1.897%**	**2**	**1,295.30**	**2.961%**	**4.20**	**1,843.76**	**-**
M6_1to3	353,972.76	8,551.78	0.00	41,198.97	0	206	32	36	432,132.92	0.000%	4	261.83	403,723.51	0.000%	2	107.47	0.000%	6	369.30	Ending
M6_2to4	239,526.77	10,509.65	0.00	2,199.95	0	11	26	31	283,569.42	0.001%	4	128.93	252,236.37	0.000%	2	68.72	0.001%	6	197.65	Ending
M6_3to5	322,575.51	31,931.25	570,065.40	2,199.95	2	11	28	31	955,375.28	0.002%	4	117.60	926,772.11	0.000%	2	54.24	0.001%	6	171.84	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	19	332,115.23	0.000%	4	56.95	328,431.01	0.000%	2	26.57	0.000%	6	83.52	Ending
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	18	449,164.08	0.000%	4	52.10	449,164.08	0.000%	2	23.30	0.000%	6	75.40	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	13	137,257.46	0.000%	4	32.61	134,864.25	0.000%	2	13.77	0.000%	6	46.38	Ending
M6_7to9	197,684.64	3,973.06	0.00	33,399.17	0	167	12	16	235,056.87	0.000%	4	20.02	235,056.87	0.000%	2	9.00	0.000%	6	29.02	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	80,989.52	0.000%	4	10.93	80,989.52	0.000%	2	4.67	0.000%	6	15.60	Ending
M6_9to11	88,144.05	0.00	970,455.08	11,199.72	3	56	8	6	1,069,798.85	0.000%	4	10.60	1,069,798.85	0.000%	2	3.52	0.000%	6	14.12	Ending
M6_10to12	163,502.88	9,823.05	335,991.60	68,198.30	1	341	8	9	609,383.95	0.000%	4	8.66	577,515.83	0.000%	2	4.30	0.000%	6	12.96	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.15	1	171	3	9	283,953.15	0.000%	4	6.87	283,953.15	0.000%	2	3.43	0.000%	6	10.30	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	405,466.47	0.000%	4	12.72	405,466.47	0.000%	2	5.61	0.000%	6	18.33	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	8	76,158.33	0.000%	4	12.31	76,158.33	0.000%	2	5.53	0.000%	6	17.84	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	43,598.91	0.000%	4	10.84	43,598.91	0.000%	2	4.31	0.000%	6	15.15	Ending
Avg M6	**152,044.51**	**6,603.13**	**188,305.18**	**29,313.55**	**1**	**147**	**13**	**16**	**385,287.17**	**0.000%**	**4**	**53.07**	**376,266.37**	**0.000%**	**2**	**23.89**	**0.000%**	**6**	**76.96**	**-**

Source: Own authorship.

Table 38 – Computational results obtained with R&F-F&O[$Backward$, $X1$, 4, 1, 3].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	Cf	Cup	$Cout$	$Cover$	$nOut$	$nOver$	nF	$nRest$	OF_{ub}	$Avg\ gap$	$iter$	$CPUt$	OF_{ub}	$Avg\ gap$	$iter$	$CPUt$	$Avg\ gap$	$iter$	$CPUt$	St. Sol.
M5_1to3	616,434.58	13,038.06	0.00	128,796.78	0	644	44	56	6,877,725.57	0.002%	4	1,492.79	758,269.42	13.682%	2	1,516.42	6.842%	6	3,009.21	Ending
M5_2to4	-	-	-	-	-	-	-	-	-	-	2	1,227.19	-	-	-	-	-	2	1,227.19	No Sol. TL
M5_3to5	760,174.04	47,132.19	936,936.18	57,919.12	3	338	50	65	9,584,090.97	33.811%	4	1,636.98	1,802,161.53	38.609%	2	1,984.97	36.210%	6	3,621.95	Time Limit
M5_4to6	-	-	-	-	-	-	-	-	-	-	2	1,229.61	-	-	-	-	-	2	1,229.61	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	1	818.33	-	-	-	-	-	1	818.33	No Sol. TL
M5_6to8	-	-	-	-	-	-	-	-	-	-	2	1,229.01	-	-	-	-	-	2	1,229.01	No Sol. TL
M5_7to9	-	-	-	-	-	-	-	-	-	-	1	819.21	-	-	-	-	-	1	819.21	No Sol. TL
M5_8to10	671,089.05	20,619.70	0.00	10,399.74	0	52	50	45	6,169,534.30	0.005%	4	1,328.04	702,108.49	5.962%	2	1,361.60	2.984%	6	2,689.64	Ending
M5_9to11	683,120.82	16,746.65	0.00	119,694.76	0	609.5	56	65	8,170,540.75	16.483%	4	1,633.62	819,562.23	17.616%	2	1,512.24	17.049%	6	3,145.86	Ending
M5_10to12	-	-	-	-	-	-	-	-	-	-	2	1,226.76	-	-	-	-	-	2	1,226.76	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	2	1,233.14	-	-	-	-	-	2	1,233.14	No Sol. TL
M5_12to14	616,047.29	20,967.28	573,429.60	107,397.39	2	547	58	60	6,368,283.07	0.208%	4	1,633.42	1,317,841.56	9.042%	2	1,075.24	4.625%	6	2,708.66	Ending
M5_13to15	415,748.93	9,935.70	1,259,693.76	9,599.76	4	48	34	57	4,631,359.02	0.005%	4	358.89	1,694,978.15	0.000%	2	191.42	0.003%	6	550.31	Ending
M5_14to16	423,780.53	9,665.52	276,889.68	101,797.45	1	509	35	45	2,327,609.62	0.002%	4	360.32	812,133.18	0.000%	2	164.80	0.001%	6	525.12	Ending
M5_15to17	415,204.19	20,917.05	0.00	115,447.22	0	581	32	47	3,981,358.78	0.003%	4	699.63	551,568.45	0.004%	2	507.72	0.003%	6	1,207.35	Ending
Avg M5	**575,199.93**	**19,877.77**	**380,868.65**	**81,381.53**	**1**	**416**	**45**	**55**	**6,013,812.76**	**6.315%**	**2.93**	**1,128.46**	**1,057,327.88**	**10.614%**	**2**	**1,039.30**	**8.465%**	**4.00**	**1,682.76**	**-**
M6_1to3	384,762.29	14,775.55	0.00	70,073.26	0	351	32	40	4,433,169.85	0.002%	4	358.80	469,611.10	0.000%	2	114.47	0.001%	6	473.27	Ending
M6_2to4	342,670.99	1,096.98	0.00	7,799.81	0	39	29	34	3,545,762.89	0.000%	4	168.27	351,567.78	0.000%	2	90.82	0.000%	6	259.09	Ending
M6_3to5	337,009.45	29,565.07	570,065.40	2,199.94	2	11	31	31	3,857,957.98	0.001%	4	150.02	938,839.86	0.000%	2	78.46	0.000%	6	228.48	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	22	542,426.92	0.001%	4	72.70	328,431.01	0.000%	2	29.33	0.000%	6	102.03	Ending
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	19	2,076,805.56	0.000%	4	59.24	449,164.08	0.000%	2	25.77	0.000%	6	85.01	Ending
M6_6to8	116,549.11	6,315.44	0.00	11,999.70	0	60	11	13	150,301.38	0.000%	4	31.31	134,864.25	0.000%	2	13.89	0.000%	6	45.20	Ending
M6_7to9	219,524.22	0.00	0.00	51,398.72	0	257	12	16	365,200.60	0.000%	4	20.76	270,922.94	0.000%	2	9.44	0.000%	6	30.20	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	128,455.06	0.000%	4	10.77	80,989.52	0.000%	2	5.17	0.000%	6	15.94	Ending
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,071,212.21	0.000%	4	7.91	1,041,799.55	0.000%	2	3.65	0.000%	6	11.56	Ending
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	11	663,182.60	0.000%	4	9.68	561,693.60	0.000%	2	4.74	0.000%	6	14.42	Ending
M6_11to13	39,546.16	0.00	219,727.84	34,199.14	1	171	4	9	409,742.29	0.000%	4	8.30	293,473.14	0.000%	2	3.51	0.000%	6	11.81	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	9	441,279.42	0.000%	4	12.85	405,466.47	0.000%	2	5.94	0.000%	6	18.79	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	9	90,984.71	0.000%	4	12.96	76,158.33	0.000%	2	5.82	0.000%	6	18.78	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	66,158.91	0.000%	4	12.38	43,598.91	0.000%	2	4.96	0.000%	6	17.34	Ending
Avg M6	**163,853.63**	**5,922.55**	**188,305.18**	**30,960.11**	**1**	**161**	**14**	**17**	**1,274,474.31**	**0.000%**	**4.00**	**66.85**	**389,041.47**	**0.000%**	**2**	**28.28**	**0.000%**	**6.00**	**95.14**	**-**

Source: Own authorship.

Table 39 – Computational results obtained with R&F-F&O[$Forward$, $X2$, 4, 1, 3].

	Operational Costs				Counters				R&F Part				F&O Part				R&F-F&O			
Instance	*Cf*	*Cup*	*Cout*	*Cover*	*nOut*	*nOver*	*nF*	*nRest*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	OF_{ub}	*Avg gap*	*iter*	*CPUt*	*Avg gap*	*iter*	*CPUt*	St. Sol.
M5_1to3	617,536.26	24,793.73	0.00	150,646.32	0	756	42	63	20,558,684.69	19.106%	4	1,633.97	792,976.31	4.818%	2	1,373.47	11.962%	6	3,007.44	Ending
M5_2to4	-	-	-	-	-	-	-	-	-	-	2	1,226.76	-	-	-	-	-	2	1,226.76	No Sol. TL
M5_3to5	-	-	-	-	-	-	-	-	-	-	3	1,634.17	-	-	-	-	-	3	1,634.17	No Sol. TL
M5_4to6	-	-	-	-	-	-	-	-	-	-	2	1,225.99	-	-	-	-	-	2	1,225.99	No Sol. TL
M5_5to7	-	-	-	-	-	-	-	-	-	-	2	1,227.74	-	-	-	-	-	2	1,227.74	No Sol. TL
M5_6to8	-	-	-	-	-	-	-	-	-	-	3	1,636.97	-	-	-	-	-	3	1,636.97	No Sol. TL
M5_7to9	-	-	-	-	-	-	-	-	-	-	2	1,228.49	-	-	-	-	-	2	1,228.49	No Sol. TL
M5_8to10	-	-	-	-	-	-	-	-	-	-	2	1,224.18	-	-	-	-	-	2	1,224.18	No Sol. TL
M5_9to11	-	-	-	-	-	-	-	-	-	-	2	1,229.66	-	-	-	-	-	2	1,229.66	No Sol. TL
M5_10to12	-	-	-	-	-	-	-	-	-	-	3	1,636.76	-	-	-	-	-	3	1,636.76	No Sol. TL
M5_11to13	-	-	-	-	-	-	-	-	-	-	2	1,227.82	-	-	-	-	-	2	1,227.82	No Sol. TL
M5_12to14	651,582.51	16,224.78	573,429.60	104,397.39	2	522	60	57	6,837,578.34	0.005%	4	1,508.11	1,345,634.28	0.003%	2	862.25	0.004%	6	2,370.36	Ending
M5_13to15	503,859.36	13,851.47	1,259,693.76	9,599.76	4	48	36	45	5,667,159.98	0.005%	4	562.26	1,787,004.35	0.004%	2	209.65	0.005%	6	771.91	Ending
M5_14to16	429,060.30	14,746.28	276,889.68	101,797.45	1	509	35	39	2,904,221.72	0.005%	4	1,009.64	822,493.71	0.003%	2	167.16	0.004%	6	1,176.80	Ending
M5_15to17	542,185.41	41,624.27	0.00	86,397.84	0	432	33	44	5,695,010.05	0.001%	4	1,481.86	670,207.52	0.005%	2	858.85	0.003%	6	2,340.71	Ending
Avg M5	**548,844.77**	**22,248.11**	**422,002.61**	**90,567.75**	**1**	**453**	**41**	**50**	**8,332,530.95**	**3.824%**	**2.87**	**1,312.96**	**1,083,663.23**	**0.967%**	**2**	**694.28**	**2.395%**	**3.53**	**1,544.38**	**-**
M6_1to3	366,062.99	10,471.61	0.00	41,198.97	0	206	32	49	1,732,460.94	0.000%	4	123.18	417,733.57	0.000%	2	208.28	0.000%	6	331.46	Ending
M6_2to4	242,993.35	10,509.65	0.00	2,199.94	0	11	27	29	1,064,366.69	0.002%	4	100.40	255,702.94	0.000%	2	82.06	0.001%	6	182.46	Ending
M6_3to5	332,793.53	26,490.67	570,065.40	2,199.95	2	11	31	30	1,353,062.81	0.000%	4	72.09	931,549.55	0.003%	2	108.49	0.002%	6	180.58	Ending
M6_4to6	262,425.81	14,206.49	0.00	51,798.71	0	259	23	20	410,064.79	0.000%	4	21.93	328,431.01	0.000%	2	40.57	0.000%	6	62.50	Ending
M6_5to7	203,490.52	6,479.54	227,194.32	11,999.70	1	60	23	17	634,926.11	0.000%	4	24.93	449,164.08	0.000%	2	35.70	0.000%	6	60.63	Ending
M6_6to8	117,509.11	6,315.44	0.00	11,999.70	0	60	11	12	175,453.09	0.000%	4	14.44	135,824.25	0.000%	2	19.84	0.000%	6	34.28	Ending
M6_7to9	197,684.64	3,973.06	0.00	33,399.16	0	167	12	13	748,380.28	0.000%	4	12.25	235,056.86	0.000%	2	12.41	0.000%	6	24.66	Ending
M6_8to10	69,789.80	0.00	0.00	11,199.72	0	56	7	8	692,842.04	0.000%	4	6.99	80,989.52	0.000%	2	6.36	0.000%	6	13.35	Ending
M6_9to11	60,144.75	0.00	970,455.08	11,199.72	3	56	8	5	1,044,920.03	0.000%	4	4.55	1,041,799.55	0.000%	2	4.74	0.000%	6	9.29	Ending
M6_10to12	177,102.54	9,823.05	335,991.60	38,776.41	1	282	8	10	832,952.58	0.000%	4	6.71	561,693.60	0.000%	2	6.70	0.000%	6	13.41	Ending
M6_11to13	30,026.16	0.00	219,727.84	34,199.14	1	171	3	10	531,014.02	0.000%	4	6.17	283,953.14	0.000%	2	4.22	0.000%	6	10.39	Ending
M6_12to14	48,375.72	653.52	312,838.32	43,598.91	1	218	3	10	769,655.47	0.000%	4	6.54	405,466.47	0.000%	2	7.44	0.000%	6	13.98	Ending
M6_13to15	32,559.42	0.00	0.00	43,598.91	0	218	2	9	680,183.94	0.002%	4	10.03	76,158.33	0.000%	2	7.41	0.001%	6	17.44	Ending
M6_14to16	0.00	0.00	0.00	43,598.91	0	218	0	5	136,743.91	0.000%	4	5.99	43,598.91	0.000%	2	5.68	0.000%	6	11.67	Ending
Avg M6	**152,925.60**	**6,351.65**	**188,305.18**	**27,211.99**	**1**	**142**	**14**	**16**	**771,930.48**	**0.000%**	**4.00**	**29.73**	**374,794.41**	**0.000%**	**2**	**39.28**	**0.000%**	**6.00**	**69.01**	**-**

Source: Own authorship.

APPENDIX B

Supplementary content of Chapter 4

In this appendix, we report the additional content of Chapter 4. The algorithms of the six duty cases presented in Subsection 4.2.1 are shown in B.1. Furthermore, in B.2, we exhibit the results per instance obtained by the B&P variants (defined in Subsection 4.6.1).

B.1 Algorithms of the six duty cases

Algorithms 8-13 detail how the resources are processed for each case rule (Subsection 4.2.1) during forward label extensions. They all start by dimensioning $\overline{W}(\ell_s)$, taking case variant *var* as a reference. Given the time imposition of an arc (r, s), $\overline{W}(\ell_s)$ is checked to ascertain if it remains in the time window required by a request s, otherwise, the algorithms return a failure situation ($\mathit{Extension} = 0$), demonstrating that label ℓ_s is infeasible and it can be discarded. Next, the procedures analyze whether all flight events are covered by any pilot time window, pushing $\overline{W}(\ell_s)$ to the nearest window, if necessary. Supposing impossibility, the routines close with $\mathit{Extension} = 0$ over again. Once the time windows have been validated, the next step determines the other resources. As previously discussed, algorithms 9-12 reset $\overline{U}(\ell_s)$, $\overline{Q}(\ell_s)$ and $\overline{aGT}(\ell_s)$ because they have to insert the minimum rest, calculating overtime when the crews pass their standard working hours.

Algorithm 8: *CheckDutyCase1* (Continued on next page)

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$, $checkFtw1$ and $checkFtw2$ be binary variables to check time widows. While $checkTW$ indicates whether the live leg is inside its window and the pilot's window, $checkFtw1$ and $checkFtw2$ inform whether a given ferry is covered by any pilot window, starting earlier or later, respectively;

Let $\overline{\mathcal{O}}$ be the overtime worked on the arc, and $Cover_r$ be overtime cost paid per minute for request r;

$\overline{var}(\ell_s) \leftarrow var$;

`// The requests have an earlier or a later start`

$$tw \leftarrow \begin{cases} st_s, & var = 1a; \\ st_s + \Delta_{\mathcal{L}}, & var = 1b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 1b \wedge s \in \mathcal{M}. \end{cases}$$

`// Serviced time of the requests`

$$\overline{W}(\ell_s) \leftarrow \max \left\{ tw, \begin{cases} av_v + PRE + tat^s_{i^s}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v = i^s; \\ av_v, & r = 0 \wedge s \in \mathcal{M} \wedge k^v = i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TL_r + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TL_r, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ av_v + PRE + TF^{\check{p}^v}_{k^v i^s} + tat^s_{i^s}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v \neq i^s; \\ av_v + PRE + TF^{\check{p}^v}_{k^v i^s}, & r = 0 \wedge s \in \mathcal{M} \wedge k^v \neq i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases} \right\};$$

`// Time window of the requests`

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0$;

`// Time window of the pilots in chief`

Verify if all live and ferry legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. Indicate whether a ferry starts earlier or later using variables $checkFtw1$ and $checkFtw2$. If the attempt fails, **return** $Extension \leftarrow 0$;

Algorithm 8: *CheckDutyCase1* (Continued)

`// Determine the other resources`

$$\overline{aGT}(\ell_s) \leftarrow \begin{cases} [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{W}(\ell_s) - (av_v + PRE + TF^{\check{p}^v}_{k^v i^s}), & r = 0 \wedge s \in \mathcal{M} \wedge k^v \neq i^s \\ & \wedge\, checkFtw1 = 1; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \\ & \wedge\, checkFtw1 = 1; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TL_r + TF^{\check{p}^v}_{j^r i^s}), & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \\ & \wedge\, checkFtw1 = 1; \\ 0, & r = 0 \wedge s \in \mathcal{M} \wedge k^v \neq i^s \\ & \wedge\, checkFtw1 = 0 \wedge checkFtw2 = 1; \\ 0, & r \in \mathcal{L} \cup \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \\ & \wedge\, checkFtw1 = 0 \wedge checkFtw2 = 1. \end{cases}$$

$$\overline{U}(\ell_s) \leftarrow \begin{cases} PRE + tat^s_{i^s}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v = i^s; \\ PRE, & r = 0 \wedge s \in \mathcal{M} \wedge k^v = i^s; \\ \overline{U}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s \wedge \overline{firstM}(\ell_r) = 0; \\ PRE + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s \wedge \overline{firstM}(\ell_r) = 1; \\ \overline{U}(\ell_r), & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ PRE + TF^{\check{p}^v}_{k^v i^s} + tat^s_{i^s}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v \neq i^s; \\ PRE + TF^{\check{p}^v}_{k^v i^s}, & r = 0 \wedge s \in \mathcal{M} \wedge k^v \neq i^s; \\ \overline{U}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \wedge checkFtw1 = 1; \\ \overline{U}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \\ & \wedge\, checkFtw1 = 0 \wedge checkFtw2 = 1; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 0; \\ PRE + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 1; \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \\ & \wedge\, (\overline{firstM}(\ell_r) = 0 \vee checkFtw1 = 1); \\ \overline{U}(\ell_r) + \overline{aGT}(\ell_r) + [\overline{W}(\ell_s) - \overline{W}(\ell_r)], & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \wedge [\overline{firstM}(\ell_r) = 0 \\ & \vee\, (checkFtw1 = 0 \wedge checkFtw2 = 1)]; \\ PRE + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 1. \end{cases}$$

$$\overline{Q}(\ell_s) \leftarrow \begin{cases} 0, & r = 0 \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge k^v = i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ \overline{Q}(\ell_r), & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r = i^s; \\ TF^{\check{p}^v}_{k^v i^s}, & r = 0 \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge k^v \neq i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

$$\overline{firstM}(\ell_s) \leftarrow \begin{cases} 1, & \text{if } r = 0 \wedge s \in \mathcal{M} \wedge k^v = i^s; \\ \overline{firstM}(\ell_r), & \text{else if } r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r = i^s; \\ 0, & \text{otherwise.} \end{cases}$$

`// Calculate the overtime cost`

$\overline{\mathcal{O}} \leftarrow \max\left\{0, \overline{U}(\ell_s) - maxDuty, \overline{Q}(\ell_s) - maxFlying\right\}$;

$$\overline{\mathcal{CO}}(\ell_s) \leftarrow \begin{cases} Cover_r.\overline{\mathcal{O}}, & \text{if } s = R + 1 \wedge \{r \in \mathcal{L} \vee [r \in \mathcal{M} \\ & \wedge\, (\overline{var}(\ell_r) = 1a \vee \overline{var}(\ell_r) = 1b \vee \overline{var}(\ell_r) = 6a \vee \overline{var}(\ell_r) = 6b)]\}; \\ 0, & \text{otherwise.} \end{cases}$$

return *Extension* $\leftarrow (s \in \mathcal{M} \cup \{R+1\}) \vee (\overline{\mathcal{CO}}(\ell_s) = 0)$;

Algorithm 9: *CheckDutyCase2*

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$ be a binary variable that indicates whether the live leg is inside its window and the pilot's window;

Let $RestM$ be the amount of rest to be inserted, with a view to utilizing ground time/maintenance;

Let $\overline{\mathcal{O}}$ be the overtime worked on the arc, and $Cover_r$ be overtime cost paid per minute for request r;

if $r > 0 \wedge s \in \mathcal{L} \wedge j^r = i^s \wedge \overline{firstM}(\ell_r) = 0$ **then**

$\overline{var}(\ell_s) \leftarrow var;$

`// The requests have an earlier or a later start`

$$tw \leftarrow \begin{cases} st_s, & var = 2a; \\ st_s + \Delta_{\mathcal{L}}, & var = 2b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 2b \wedge s \in \mathcal{M}. \end{cases}$$

`// Serviced time of the requests`

$$RestM \leftarrow \max\left\{0, (POS + minRest + PRE) - (\overline{aGT}(\ell_r) + TL_r)\right\} \mid r \in \mathcal{M};$$

$$\overline{W}(\ell_s) \leftarrow \max\left\{ tw, \begin{cases} \overline{W}(\ell_r) + TF^{\tilde{p}^v}_{i^r j^r} \\ + (POS + minRest + PRE) + tat^s_{i^s} \end{cases}, \quad r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TL_r + RestM + tat^s_{i^s}, \quad r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s. \right\};$$

`// Time window of the requests`

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0;$

`// Time window of the pilots in chief`

Verify if all live legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. If the attempt fails, **return** $Extension \leftarrow 0;$

`// Determine the other resources`

$\overline{U}(\ell_s) \leftarrow PRE + tat^s_{i^s};$

$\overline{aGT}(\ell_s) \leftarrow 0;$

$\overline{Q}(\ell_s) \leftarrow 0;$

`// Calculate the overtime cost`

$$\overline{\mathcal{O}} \leftarrow \begin{cases} \max\begin{Bmatrix} 0, \overline{U}(\ell_r) + TF^{\tilde{p}^v}_{i^r j^r} - maxDuty, \\ \overline{Q}(\ell_r) + TF^{\tilde{p}^v}_{i^r j^r} - maxFlying \end{Bmatrix}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \max\left\{0, \overline{U}(\ell_r) - maxDuty, \overline{Q}(\ell_r) - maxFlying\right\}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s. \end{cases}$$

$\overline{C\mathcal{O}}(\ell_s) \leftarrow Cover_r . \overline{\mathcal{O}};$

return $Extension \leftarrow 1$; `// return successfully`

else

return $Extension \leftarrow 0;$

Algorithm 10: *CheckDutyCase3*

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$ be a binary variable that indicates whether the live leg is inside its window and the pilot's window;

Let $RestM$ be the amount of rest to be inserted, with a view to utilizing ground time/maintenance;

Let $\overline{\mathcal{O}}$ be the overtime worked on the arc, and $Cover_r$ be overtime cost paid per minute for request r;

if $r > 0 \wedge s < R+1 \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 0$ **then**

$\overline{var}(\ell_s) \leftarrow var$;

// The requests have an earlier or a later start

$$tw \leftarrow \begin{cases} st_s, & var = 3a; \\ st_s + \Delta_{\mathcal{L}}, & var = 3b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 3b \wedge s \in \mathcal{M}. \end{cases}$$

// Serviced time of the requests

$RestM \leftarrow \max\left\{0, (POS + minRest + PRE) - (\overline{aGT}(\ell_r) + TL_r)\right\} \mid r \in \mathcal{M}$;

$$\overline{W}(\ell_s) \leftarrow \max\left\{ tw, \begin{cases} \begin{matrix}\overline{W}(\ell_r) + TF^{\breve{p}^v}_{i^r j^r} + (POS + minRest + PRE) \\ + tat^s_{j^r} + TF^{\breve{p}^v}_{j^r i^s} + tat^s_{i^s}\end{matrix}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \begin{matrix}\overline{W}(\ell_r) + TF^{\breve{p}^v}_{i^r j^r} + (POS + minRest + PRE) \\ + tat^s_{j^r} + TF^{\breve{p}^v}_{j^r i^s}\end{matrix}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + RestM + TF^{\breve{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + RestM + TF^{\breve{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases} \right\};$$

// Time window of the requests

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0$;

// Time window of the pilots in chief

Verify if all live and ferry legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. If the attempt fails, **return** $Extension \leftarrow 0$;

// Determine the other resources

$$\overline{U}(\ell_s) \leftarrow \begin{cases} PRE + tat^s_{j^r} + TF^{\breve{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ PRE + tat^s_{j^r} + TF^{\breve{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{M} \wedge j^r \neq i^s; \\ PRE + TF^{\breve{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ PRE + TF^{\breve{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

$\overline{aGT}(\ell_s) \leftarrow 0$;

$\overline{Q}(\ell_s) \leftarrow TF^{\breve{p}^v}_{j^r i^s}$;

// Calculate the overtime cost

$$\overline{\mathcal{O}} \leftarrow \begin{cases} \max\begin{Bmatrix} 0, \overline{U}(\ell_r) + TF^{\breve{p}^v}_{i^r j^r} - maxDuty, \\ \overline{Q}(\ell_r) + TF^{\breve{p}^v}_{i^r j^r} - maxFlying \end{Bmatrix}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \max\left\{0, \overline{U}(\ell_r) - maxDuty, \overline{Q}(\ell_r) - maxFlying\right\}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s. \end{cases}$$

$\overline{\mathcal{CO}}(\ell_s) \leftarrow Cover_r.\overline{\mathcal{O}}$;

return $Extension \leftarrow 1$; // return successfully

else

return $Extension \leftarrow 0$;

Algorithm 11: *CheckDutyCase4*

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$ be a binary variable that indicates whether the live leg is inside its window and the pilot's window;

Let $\overline{\mathcal{O}}$ be the overtime worked on the arc, and $Cover_r$ be overtime cost paid per minute for request r;

if $s \in \mathcal{L} \wedge [(r = 0 \wedge k^v \neq i^s) \vee (r > 0 \wedge j^r \neq i^s)]$ **then**

$\overline{var}(\ell_s) \leftarrow var;$

`// The requests have an earlier or a later start`

$$tw \leftarrow \begin{cases} st_s, & var = 4a; \\ st_s + \Delta_{\mathcal{L}}, & var = 4b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 4b \wedge s \in \mathcal{M}. \end{cases}$$

`// Serviced time of the requests`

$$\overline{W}(\ell_s) \leftarrow \max \left\{ tw, \begin{cases} av_v + PRE + TF^{\check{p}^v}_{k^v i^s} + (POS + minRest + PRE) + tat^s_{i^s}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v \neq i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + (POS + minRest + PRE) + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s} + (POS + minRest + PRE) + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s. \end{cases} \right\};$$

`// Time window of the requests`

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0;$

`// Time window of the pilots in chief`

Verify if all live and ferry legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. If the attempt fails, **return** $Extension \leftarrow 0;$

`// Determine the other resources`

$\overline{U}(\ell_s) \leftarrow PRE + tat^s_{i^s};$

$\overline{aGT}(\ell_s) \leftarrow 0;$

$\overline{Q}(\ell_s) \leftarrow 0;$

$\overline{firstM}(\ell_s) \leftarrow 0;$

`// Calculate the overtime cost`

$$\overline{\mathcal{O}} \leftarrow \begin{cases} \max \begin{Bmatrix} 0, PRE + TF^{\check{p}^v}_{k^v i^s} - maxDuty, \\ TF^{\check{p}^v}_{k^v i^s} - maxFlying \end{Bmatrix}, & r = 0 \wedge s \in \mathcal{L} \wedge k^v \neq i^s; \\ \max \begin{Bmatrix} 0, \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} - maxDuty, \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s} - maxFlying \end{Bmatrix}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \max \begin{Bmatrix} 0, \overline{aGT}(\ell_r) + \overline{U}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s} - maxDuty, \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{j^r i^s} - maxFlying \end{Bmatrix}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 0; \\ \max \begin{Bmatrix} 0, PRE + TF^{\check{p}^v}_{j^r i^s} - maxDuty, \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{j^r i^s} - maxFlying \end{Bmatrix}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 1. \end{cases}$$

$\overline{\mathcal{CO}}(\ell_s) \leftarrow Cover_r.\overline{\mathcal{O}};$

return $Extension \leftarrow 1;$ `// return successfully`

else

return $Extension \leftarrow 0;$

Algorithm 12: *CheckDutyCase5*

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$ be a binary variable that indicates whether the live leg is inside its window and the pilot's window;

Let $RestM$ be the amount of rest to be inserted, with a view to utilizing ground time/maintenance;

Let $\overline{\mathcal{O}}$ be the overtime worked on the arc, and $Cover_r$ be overtime cost paid per minute for request r;

if $r > 0 \wedge s \in \mathcal{L} \wedge j^r \neq i^s \wedge \overline{firstM}(\ell_r) = 0$ **then**

$\overline{var}(\ell_s) \leftarrow var$;

`// The requests have an earlier or a later start`

$$tw \leftarrow \begin{cases} st_s, & var = 5a; \\ st_s + \Delta_{\mathcal{L}}, & var = 5b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 5b \wedge s \in \mathcal{M}. \end{cases}$$

`// Serviced time of the requests`

$RestM \leftarrow \max\left\{0, (POS + minRest + PRE) - (\overline{aGT}(\ell_r) + TL_r)\right\} \mid r \in \mathcal{M}$;

$$\overline{W}(\ell_s) \leftarrow \max \left\{ tw, \begin{cases} \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s} \\ + 2.(POS + minRest + PRE) \end{cases}, \quad r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \quad\quad \begin{cases} \overline{W}(\ell_r) + TL_r + RestM + TF^{\check{p}^v}_{j^r i^s} \\ + (POS + minRest + PRE) + tat^s_{i^s} \end{cases}, \quad r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s. \right\};$$

`// Time window of the requests`

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0$;

`// Time window of the pilots in chief`

Verify if all live and ferry legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. If the attempt fails, **return** $Extension \leftarrow 0$;

`// Determine the other resources`

$\overline{U}(\ell_s) \leftarrow PRE + tat^s_{i^s}$;

$\overline{aGT}(\ell_s) \leftarrow 0$;

$\overline{Q}(\ell_s) \leftarrow 0$;

`// Calculate the overtime cost`

$$\overline{\mathcal{O}} \leftarrow \begin{cases} \max \begin{Bmatrix} \max\left\{0, \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} - maxDuty\right\} \\ + \max\left\{0, tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} - maxDuty\right\}, \\ \max\left\{0, \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} - maxFlying\right\} \\ + \max\left\{0, TF^{\check{p}^v}_{j^r i^s} - maxFlying\right\} \end{Bmatrix}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \max \begin{Bmatrix} \max\left\{0, \overline{U}(\ell_r) - maxDuty\right\} \\ + \max\left\{0, TF^{\check{p}^v}_{j^r i^s} - maxDuty\right\}, \\ \max\left\{0, \overline{Q}(\ell_r) - maxFlying\right\} \\ + \max\left\{0, TF^{\check{p}^v}_{j^r i^s} - maxFlying\right\} \end{Bmatrix}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s. \end{cases}$$

$\overline{\mathcal{CO}}(\ell_s) \leftarrow Cover_r.\overline{\mathcal{O}}$;

return $Extension \leftarrow 1$; `// return successfully`

else

return $Extension \leftarrow 0$;

Algorithm 13: *CheckDutyCase6*

Input: problem instance, [case variant var, parent label ℓ_r, child label ℓ_s].

Let tw and $Extension$ be auxiliary variables;

Let $checkTW$ be a binary variable that indicates whether the live leg is inside its window and the pilot's window;

Let GT, B and $newGT$ be the current ground time already considering previous $\overline{aGT}(\ell_r$, a variable that classifies which time range GT is in, and the new ground time that can be obtained due to the reduction in duty;

if $r > 0 \wedge s < R+1 \wedge \{(s \in \mathcal{L} \wedge j^r = i^s) \vee [\neg(r \in \mathcal{L} \wedge s \in \mathcal{M}) \wedge j^r \neq i^s]\} \wedge \overline{firstM}(\ell_r) = 0$ **then**

$\overline{var}(\ell_s) \leftarrow var$;

`// The requests have an earlier or a later start`

$$tw \leftarrow \begin{cases} st_s, & var = 6a; \\ st_s + \Delta_{\mathcal{L}}, & var = 6b \wedge s \in \mathcal{L}; \\ st_s + \Delta_{\mathcal{M}}, & var = 6b \wedge s \in \mathcal{M}. \end{cases}$$

`// Serviced time of the requests`

$$\overline{W}(\ell_s) \leftarrow \max \left\{ tw, \begin{cases} \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TL_r + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{W}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{W}(\ell_r) + TL_r + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases} \right\};$$

`// Time window of the requests`

$$checkTW \leftarrow \begin{cases} \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{L}}, & s \in \mathcal{L}; \\ \overline{W}(\ell_s) \leq st_s + \Delta_{\mathcal{M}}, & s \in \mathcal{M}. \end{cases}$$

if $\neg checkTW$ **then return** $Extension \leftarrow 0$;

`// Time window of the pilots in chief`

Verify if all live and ferry legs are covered by any pilot's time window. If not, try to bring the respective flight event to a nearest adjacent window. If the attempt fails, **return** $Extension \leftarrow 0$;

`// Check in which time range the cost is`

$$GT \leftarrow \begin{cases} [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] + \overline{aGT}(\ell_r) - tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] - (TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}), & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] + \overline{aGT}(\ell_r) - (TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}), & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ [\overline{W}(\ell_s) - \overline{W}(\ell_r)] + \overline{aGT}(\ell_r) - TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

$$B \leftarrow \begin{cases} 1, & GT \leq 90; \\ 2, & 90 < GT \leq 360; \\ 3, & 360 < GT \leq minRest; \\ 4, & GT > minRest. \end{cases}$$

if $B = 2 \vee B = 3$ **then**

$$newGT \leftarrow \begin{cases} (GT - 90)/2 + 90, & B = 2; \\ 60, & B = 3. \end{cases}$$

`// Determine the other resources`

$$\overline{U}(\ell_s) \leftarrow \begin{cases} \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + newGT + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + newGT + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{U}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + tat^s_{j^r} + TF^{\check{p}^v}_{j^r i^s} + newGT + tat^s_{i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + newGT + TF^{\check{p}^v}_{j^r i^s} + tat^s_{i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{U}(\ell_r) + newGT + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

$\overline{aGT}(\ell_s) \leftarrow 0$;

$$\overline{Q}(\ell_s) \leftarrow \begin{cases} \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{Q}(\ell_r), & r \in \mathcal{M} \wedge s \in \mathcal{L} \wedge j^r = i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{i^r j^r} + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{L} \wedge s \in \mathcal{L} \wedge j^r \neq i^s; \\ \overline{Q}(\ell_r) + TF^{\check{p}^v}_{j^r i^s}, & r \in \mathcal{M} \wedge s \in \mathcal{L} \cup \mathcal{M} \wedge j^r \neq i^s. \end{cases}$$

`// Calculate the overtime cost`

$\overline{\mathcal{CO}}(\ell_s) \leftarrow 0$;

return $Extension \leftarrow 1$; `// return successfully`

else

return $Extension \leftarrow 0$;

else

return $Extension \leftarrow 0$;

B.2 Detailed results of the B&P variants

Tables 40 to 43 show the results of computational experiments carried out with real-life instances defined in Section 3.4, which were solved by the B&P variants presented in Subsection 4.6.1, denoted B&P$[0,0]$, B&P$[1,0]$, B&P$[0,1]$ and B&P$[1,1]$, where the first term inside the brackets indicates whether the approach uses two-step (0) or strong branching (1), and the second, if the primal heuristic is off (0) or on (1). These tables have a similar layout to Table 10, however, columns OF_{lb} to $CPUt$ depict values for each instance (first column), not only average values for each month/B&P variant (as seen in Table 10). In the first column, each instance is in the format Mx_$y$to$z$, where x indicates the month, and y and z represent the first and last day considered in the month. Finally, at the end of each month, we calculate the average of each column ("Avg Mx").

Table 40 – Computational results obtained with B&P$[0, 0]$.

Instance	OF_{lb}	OF_{ub}	gap	$nCol$	$nNode$	$RMPt$	SPt	$CPUt$
M1_1to3	68,274.49	68,274.49	0.000%	304	1	0.045	0.686	0.771
M1_2to4	103,236.18	103,236.18	0.000%	446	2	0.089	1.070	1.214
M1_3to5	84,257.96	84,257.96	0.000%	747	5	0.255	3.128	3.521
M1_4to6	63,677.10	63,677.10	0.000%	659	6	0.268	4.590	5.058
M1_5to7	71,632.73	71,632.73	0.000%	439	2	0.102	1.648	1.832
M1_6to8	72,926.87	72,926.87	0.000%	281	4	0.081	1.731	1.935
M1_7to9	61,871.80	61,871.80	0.000%	266	1	0.039	0.381	0.448
M1_8to10	71,937.82	71,937.82	0.000%	186	1	0.027	0.354	0.408
Avg M1	**74,726.87**	**74,726.87**	**0.000%**	**416**	**2.75**	**0.113**	**1.699**	**1.898**
M2_1to3	102,880.69	102,880.69	0.000%	172	1	0.027	0.443	0.498
M2_2to4	126,331.91	126,331.91	0.000%	209	2	0.044	1.019	1.132
M2_3to5	126,357.94	126,357.94	0.000%	351	1	0.048	0.584	0.667
M2_4to6	232,142.37	232,142.37	0.000%	426	1	0.059	1.171	1.289
M2_5to7	158,376.22	158,376.22	0.000%	526	3	0.137	2.915	3.193
M2_6to8	241,022.34	241,022.34	0.000%	392	2	0.084	1.589	1.753
M2_7to9	130,213.49	130,213.49	0.000%	177	2	0.037	1.023	1.129
M2_8to10	200,979.16	200,979.16	0.000%	283	1	0.040	0.708	0.792
Avg M2	**164,788.01**	**164,788.01**	**0.000%**	**317**	**1.63**	**0.060**	**1.182**	**1.307**
M3_1to3	243,932.79	243,932.79	0.000%	922	9	0.469	16.292	17.084
M3_2to4	520,499.87	520,499.87	0.000%	766	3	0.186	5.028	5.356
M3_3to5	204,764.81	204,764.81	0.000%	565	1	0.069	1.509	1.628
M3_4to6	526,115.69	526,115.69	0.000%	406	4	0.085	2.024	2.233
M3_5to7	218,272.61	218,272.61	0.000%	490	8	0.154	3.035	3.374
M3_6to8	39,247.12	39,247.12	0.000%	417	2	0.071	0.780	0.894
Avg M3	**292,138.81**	**292,138.81**	**0.000%**	**594.33**	**4.50**	**0.172**	**4.778**	**5.095**
M4_1to3	462,133.33	462,133.33	0.000%	2,290	27	3.347	203.479	208.673
M4_2to4	264,423.30	264,423.30	0.000%	2,602	27	3.431	160.338	165.149
M4_3to5	237,273.06	237,273.06	0.000%	2,401	45	4.774	286.713	294.288
M4_4to6	246,859.20	246,859.20	0.000%	1,942	14	2.470	144.016	147.620
M4_5to7	388,373.53	388,373.53	0.000%	1,517	13	1.846	132.822	135.580
M4_6to8	320,275.47	320,275.47	0.000%	1,630	45	3.344	249.154	254.556
M4_7to9	302,686.88	302,686.88	0.000%	1,699	15	1.640	126.142	128.929
M4_8to10	332,778.61	332,778.61	0.000%	1,684	27	1.866	149.897	153.183
M4_9to11	703,498.25	703,498.25	0.000%	2,010	10	1.546	99.899	102.286
M4_10to12	484,849.25	484,849.25	0.000%	1,853	12	2.361	183.780	187.247
M4_11to13	491,144.00	491,144.00	0.000%	1,987	7	2.679	173.999	177.608
M4_12to14	666,998.55	666,998.55	0.000%	1,806	4	1.544	95.445	97.658
M4_13to15	593,957.84	593,957.84	0.000%	1,593	10	2.032	114.015	116.962
M4_14to16	318,407.95	318,407.95	0.000%	1,602	6	1.068	55.977	57.573
Avg M4	**415,261.37**	**415,261.37**	**0.000%**	**1,901.14**	**18.71**	**2.425**	**155.405**	**159.094**
M5_1to3	630,747.54	630,747.54	0.000%	1,779	6	2.022	175.830	178.796
M5_2to4	970,635.74	970,635.74	0.000%	2,246	11	3.916	339.166	344.404
M5_3to5	599,584.81	599,584.81	0.000%	2,194	8	2.965	294.430	298.456
M5_4to6	879,487.21	879,487.21	0.000%	2,316	24	4.519	392.508	399.450
M5_5to7	926,710.66	926,710.66	0.000%	2,474	27	6.655	568.722	578.253
M5_6to8	838,037.56	838,037.56	0.000%	1,759	20	3.126	361.532	366.704
M5_7to9	719,111.83	719,111.83	0.000%	1,770	8	2.334	223.265	226.729
M5_8to10	661,246.56	661,246.56	0.000%	1,919	35	4.530	411.685	419.262
M5_9to11	718,975.36	718,975.36	0.000%	2,118	55	7.811	819.002	831.402
M5_10to12	1,297,831.26	1,297,831.26	0.000%	2,546	83	12.700	1,583.242	1,602.817
M5_11to13	1,063,134.42	1,063,134.42	0.000%	2,176	14	4.018	339.761	345.385
M5_12to14	1,298,091.85	1,298,091.85	0.000%	1,265	7	1.070	112.093	114.067
M5_13to15	1,694,978.23	1,694,978.23	0.000%	1,286	7	0.756	59.273	60.666
M5_14to16	812,133.22	812,133.22	0.000%	1,288	6	0.994	72.269	73.880
M5_15to17	548,931.56	548,931.56	0.000%	1,518	8	0.941	66.421	67.988
Avg M5	**910,642.52**	**910,642.52**	**0.000%**	**1,910.27**	**21.27**	**3.890**	**387.947**	**393.884**
M6_1to3	390,328.77	390,328.77	0.000%	1,771	10	1.181	79.875	81.751
M6_2to4	252,236.40	252,236.40	0.000%	1,357	22	1.129	76.236	78.520
M6_3to5	926,772.21	926,772.21	0.000%	1,593	20	1.242	62.537	64.617
M6_4to6	328,431.04	328,431.04	0.000%	1,168	12	0.781	27.190	28.583
M6_5to7	449,164.10	449,164.10	0.000%	1,420	12	0.961	29.122	30.601
M6_6to8	134,864.27	134,864.27	0.000%	1,263	5	0.420	9.463	10.119
M6_7to9	235,056.88	235,056.88	0.000%	871	8	0.329	7.453	8.036
M6_8to10	80,989.52	80,989.52	0.000%	496	2	0.078	1.705	1.859
M6_9to11	1,041,799.60	1,041,799.60	0.000%	484	1	0.054	0.799	0.895
M6_10to12	561,693.63	561,693.63	0.000%	498	1	0.070	0.976	1.082
M6_11to13	283,953.27	283,953.27	0.000%	567	1	0.065	0.760	0.860
M6_12to14	405,466.52	405,466.52	0.000%	705	3	0.143	2.814	3.065
M6_13to15	76,158.34	76,158.34	0.000%	785	6	0.208	4.659	5.085
M6_14to16	43,598.92	43,598.92	0.000%	722	4	0.137	3.127	3.398
Avg M6	**372,179.53**	**372,179.53**	**0.000%**	**978.57**	**7.64**	**0.486**	**21.908**	**22.748**

Source: Own authorship.

Table 41 – Computational results obtained with B&P$[1, 0]$.

Instance	OF_{lb}	OF_{ub}	gap	$nCol$	$nNode$	$RMPt$	SPt	$CPUt$
M1_1to3	68,274.49	68,274.49	0.000%	304	1	0.049	0.685	0.768
M1_2to4	103,236.18	103,236.18	0.000%	463	2	0.088	1.050	1.223
M1_3to5	84,257.95	84,257.95	0.000%	1403	7	0.383	4.607	5.576
M1_4to6	63,677.10	63,677.10	0.000%	645	6	0.250	4.440	5.337
M1_5to7	71,632.73	71,632.73	0.000%	439	2	0.104	1.650	1.862
M1_6to8	72,926.87	72,926.87	0.000%	275	2	0.043	0.864	1.004
M1_7to9	61,871.80	61,871.80	0.000%	266	1	0.035	0.373	0.435
M1_8to10	71,937.82	71,937.82	0.000%	186	1	0.027	0.364	0.415
Avg M1	**74,726.87**	**74,726.87**	**0.000%**	**497.63**	**2.75**	**0.122**	**1.754**	**2.078**
M2_1to3	102,880.69	102,880.69	0.000%	172	1	0.023	0.419	0.475
M2_2to4	126,331.91	126,331.91	0.000%	213	2	0.044	0.998	1.116
M2_3to5	126,357.94	126,357.94	0.000%	351	1	0.047	0.582	0.663
M2_4to6	232,142.37	232,142.37	0.000%	426	1	0.061	1.180	1.293
M2_5to7	158,376.22	158,376.22	0.000%	526	3	0.137	2.915	3.259
M2_6to8	241,022.34	241,022.34	0.000%	707	2	0.112	1.751	1.991
M2_7to9	130,213.49	130,213.49	0.000%	177	2	0.041	1.042	1.167
M2_8to10	200,979.16	200,979.16	0.000%	283	1	0.038	0.695	0.775
Avg M2	**164,788.01**	**164,788.01**	**0.000%**	**356.88**	**1.63**	**0.063**	**1.198**	**1.342**
M3_1to3	243,932.79	243,932.79	0.000%	915	6	0.401	12.568	13.852
M3_2to4	520,499.86	520,499.86	0.000%	767	3	0.181	4.910	5.322
M3_3to5	204,764.81	204,764.81	0.000%	565	1	0.074	1.511	1.634
M3_4to6	526,115.69	526,115.69	0.000%	407	3	0.071	1.576	1.796
M3_5to7	218,272.61	218,272.61	0.000%	490	6	0.119	2.390	2.903
M3_6to8	39,247.12	39,247.12	0.000%	791	4	0.140	1.568	1.882
Avg M3	**292,138.81**	**292,138.81**	**0.000%**	**655.83**	**3.83**	**0.164**	**4.087**	**4.565**
M4_1to3	462,133.30	462,133.30	0.000%	2,206	28	3.698	215.670	229.661
M4_2to4	264,423.30	264,423.30	0.000%	2,440	35	4.311	203.651	220.226
M4_3to5	237,273.08	237,273.08	0.000%	2,366	36	4.679	238.658	256.177
M4_4to6	246,859.20	246,859.20	0.000%	1,973	23	3.412	202.066	213.524
M4_5to7	388,373.56	388,373.56	0.000%	1,573	22	2.402	198.775	207.262
M4_6to8	320,275.47	320,275.47	0.000%	6,202	33	4.442	262.419	275.387
M4_7to9	302,686.88	302,686.88	0.000%	1,683	8	1.172	82.321	85.459
M4_8to10	332,778.58	332,778.58	0.000%	1,650	19	1.680	113.529	120.116
M4_9to11	703,498.19	703,498.19	0.000%	2,004	9	1.534	95.297	100.029
M4_10to12	484,849.25	484,849.25	0.000%	1,852	13	2.584	192.139	199.555
M4_11to13	491,144.00	491,144.00	0.000%	2,060	19	4.015	290.738	301.836
M4_12to14	666,998.55	666,998.55	0.000%	1,808	4	1.521	93.533	95.961
M4_13to15	593,957.89	593,957.89	0.000%	1,584	6	1.556	76.424	79.504
M4_14to16	318,407.96	318,407.96	0.000%	1,610	4	0.867	42.324	44.025
Avg M4	**415,261.37**	**415,261.37**	**0.000%**	**2,215.07**	**18.50**	**2.705**	**164.825**	**173.480**
M5_1to3	630,747.54	630,747.54	0.000%	1,851	7	2.296	187.789	193.429
M5_2to4	970,635.74	970,635.74	0.000%	2,309	11	4.124	353.434	362.677
M5_3to5	599,584.76	599,584.76	0.000%	2,206	10	3.204	317.271	324.580
M5_4to6	879,487.14	879,487.14	0.000%	4,275	24	6.343	489.082	505.874
M5_5to7	926,710.60	926,710.60	0.000%	2,443	20	5.623	456.374	472.000
M5_6to8	838,037.56	838,037.56	0.000%	1,771	11	2.285	252.690	258.709
M5_7to9	719,111.83	719,111.83	0.000%	1,758	9	2.356	229.304	234.921
M5_8to10	661,246.61	661,246.61	0.000%	1,766	30	3.925	348.476	363.282
M5_9to11	718,975.30	718,975.30	0.000%	2,170	56	7.435	817.764	847.408
M5_10to12	1,297,831.26	1,297,831.26	0.000%	2,148	11	3.428	308.836	318.476
M5_11to13	1,063,134.42	1,063,134.42	0.000%	2,125	7	2.871	228.014	234.594
M5_12to14	1,298,091.82	1,298,091.82	0.000%	1,273	7	1.149	118.425	121.880
M5_13to15	1,694,978.35	1,694,978.35	0.000%	1,333	8	0.835	62.857	65.678
M5_14to16	812,133.22	812,133.22	0.000%	1,251	9	1.113	87.939	91.343
M5_15to17	548,931.52	548,931.52	0.000%	1,548	9	1.050	76.165	79.286
Avg M5	**910,642.51**	**910,642.51**	**0.000%**	**2,015.13**	**15.27**	**3.202**	**288.961**	**298.276**
M6_1to3	390,328.77	390,328.77	0.000%	1,806	13	1.386	94.819	99.631
M6_2to4	252,236.40	252,236.40	0.000%	1,487	32	2.071	117.576	126.657
M6_3to5	926,772.15	926,772.15	0.000%	1,530	15	1.073	52.347	56.725
M6_4to6	328,431.05	328,431.05	0.000%	1,207	9	0.661	22.393	24.511
M6_5to7	449,164.10	449,164.10	0.000%	1,403	7	0.670	21.092	22.916
M6_6to8	134,864.26	134,864.26	0.000%	1,303	7	0.622	12.770	14.860
M6_7to9	235,056.89	235,056.89	0.000%	1,910	5	0.472	6.425	7.617
M6_8to10	80,989.52	80,989.52	0.000%	496	2	0.081	1.680	1.856
M6_9to11	1,041,799.60	1,041,799.60	0.000%	484	1	0.053	0.779	0.873
M6_10to12	561,693.63	561,693.63	0.000%	498	1	0.073	0.973	1.082
M6_11to13	283,953.27	283,953.27	0.000%	567	1	0.067	0.769	0.867
M6_12to14	405,466.49	405,466.49	0.000%	708	3	0.158	3.047	3.345
M6_13to15	76,158.34	76,158.34	0.000%	784	6	0.210	4.669	5.336
M6_14to16	43,598.91	43,598.91	0.000%	724	3	0.131	2.726	3.024
Avg M6	**372,179.53**	**372,179.53**	**0.000%**	**1,064.79**	**7.50**	**0.552**	**24.433**	**26.379**

Source: Own authorship.

Table 42 – Computational results obtained with B&P$[0, 1]$.

Instance	OF_{lb}	OF_{ub}	gap	$nCol$	$nNode$	$RMPt$	SPt	$CPUt$
M1_1to3	68,274.49	68,274.49	0.000%	304	1	0.045	0.680	0.761
M1_2to4	103,236.18	103,236.18	0.000%	442	1	0.060	0.587	0.698
M1_3to5	84,257.95	84,257.95	0.000%	730	1	0.126	0.983	1.163
M1_4to6	63,674.08	63,677.10	0.005%	619	0	0.111	1.171	1.362
M1_5to7	71,632.73	71,632.73	0.000%	431	1	0.077	0.954	1.093
M1_6to8	72,926.86	72,926.86	0.000%	275	1	0.034	0.493	0.572
M1_7to9	61,871.80	61,871.80	0.000%	266	1	0.035	0.376	0.437
M1_8to10	71,937.82	71,937.82	0.000%	186	1	0.024	0.351	0.404
Avg M1	**74,726.49**	**74,726.87**	**0.001%**	**406.63**	**0.88**	**0.064**	**0.699**	**0.811**
M2_1to3	102,880.69	102,880.69	0.000%	172	1	0.023	0.420	0.476
M2_2to4	126,331.90	126,331.90	0.000%	207	1	0.029	0.530	0.606
M2_3to5	126,357.94	126,357.94	0.000%	351	1	0.047	0.587	0.671
M2_4to6	232,142.36	232,142.36	0.000%	426	1	0.061	1.162	1.299
M2_5to7	158,376.20	158,376.20	0.000%	524	1	0.087	1.370	1.529
M2_6to8	241,022.33	241,022.33	0.000%	391	1	0.062	0.936	1.066
M2_7to9	130,213.48	130,213.48	0.000%	177	1	0.032	0.599	0.677
M2_8to10	200,979.16	200,979.16	0.000%	283	1	0.037	0.690	0.770
Avg M2	**164,788.01**	**164,788.01**	**0.000%**	**316.38**	**1**	**0.047**	**0.787**	**0.887**
M3_1to3	243,932.78	243,932.78	0.000%	892	1	0.167	4.206	4.464
M3_2to4	520,499.80	520,499.80	0.000%	765	1	0.120	2.795	2.999
M3_3to5	204,764.81	204,764.81	0.000%	565	1	0.074	1.526	1.647
M3_4to6	526,115.63	526,115.63	0.000%	406	1	0.046	0.666	0.761
M3_5to7	218,272.58	218,272.58	0.000%	488	1	0.057	0.597	0.702
M3_6to8	39,247.11	39,247.11	0.000%	417	1	0.058	0.498	0.595
Avg M3	**292,138.78**	**292,138.78**	**0.000%**	**588.83**	**1**	**0.087**	**1.715**	**1.861**
M4_1to3	462,130.74	462,130.80	0.001%	2,084	1	0.886	38.616	39.895
M4_2to4	264,423.28	264,423.28	0.000%	2,295	1	1.176	33.047	34.520
M4_3to5	237,273.05	237,273.05	0.000%	2,286	1	1.153	39.835	41.270
M4_4to6	246,859.19	246,859.19	0.000%	1,916	1	1.374	59.086	60.815
M4_5to7	388,368.90	388,373.52	0.001%	1,470	0	0.933	45.991	47.245
M4_6to8	320,275.46	320,275.46	0.000%	1,469	1	0.409	24.106	24.800
M4_7to9	302,686.85	302,686.85	0.000%	1,652	1	0.544	30.342	31.195
M4_8to10	332,778.57	332,778.57	0.000%	1,624	1	0.502	25.518	26.264
M4_9to11	703,498.16	703,498.16	0.000%	1,951	1	0.769	37.143	38.230
M4_10to12	484,849.23	484,849.23	0.000%	1,761	1	1.292	77.498	79.161
M4_11to13	491,143.94	491,143.94	0.000%	1,957	1	1.869	98.893	101.166
M4_12to14	666,998.52	666,998.52	0.000%	1,792	1	1.191	64.710	66.276
M4_13to15	593,957.81	593,957.81	0.000%	1,576	1	1.206	45.070	46.636
M4_14to16	318,399.66	318,408.46	0.003%	1,585	1	0.548	23.362	24.212
Avg M4	**415,260.24**	**415,261.42**	**0.000%**	**1,815.57**	**0.79**	**0.989**	**45.944**	**47.263**
M5_1to3	630,747.51	630,747.51	0.000%	1,756	1	1.421	90.412	92.262
M5_2to4	970,635.70	970,635.70	0.000%	2,144	1	2.601	148.121	151.206
M5_3to5	599,584.74	599,584.74	0.000%	2,125	1	1.933	152.759	155.155
M5_4to6	879,487.10	879,487.10	0.000%	2,253	1	2.277	132.212	134.974
M5_5to7	926,710.56	926,710.56	0.000%	2,391	1	3.215	157.414	161.187
M5_6to8	838,037.45	838,037.45	0.000%	1,722	1	1.353	100.926	102.824
M5_7to9	719,111.74	719,111.74	0.000%	1,695	1	1.397	103.242	105.210
M5_8to10	661,246.54	661,246.54	0.000%	1,706	1	1.317	72.929	74.684
M5_9to11	718,975.27	718,975.27	0.000%	1,797	1	1.521	99.820	101.786
M5_10to12	1,297,791.28	1,297,831.20	0.003%	2,109	0	2.007	142.395	144.979
M5_11to13	1,063,110.36	1,063,134.92	0.002%	2,063	1	1.950	127.547	130.026
M5_12to14	1,298,091.67	1,298,091.67	0.000%	1,247	1	0.664	54.830	55.922
M5_13to15	1,694,978.15	1,694,978.15	0.000%	1,273	1	0.369	19.295	19.984
M5_14to16	812,133.19	812,133.19	0.000%	1,227	1	0.592	32.490	33.397
M5_15to17	548,931.50	548,931.50	0.000%	1,512	1	0.500	29.671	30.439
Avg M5	**910,638.18**	**910,642.48**	**0.000%**	**1,801.33**	**0.87**	**1.541**	**97.604**	**99.602**
M6_1to3	390,328.75	390,328.75	0.000%	1,672	1	0.495	23.741	24.481
M6_2to4	252,236.37	252,236.37	0.000%	1,269	1	0.297	13.278	13.744
M6_3to5	926,772.11	926,772.11	0.000%	1,399	1	0.354	13.336	13.850
M6_4to6	328,431.01	328,431.01	0.000%	1,149	1	0.302	7.283	7.708
M6_5to7	449,164.08	449,164.08	0.000%	1,362	1	0.370	8.986	9.483
M6_6to8	134,864.25	134,864.25	0.000%	1,229	1	0.214	4.087	4.399
M6_7to9	235,056.87	235,056.87	0.000%	861	1	0.136	2.060	2.270
M6_8to10	80,989.52	80,989.52	0.000%	493	1	0.052	0.953	1.062
M6_9to11	1,041,799.60	1,041,799.60	0.000%	484	1	0.055	0.784	0.880
M6_10to12	561,693.63	561,693.63	0.000%	498	1	0.073	0.957	1.070
M6_11to13	283,953.27	283,953.27	0.000%	567	1	0.067	0.753	0.851
M6_12to14	405,466.47	405,466.47	0.000%	703	1	0.084	1.368	1.520
M6_13to15	76,158.33	76,158.33	0.000%	784	1	0.108	1.447	1.625
M6_14to16	43,598.91	43,598.91	0.000%	722	1	0.087	1.330	1.481
Avg M6	**372,179.51**	**372,179.51**	**0.000%**	**942.29**	**1**	**0.192**	**5.740**	**6.030**

Source: Own authorship.

Table 43 – Computational results obtained with B&P[1, 1].

Instance	OF_{lb}	OF_{ub}	gap	$nCol$	$nNode$	$RMPt$	SPt	$CPUt$
M1_1to3	68,274.49	68,274.49	0.000%	304	1	0.045	0.691	0.776
M1_2to4	103,236.18	103,236.18	0.000%	442	1	0.066	0.598	0.747
M1_3to5	84,257.95	84,257.95	0.000%	730	1	0.123	0.976	1.284
M1_4to6	63,674.08	63,677.10	0.005%	619	0	0.105	1.185	1.482
M1_5to7	71,632.73	71,632.73	0.000%	431	1	0.075	0.951	1.124
M1_6to8	72,926.86	72,926.86	0.000%	275	1	0.034	0.503	0.624
M1_7to9	61,871.80	61,871.80	0.000%	266	1	0.039	0.382	0.448
M1_8to10	71,937.82	71,937.82	0.000%	186	1	0.024	0.354	0.409
Avg M1	**74,726.49**	**74,726.87**	**0.001%**	**406.63**	**0.88**	**0.064**	**0.705**	**0.862**
M2_1to3	102,880.69	102,880.69	0.000%	172	1	0.027	0.442	0.505
M2_2to4	126,331.90	126,331.90	0.000%	207	1	0.029	0.529	0.614
M2_3to5	126,357.94	126,357.94	0.000%	351	1	0.054	0.596	0.689
M2_4to6	232,142.36	232,142.36	0.000%	426	1	0.068	1.196	1.334
M2_5to7	158,376.20	158,376.20	0.000%	524	1	0.085	1.365	1.590
M2_6to8	241,022.33	241,022.33	0.000%	391	1	0.061	0.921	1.102
M2_7to9	130,213.48	130,213.48	0.000%	177	1	0.032	0.601	0.700
M2_8to10	200,979.16	200,979.16	0.000%	283	1	0.036	0.704	0.786
Avg M2	**164,788.01**	**164,788.01**	**0.000%**	**316.38**	**1**	**0.049**	**0.794**	**0.915**
M3_1to3	243,932.78	243,932.78	0.000%	892	1	0.167	4.275	4.714
M3_2to4	520,499.80	520,499.80	0.000%	765	1	0.120	2.801	3.073
M3_3to5	204,764.81	204,764.81	0.000%	565	1	0.070	1.519	1.646
M3_4to6	526,115.63	526,115.63	0.000%	406	1	0.047	0.680	0.823
M3_5to7	218,272.58	218,272.58	0.000%	488	1	0.056	0.600	0.790
M3_6to8	39,247.11	39,247.11	0.000%	417	1	0.062	0.513	0.651
Avg M3	**292,138.78**	**292,138.78**	**0.000%**	**588.83**	**1**	**0.087**	**1.731**	**1.950**
M4_1to3	462,130.64	462,133.70	0.001%	2,084	1	0.890	38.789	40.508
M4_2to4	264,423.28	264,423.28	0.000%	2,295	1	1.174	33.284	35.216
M4_3to5	237,273.05	237,273.05	0.000%	2,286	1	1.157	40.577	42.528
M4_4to6	246,859.19	246,859.19	0.000%	1,916	1	1.355	59.273	61.421
M4_5to7	388,368.90	388,373.52	0.001%	1,470	0	0.928	46.426	48.068
M4_6to8	320,275.46	320,275.46	0.000%	1,469	1	0.410	24.253	25.247
M4_7to9	302,686.85	302,686.85	0.000%	1,652	1	0.549	31.444	32.678
M4_8to10	332,778.57	332,778.57	0.000%	1,624	1	0.491	25.866	26.912
M4_9to11	703,498.16	703,498.16	0.000%	1,951	1	0.776	37.859	39.388
M4_10to12	484,849.23	484,849.23	0.000%	1,761	1	1.297	78.997	81.139
M4_11to13	491,143.94	491,143.94	0.000%	1,957	1	1.872	100.714	103.507
M4_12to14	666,998.52	666,998.52	0.000%	1,792	1	1.195	66.625	68.453
M4_13to15	593,957.81	593,957.81	0.000%	1,576	1	1.211	46.671	48.609
M4_14to16	318,399.46	318,408.26	0.003%	1,585	1	0.559	23.961	25.162
Avg M4	**415,260.22**	**415,261.39**	**0.000%**	**1,815.57**	**0.79**	**0.990**	**46.767**	**48.488**
M5_1to3	630,747.51	630,747.51	0.000%	1,756	1	1.411	92.008	94.382
M5_2to4	970,635.70	970,635.70	0.000%	2,144	1	2.610	150.753	154.456
M5_3to5	599,584.74	599,584.74	0.000%	2,125	1	1.939	155.445	158.419
M5_4to6	879,487.10	879,487.10	0.000%	2,253	1	2.253	133.929	137.270
M5_5to7	926,710.56	926,710.56	0.000%	2,391	1	3.163	159.155	163.545
M5_6to8	838,037.45	838,037.45	0.000%	1,722	1	1.341	103.979	106.361
M5_7to9	719,111.74	719,111.74	0.000%	1,695	1	1.401	106.007	108.445
M5_8to10	661,246.54	661,246.54	0.000%	1,706	1	1.335	75.109	77.317
M5_9to11	718,975.27	718,975.27	0.000%	1,797	1	1.497	101.131	103.556
M5_10to12	1,297,791.28	1,297,831.20	0.003%	2,109	0	1.998	144.016	147.190
M5_11to13	1,063,110.06	1,063,134.62	0.002%	2,063	1	1.959	130.112	133.147
M5_12to14	1,298,091.67	1,298,091.67	0.000%	1,247	1	0.668	56.445	57.924
M5_13to15	1,694,978.15	1,694,978.15	0.000%	1,273	1	0.371	20.053	21.038
M5_14to16	812,133.19	812,133.19	0.000%	1,227	1	0.594	33.627	34.832
M5_15to17	548,931.50	548,931.50	0.000%	1,512	1	0.500	30.032	31.117
Avg M5	**910,638.16**	**910,642.46**	**0.000%**	**1,801.33**	**0.87**	**1.536**	**99.453**	**101.933**
M6_1to3	390,328.75	390,328.75	0.000%	1,672	1	0.493	24.362	25.451
M6_2to4	252,236.37	252,236.37	0.000%	1,269	1	0.291	13.099	13.776
M6_3to5	926,772.11	926,772.11	0.000%	1,399	1	0.357	13.392	14.163
M6_4to6	328,431.01	328,431.01	0.000%	1,149	1	0.303	7.329	7.960
M6_5to7	449,164.08	449,164.08	0.000%	1,362	1	0.366	8.967	9.698
M6_6to8	134,864.25	134,864.25	0.000%	1,229	1	0.214	4.116	4.647
M6_7to9	235,056.87	235,056.87	0.000%	861	1	0.135	2.081	2.432
M6_8to10	80,989.52	80,989.52	0.000%	493	1	0.054	0.970	1.111
M6_9to11	1,041,799.60	1,041,799.60	0.000%	484	1	0.054	0.799	0.895
M6_10to12	561,693.63	561,693.63	0.000%	498	1	0.073	0.989	1.102
M6_11to13	283,953.27	283,953.27	0.000%	567	1	0.066	0.756	0.852
M6_12to14	405,466.47	405,466.47	0.000%	703	1	0.086	1.371	1.544
M6_13to15	76,158.33	76,158.33	0.000%	784	1	0.104	1.455	1.726
M6_14to16	43,598.91	43,598.91	0.000%	722	1	0.085	1.320	1.526
Avg M6	**372,179.51**	**372,179.51**	**0.000%**	**942.29**	**1**	**0.192**	**5.786**	**6.206**

Source: Own authorship.

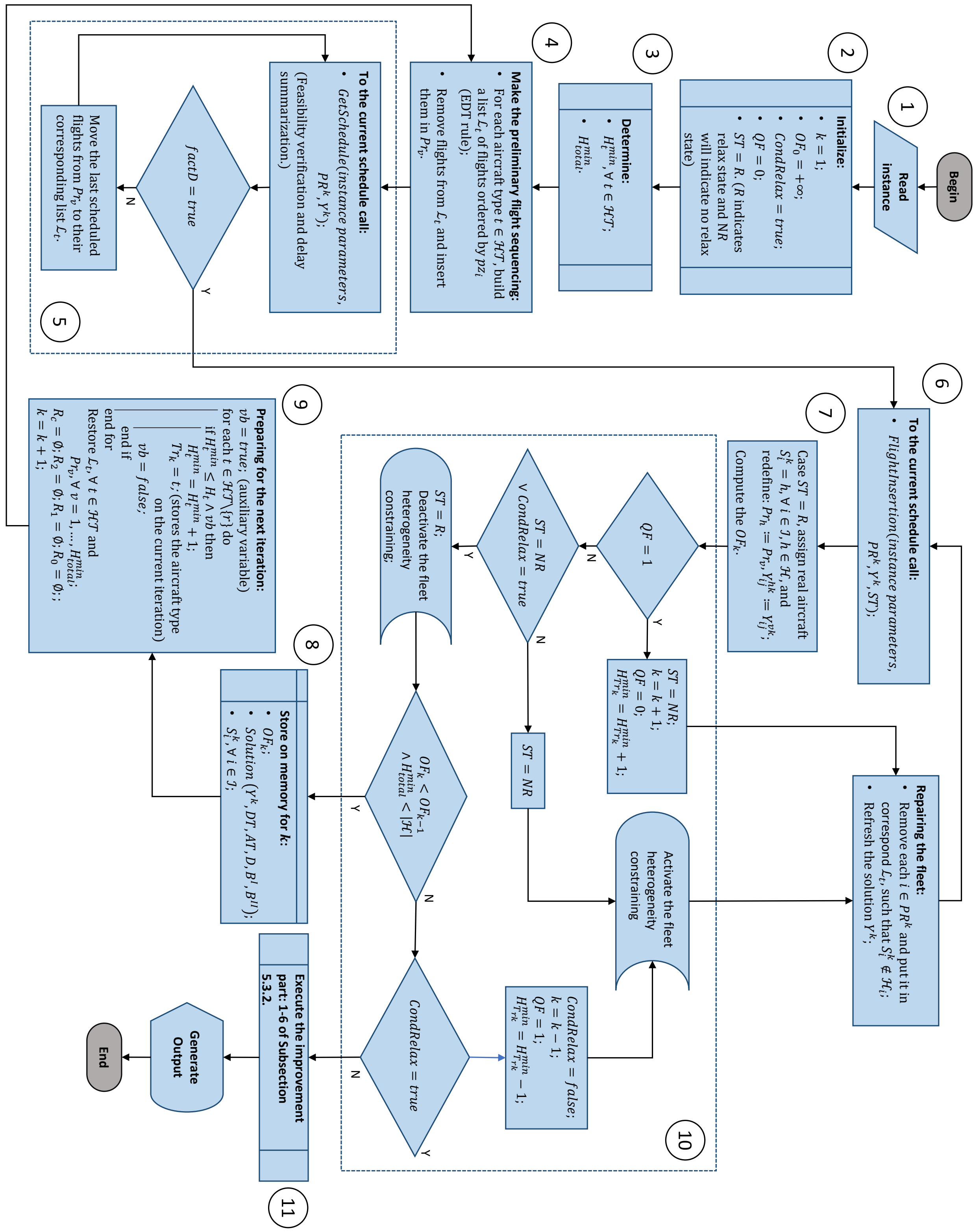
APPENDIX C

Further details on the heuristic approach of Chapter 5

To show in details the overall execution scheme of the proposed heuristic, we present the flowchart in Figure 45. 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 condition 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 part.

Moreover, the six procedures of the improvement part described in Subsection 5.3.2 are detailed in Algorithms 14 to 19. In the following section, we introduce the additional notation used in these algorithms. Let $altIH^k_{i,h}$ be the alternative regarding the insertion of flight i in aircraft h of the iteration k, which was stored in the construction part, as mentioned in Subsection 5.3.1. Consider $\mathcal{HP}$ as the set containing only the aircraft allocated in the current schedule, and W_l as the l-th 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 $Y^{2,5}_{3,7} = 1$ 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 $Y^{2,5}_{3,7} = 0$, this function returns -1. It is important enough to mention that every time a given set Pr_h is modified in the algorithms, set PR^k is updated accordingly.

Figure 45 – Flowchart of the heuristic approach.



Source: Own authorship.

Algorithm 14: *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 15: *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**
      $FV1 \leftarrow OF_k; j \leftarrow findIndex(i, h, \text{'o'}); b \leftarrow findIndex(i, h, \text{'d'});$
      $Y^{h,k}_{j,i} \leftarrow 0; Y^{h,k}_{i,b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i\};$
      switch i **do**
        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**
          $Y^{h,k}_{j,i'} \leftarrow 1; Y^{h,k}_{i',b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i'\};$
          switch i' **do**
            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**
            $FV1 \leftarrow FV2;$
            $i^* \leftarrow i'; b^* \leftarrow (v1 + v2);$
          $Y^{h,k}_{j,i'} \leftarrow 0; Y^{h,k}_{i',b} \leftarrow 0; Pr_h \leftarrow Pr_h \backslash \{i'\};$
          switch i' **do**
            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**
        $Y^{h,k}_{j,i^*} \leftarrow 1; Y^{h,k}_{i^*,b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i^*\};$
        switch i^* **do**
          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
        $Y^{h,k}_{j,i} \leftarrow 1; Y^{h,k}_{i,b} \leftarrow 1; Pr_h \leftarrow Pr_h \cup \{i\};$
        switch i **do**
          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 16: *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 17: *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 18: *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 $i' = 0$ **to** $|\mathcal{I}|+1$, step $+1$ **do**
      for $j' = 0$ **to** $|\mathcal{I}|+1$, 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 $i^* = 0$ **to** $|\mathcal{I}|+1$, step $+1$ **do**
          for $j^* = 0$ **to** $|\mathcal{I}|+1$, 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 19: *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 $\overline{DT}_j - \overline{DT}_i < sb$ **then**
        $\overline{DT}_i \leftarrow r_j + sb$;
      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 $p_j = p_i \wedge \overline{DT}_j - \overline{DT}_i < tu_i$ **then** $factD \leftarrow false$;
        if $\overline{DT}_j - \overline{DT}_i < sb$ **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 $0 < \overline{D}_i \leq d_I^{max} \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $\overline{B}_i^I \leftarrow 1$;
    if $d_{II}^{max} < \overline{D}_i \leq d_{II}^{max} \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c$ **then** $\overline{B}_i^{II} \leftarrow 1$;
    if $(\overline{D}_i > d_{II}^{max} \wedge i \in \mathcal{I}_0 \cup \mathcal{I}_c) \vee \overline{AT}_i > tw^B$ **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;

APPENDIX D

Heuristic results obtained by scenario SA from Chapter 6

As a complement to Subsection 6.4.4, Tables 44-50 summarize the heuristic results for the remaining problem instances of the numerical experiments. These experiments were conducted to study the impact on computational performance and solution quality of heuristic H with regard to the changes in the weights of the objective function (Scenario SA). The analysis of the information of these tables is similar to the analysis presented for Table 24.

Table 44 – Result variations between scenarios SA and Real for instance I23_2.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-0.29	0	0	0	1	1	0	0	-1	0	-124	124	2	0	-1	0
$\text{Test}^1(f_2)$	-0.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^1(f_3)$	-0.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^1(f_4)$	-0.37	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8
$\text{Test}^2(\mathcal{F}_1)$	-0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^2(\mathcal{F}_3)$	-0.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^2(\mathcal{F}_4)$	-0.37	0	0	0	0	0	0	0	0	0	138	-155	0	0	0	-17
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	0.38	0	0	0	0	0	0	0	0	0	0	11	0	0	0	11
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	0.05	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	0.1	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	0.03	0	0	0	0	0	0	0	0	0	152	-144	0	0	0	8
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-0.2	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-0.16	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-0.2	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-0.22	0	0	0	0	2	0	0	-1	0	300	-20	2	0	-1	280

Source: Own authorship.

Table 45 – Result variations between scenarios SA and Real for instance I35_2.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-3.56	0	0	0	2	3	0	0	-3	0	-141	-218	5	0	-3	-359
$\text{Test}^1(f_2)$	-1.18	0	0	0	0	0	0	0	-1	0	-79	-254	0	0	-1	-333
$\text{Test}^1(f_3)$	-1.27	0	0	0	0	0	0	0	0	-1	-79	-287	0	0	-1	-366
$\text{Test}^1(f_4)$	-3.17	0	0	0	0	0	0	0	-1	0	25	155	0	0	-1	180
$\text{Test}^2(\mathcal{F}_1)$	-1.15	0	0	0	0	0	0	0	-1	0	-79	-254	0	0	-1	-333
$\text{Test}^2(\mathcal{F}_3)$	-1.03	0	0	0	0	0	0	0	0	-1	-79	-287	0	0	-1	-366
$\text{Test}^2(\mathcal{F}_4)$	-2.52	0	0	0	0	0	0	0	-1	0	2	-371	0	0	-1	-369
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-1.69	0	0	0	0	0	0	0	-1	0	-79	254	0	0	-1	175
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-2.64	0	0	0	0	0	0	0	-1	0	25	155	0	0	-1	180
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-2.53	0	0	0	0	0	0	0	0	-1	25	-197	0	0	-1	-172
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-2.52	0	0	0	0	0	0	0	0	-1	25	-197	0	0	-1	-172
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-1.86	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-2.46	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-2.31	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-2.23	0	0	0	0	3	0	0	-2	-1	408	27	3	0	-3	435

Source: Own authorship.

Table 46 – Result variations between scenarios SA and Real for instance I48_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-4.42	0	0	1	3	0	0	0	0	-2	0	-205	50	4	0	-2	-155
$\text{Test}^1(f_2)$	-5	0	0	0	0	0	0	0	0	0	1	-259	-230	0	0	1	-489
$\text{Test}^1(f_3)$	-4.73	0	0	0	0	0	0	0	0	1	3	-349	-1101	0	0	4	-1450
$\text{Test}^1(f_4)$	-5.83	0	0	0	0	0	0	0	0	0	1	155	-439	0	0	1	-284
$\text{Test}^2(\mathcal{F}_1)$	-4.82	0	0	0	0	0	0	0	0	0	1	-259	-230	0	0	1	-489
$\text{Test}^2(\mathcal{F}_3)$	-4.46	0	0	0	0	0	0	0	0	1	0	-270	-246	0	0	1	-516
$\text{Test}^2(\mathcal{F}_4)$	-6.16	0	0	0	0	0	0	0	0	0	1	47	-466	0	0	1	-419
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-4.25	0	0	0	0	0	0	0	0	0	1	-337	768	0	0	1	431
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-6.42	0	0	0	0	0	0	0	0	0	1	155	-439	0	0	1	-284
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-6.33	0	0	0	0	0	0	0	0	1	0	174	-503	0	0	1	-329
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-5.78	0	0	0	0	0	0	0	0	1	0	174	-503	0	0	1	-329
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-6.32	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-5.9	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-6.34	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-7.02	0	0	0	0	2	0	0	0	1	-3	740	57	2	0	-2	797

Source: Own authorship.

Table 47 – Result variations between scenarios SA and Real for instance I56_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-23.27	0	0	1	1	2	-1	0	0	-2	1	-127	-335	4	-1	-1	-462
$\text{Test}^1(f_2)$	-23.36	0	0	0	0	1	-1	0	0	-1	1	13	-237	1	-1	0	-224
$\text{Test}^1(f_3)$	-21.92	0	0	0	0	1	-1	0	0	0	2	13	-502	1	-1	2	-489
$\text{Test}^1(f_4)$	-30.66	0	0	0	0	0	0	0	0	-1	1	1,377	-106	0	0	0	1,271
$\text{Test}^2(\mathcal{F}_1)$	-22.55	0	0	0	0	0	0	0	0	-1	1	39	224	0	0	0	263
$\text{Test}^2(\mathcal{F}_3)$	-6.55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^2(\mathcal{F}_4)$	9.9	0	0	0	0	0	0	0	0	-1	1	1,097	-1,040	0	0	0	57
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-3.06	0	0	0	0	0	0	0	0	0	0	-86	296	0	0	0	210
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-30.48	0	0	0	0	0	0	0	0	-1	1	1,377	-106	0	0	0	1,271
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-29.81	0	0	0	0	0	0	0	0	-1	1	1,377	-106	0	0	0	1,271
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-30	0	0	0	0	0	0	0	0	-1	1	1,377	-106	0	0	0	1,271
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-30.23	0	0	0	0	2	-1	0	0	-1	0	1,377	-131	2	-1	-1	1,246
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-30.67	0	0	0	1	1	-1	0	0	-1	0	1,044	44	2	-1	-1	1,088
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-30.13	0	0	0	1	1	-1	0	0	-1	0	1,044	44	2	-1	-1	1,088
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-30.07	0	0	0	1	1	-1	0	0	-1	0	1,044	44	2	-1	-1	1,088

Source: Own authorship.

Table 48 – Result variations between scenarios SA and Real for instance I67_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	55.57	0	0	0	3	0	0	0	0	-2	0	-492	534	3	0	-2	42
$\text{Test}^1(f_2)$	43.64	0	0	0	0	0	0	0	0	0	0	-24	-147	0	0	0	-171
$\text{Test}^1(f_3)$	42.12	0	0	0	0	0	0	0	0	1	0	-14	-354	0	0	1	-368
$\text{Test}^1(f_4)$	0.25	0	0	0	0	0	0	0	0	0	0	1,087	-177	0	0	0	910
$\text{Test}^2(\mathcal{F}_1)$	42.8	0	0	0	0	0	0	0	0	0	0	-24	-147	0	0	0	-171
$\text{Test}^2(\mathcal{F}_3)$	41.94	0	0	0	0	0	0	0	0	0	0	-24	-149	0	0	0	-173
$\text{Test}^2(\mathcal{F}_4)$	40.66	0	0	0	0	0	0	0	0	0	0	539	-757	0	0	0	-218
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	83.04	0	0	0	0	0	0	0	0	0	0	-220	640	0	0	0	420
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	2.98	0	0	0	0	0	0	0	0	0	0	1,087	-177	0	0	0	910
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-1.71	0	0	0	0	0	0	0	0	1	-1	1,105	-216	0	0	0	889
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-1.6	0	0	0	0	0	0	0	0	1	-1	1,105	-216	0	0	0	889
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	3.45	0	0	0	0	2	0	0	0	0	-1	1,092	-201	2	0	-1	891
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	0.79	0	0	0	0	2	0	0	0	0	-1	1,092	-201	2	0	-1	891
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	1.62	0	0	1	2	3	0	0	0	0	-2	748	17	6	0	-2	765
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-5.46	0	0	1	2	3	0	0	0	0	-2	748	17	6	0	-2	765

Source: Own authorship.

Table 49 – Result variations between scenarios SA and Real for instance I71_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-29.31	0	0	0	2	0	0	0	0	0	-1	-44	-152	2	0	-1	-196
$\text{Test}^1(f_2)$	-20.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^1(f_3)$	-22.97	0	0	0	0	0	0	0	0	1	0	2	-447	0	0	1	-445
$\text{Test}^1(f_4)$	-39.34	0	0	0	0	0	0	0	0	0	0	1,062	-245	0	0	0	817
$\text{Test}^2(\mathcal{F}_1)$	-20.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{Test}^2(\mathcal{F}_3)$	-20.57	0	0	0	0	0	0	0	0	1	-1	-12	-14	0	0	0	-26
$\text{Test}^2(\mathcal{F}_4)$	-8.45	0	0	0	0	0	0	0	0	0	0	503	-392	0	0	0	111
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-0.17	0	0	0	0	0	0	0	0	0	0	-237	854	0	0	0	617
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-39.39	0	0	0	0	0	0	0	0	0	0	1,062	-245	0	0	0	817
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-38.16	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-36.41	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-37.58	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-37.51	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-38.02	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-37.14	0	0	0	0	0	0	0	0	1	-1	1,189	-595	0	0	0	594

Source: Own authorship.

Table 50 – Result variations between scenarios SA and Real for instance I82_3.

Test description	ΔTime (sec)	ΔR_M	ΔR_E	ΔR_{D-2}	ΔR_{D-1}	ΔR_{D0}	ΔT_{A1}	ΔT_{A2}	ΔT_{A3}	ΔH_P	ΔH_N	ΔDL_2	ΔDL_1	ΔnR	ΔnT	ΔnH	ΔnDL
$\text{Test}^1(f_1)$	-54.57	0	0	2	6	1	0	0	0	-4	1	-1,265	-390	9	0	-3	-1,655
$\text{Test}^1(f_2)$	133.08	0	0	0	0	0	0	0	0	-1	1	257	303	0	0	0	560
$\text{Test}^1(f_3)$	108.03	0	0	0	0	0	0	0	0	0	0	55	152	0	0	0	207
$\text{Test}^1(f_4)$	-60.83	0	0	0	0	0	0	0	0	-1	1	2,052	-812	0	0	0	1,240
$\text{Test}^2(\mathcal{F}_1)$	127.44	0	0	0	0	0	0	0	0	-1	1	257	303	0	0	0	560
$\text{Test}^2(\mathcal{F}_3)$	97.85	0	0	0	0	0	0	0	0	0	0	55	152	0	0	0	207
$\text{Test}^2(\mathcal{F}_4)$	25.79	0	0	0	0	0	0	0	0	-1	1	1,093	-1,110	0	0	0	-17
$\text{Test}^3_1(\{\text{pre-scheduled}\})$	-2.34	0	0	0	0	0	0	0	0	-1	1	223	285	0	0	0	508
$\text{Test}^3_3(\text{Test}^3_2, \{\text{normal}\})$	-82.89	0	0	0	0	0	0	0	0	-1	1	2,052	-812	0	0	0	1,240
$\text{Test}^3_4(\text{Test}^3_2, \mathcal{F}_3)$	-88.05	0	0	0	0	0	0	0	0	0	0	1,710	-526	0	0	0	1,184
$\text{Test}^3_5(\text{Test}^3_4, \mathcal{F}_2)$	-84.98	0	0	0	0	0	0	0	0	0	0	1,710	-526	0	0	0	1,184
$\text{Test}^3_6(\text{Test}^3_5, \{0\})$	-81.3	0	0	0	0	0	0	0	0	0	0	1,710	-526	0	0	0	1,184
$\text{Test}^3_7(\text{Test}^3_5, \{0,1\})$	-89.79	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474
$\text{Test}^3_8(\text{Test}^3_5, \{0,1,2\})$	-89.4	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474
$\text{Test}^3_9(\text{Test}^3_5, \mathcal{F}_1)$	-87.54	0	0	0	3	1	0	0	0	-1	-1	562	-88	4	0	-2	474

Source: Own authorship.

APPENDIX E

The proposed primal subproblem for Benders decomposition

Briefly, Benders decomposition (BD) (BENDERS, 1962) is a reformulation method dedicated to convex optimization problems, especially, when the constraint matrix is composed of one or more independent blocks arranged in a dual angular structure and linked by coupling variables. Basically, this approach partitions the model to be solved into two simpler formulation types, called *master problem* (generally with integer variables) and *subproblems* (in general, with continuous variables). Iteratively, this algorithm solves the master problem to grant a (primal) solution as values to be fixed in the subproblems, which in turn, provide optimality cuts (through extreme points) or feasibility cuts (by the presence of extreme rays or directions, given by an infeasible condition) to the master problem. Thus, while the subproblems estimate primal bounds for the original problem, the master problem computes the dual bounds. This procedure is repeated until the gap between the dual and primal bounds is closed. Once BD adds new constraints as it progresses towards an optimal solution, the approach is called row generation. In contrast, Danzig-Wolfe decomposition uses column generation.

As already mentioned, to apply BD, we thought about using the discrete-time model from Subsection 6.2.2. The idea was to relax the formulation so that it could provide dual bounds. The relaxation we idealize comes from the inversion of rounding done for the parameter discretizations with a factor of 5 minutes, i.e., what is "floor" becomes "ceil" and vice-versa. The motivation for this is given by the potential that this model has to provide tight bounds (since times are already considered, however, as discrete), having a much stronger linear relaxation than the other proposed models (including those in Chapter 5). As the method described here is exact, the time has to be determined as continuous, thus temporal (scheduling) constraints must be added to the formulation.

Following the traditional concept of variables' separation, the discrete-time model (without crew workday constraints, to make it easier to solve) was left in the master problem and the temporal constraints in a single subproblem. As aircraft routes are interdependent (i.e., the route that one helicopter takes can affect the schedule of another, due to overlapping restrictions about takeoffs at aerodromes and landings at maritime units), it is not possible for us to have a subproblem for each aircraft (commonly done in VRPs) without loss of generality.

Another habit in this decomposition is to use the dual formulation of the subproblem to avoid the hassle of infeasibility. In our case, to make the primal formulation feasible, we simply add a slack variable (for example, out_i) to permit/indicate flight cancellations.

Even though our Benders' approach was able to find optimal solutions in just one iteration for smaller instances, CPLEX (used as the standard solver throughout the dissertation) was not capable of solving the resulting subproblem from larger instances within the time limit of one hour. Because the subproblem has some interesting insights, we thought it relevant to be presented in this dissertation as an appendix. From a solution of the master problem, we get the assignment among flights and aircraft, and define the following notation:

Parameters:

- $\ddot{\mathcal{I}}$: subset of flights included in planning;
- $\ddot{\mathcal{H}}$: subset of helicopters used in planning:
- $\ddot{h}_i$: helicopter assigned to flight i;
- $\ddot{a}_i$: aerodrome assigned to flight i;
- pOF^{MP}: the sum of remaining penalty values, f_2 (local-transfer of flights among different aerodromes) and f_3 (helicopter utilization), obtained in the master problem.

Decision variables:

- out_i: 1, if flight i is not scheduled on the solution; 0, otherwise (except for mandatory flights);
- Z_{ij}: 1, if flight i precedes (also not immediately) flight j from the same aerodrome or when the maritime units of each are equal; 0, otherwise;
- DT_i: exact departure time of flight i;

And to improve the formulation, we propose for each flight $i \in \ddot{\mathcal{I}}$:

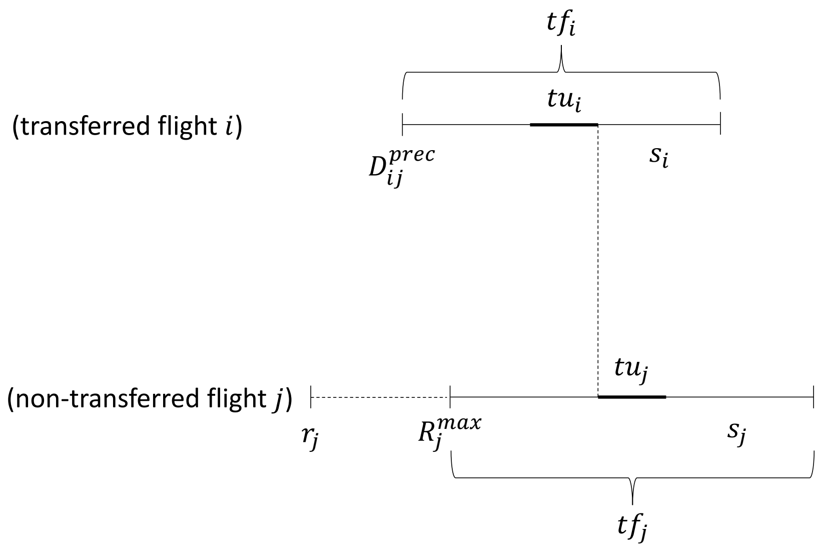
$$r_i^{min} = \max\{r_i, tw^A_{\ddot{a}_i}\};$$

$$R_i^{max} = \begin{cases} \min\{r_i + d^{max}, tw^B_{\ddot{a}_i} - tf_{i,\ddot{h}_i}\}, & n_i = 0,3,4; \\ tw^B_{\ddot{a}_i} - tf_{i,\ddot{h}_i}, & n_i = 1,2. \end{cases}$$

$$D_{ij}^{prec} = R_j^{max} + s_{j,\ddot{h}_j} - tu_i - s_{i,\ddot{h}_i} \mid pr_{ij} = 1 \wedge n_i = 1,2 \wedge n_j = 0,3,4;$$

where parameter r_i^{min} is the earlier start for the take-off of flight i considering the aerodrome opening, R_i^{max} determines the later start of flight i based on the maximum allowed delay (d^{max}), and D_{ij}^{prec} establishes a new maximum departure time by the time window inheritance from a non-transferred flight j to a transferred flight i, when we have precedence priority ($pr_{ij} = 1$). We illustrate how D_{ij}^{prec} is calculated in Figure 46.

Figure 46 – Determination of D_{ij}^{prec}.



Source: Own authorship.

Using the notation defined here, as well as the parameters denoted in Chapter 6, we state the following subproblem:

$$\min \sum_{\substack{i \in \ddot{\mathcal{I}}: \\ n_i = 0,1,2,3}} w_i^1.out_i + \sum_{i \in \ddot{\mathcal{I}}} w_i^4.[DT_i - r_i.(1 - out_i)] + pOF^{MP}; \tag{338}$$

s.t.

$$Z_{ij} + Z_{ji} = 1; \ \forall \ i,j \in \ddot{\mathcal{I}} \mid i > j \wedge (\ddot{a}_i = \ddot{a}_j \vee u_i = u_j); \tag{339}$$

$$DT_i \geq r_i^{min}.(1 - out_i); \ \forall \ i \in \ddot{\mathcal{I}}; \tag{340}$$

$$DT_i \leq R_i^{max}.(1 - out_i); \ \forall \ i \in \ddot{\mathcal{I}}; \tag{341}$$

$$DT_i \leq D_{ij}^{prec}.(1 - out_j) + R_i^{max}.out_j;$$
$$\forall\ i, j \in \ddot{\mathcal{I}} \mid n_i = 1, 2 \wedge n_j = 0, 3, 4 \wedge D_{ij}^{prec} > 0; \quad (342)$$

$$DT_j - DT_i \geq (tf_{i,\ddot{h}_i} + tat).(Z_{ij} - out_i)$$
$$- (wd_{\ddot{h}_j} - tf_{i,\ddot{h}_i}).(Z_{ji} - out_j) - BigM.out_j;$$
$$\forall\ i, j \in \ddot{\mathcal{I}} \mid i \neq j \wedge \ddot{h}_i = \ddot{h}_j$$
$$\wedge \{[r_j^{min} < R_i^{max} + (tf_{i,\ddot{h}_i} + tat)] \vee [R_i^{max} + tf_{i,\ddot{h}_i} - r_j^{min} > wd_{\ddot{h}_j}]\}; \quad (343)$$

$$DT_j - DT_i \geq sb.(Z_{ij} - out_j) - BigM.Z_{ji};$$
$$\forall\ i, j \in \ddot{\mathcal{I}} \mid i \neq j \wedge \ddot{h}_i \neq \ddot{h}_j \wedge \ddot{a}_i = \ddot{a}_j \wedge u_i \neq u_j \wedge (r_j^{min} < R_i^{max} + sb); \quad (344)$$

$$DT_j - DT_i \geq (s_{i,\ddot{h}_i} + tu_i - s_{j,\ddot{h}_j}).(Z_{ij} - out_j) - BigM.Z_{ji};$$
$$\forall\ i, j \in \ddot{\mathcal{I}} \mid i \neq j \wedge u_i = u_j \wedge \ddot{h}_i \neq \ddot{h}_j$$
$$\wedge [r_j^{min} < R_i^{max} + (s_{i,\ddot{h}_i} + tu_i - s_{j,\ddot{h}_j})]; \quad (345)$$

$$Z_{ji} = 0;\ \forall\ i, j \in \ddot{\mathcal{I}} \mid i \neq j$$
$$\wedge \{[n_i, n_j = 0, 3, 4 \wedge (\ddot{a}_i = \ddot{a}_j \vee u_i = u_j) \wedge R_i^{max} < r_j^{min}]$$
$$\vee [(\ddot{a}_i = \ddot{a}_j \vee u_i = u_j) \wedge pr_{ij} = 1]\}; \quad (346)$$

$$out_i = 0;\ \forall\ i \in \hat{\mathcal{I}} \mid n_i = 4; \quad (347)$$

$$DT_j \geq (r_i^{min} + sb).(Z_{ij} - out_j);$$
$$\forall\ i, j \in \ddot{\mathcal{I}} \mid n_i = 0, 3, 4 \wedge n_j = 1, 2 \wedge \ddot{h}_i \neq \ddot{h}_j \wedge \ddot{a}_i = \ddot{a}_j \wedge u_i \neq u_j; \quad (348)$$

$$Z_{ij} \in \{0, 1\};\ \forall\ i, j \in \ddot{\mathcal{I}} \mid i \neq j \wedge (\ddot{a}_i = \ddot{a}_j \vee u_i = u_j); \quad (349)$$

$$out_i \in \{0, 1\};\ \forall\ i \in \ddot{\mathcal{I}}; \quad (350)$$

$$DT_i \geq 0;\ \forall\ i \in \ddot{\mathcal{I}}; \quad (351)$$

The objective function (338) consists of minimizing penalties related to flight cancellation (which makes the problem feasible) and the total flight delay. When $out_i = 1$, w_i^1 is computed to (338) and DT_i becomes zero, according to what is observed in constraint (341).

Constraints (339) comply with the logic of flight precedence. Unlike constraints (300)-(302) of the continuous-time model (Subsection 6.2.1), which contain variables out_i, we kept (339) in equality and put out_i in the temporal constraints (340)-(345). In this way, Z_{ij} also sorts the canceled flights in the first positions, since in this situation, $DT_i = 0$. This makes the linear relaxation stronger compared to constraints (300)-(302).

The family of constraints (340)-(345) are in charge of determining the exact take-off times, building the aircraft schedule. Constraints (340)-(342) define the bounds of DT_i; (343) preserve flight times together with turnaround times (when $Z_{ij} = 1$ and $out_i = 0$) and ensure the minimum crew workday (for $Z_{ji} = 1$ and $out_j = 0$); (344) respect safety briefing times at the aerodromes; and (345) impose the minimum dwell time at maritime units. An improvement we found to reduce the formulation was to restrict the intervals of constraints (343)-(345) to situations where, in the worst case, flight time windows overlap and the crew workload exceeds the capacity. Therefore, these constraints are placed punctually, in really necessary situations.

Variable pre-fixing is carried out by constraints (346) and (347). Constraints (346) cover circumstances in which there is no flight precedence, while (347) oblige mandatory flights to be included in the present planning. Another improvement in the model is provided by the valid inequalities (348), which strengthen the flight time windows.

Finally, (349)-(351) define the domain of our subproblem's decision variables.