Integrated Tail Assignment and Maintenance Task Scheduling: A Decision Support Framework for Airline Schedule Efficiency and Stability

AE5310: Thesis Control and Operations

Luigi Valerio Luciano Pescio

Integrated Tail Assignment and Maintenance Task Scheduling: A Decision Support Framework for Airline Schedule Efficiency and Stability

AE5310: Thesis Control and Operations

by

Luigi Valerio Luciano Pescio

Student number:	4837142	
Project duration:	March 2023 – January 2024	
Thesis committee:	Dr.ir. B.F. Santos	TU Delft, Supervisor
	Dr.ir. A. Bombelli	TU Delft, Chair
	Dr.ir. C. Borst	TU Delft, Examiner
	H. Luijendijk	KLM, Supervisor

Acknowledgements

Over the past year, I had the pleasure of working on my TU Delft Master Thesis in collaboration with KLM Royal Dutch Airlines. I am honored to have had the opportunity to create real value with my work for the industry I love the most.

I would like to start by thanking my Thesis supervisor Dr. Bruno Santos. When we met during the Bachelor thesis project, I could have never imagined the impact you had on my academic success and happiness. Thank you for inspiring me to take this Master's track during the DSE, for helping me get my Master's Internship, and for the support throughout the Master's Thesis. Our discussions guided me, and helped me come up with alternative solutions to my problems; you always allowed me to have high independence in the thinking process and project direction.

Secondly, I thank Hans Luijendijk for allowing me to conduct my research at KLM. This opportunity truly helped me find greater value in what I did, something I would have struggled with otherwise. Thank you for always being very supportive, recognizing my work and effort, celebrating my achievements, and sponsoring me throughout KLM.

I would also like to thank my other KLM colleagues. Starting with Paul van Kessel and Floris Freeman who guided me during my Master Thesis Internship KLM E&M Big Data Team. Thank you for showing me how to structure large computation projects and teaching me about airline maintenance planning. Both have made my Thesis work significantly easier. Moreover, I would like to thank the KLM planners, Niels Steur, Barbara Winterkamp, Yannick Ververda, and Ferry van Leeuwen, who were always happy to show me how they work and available to answer my long questions.

Lastly, I would like to thank my friends and family. Firstly, my girlfriend for being my number one supporter and enduring my long conversations about airline planning. Secondly, my friends with whom I shared a lot of hard work and memories in the past 5 years. Lastly, my parents, to whom I owe a great deal of my success today.

This paper is the conclusion of a big chapter of my life. When I came to Delft more than 5 years ago it was a big transition, a move towards becoming more independent, but I now truly embrace adult life and the responsibilities that come with it. This is scary. However, I am not afraid but excited for what awaits and thankful for the help I have received and will keep receiving from you all.

Luigi Pescio
Delft, January 2024

Contents

List of Figures

List of Tables

List of Abbreviations

AMRP	Aircraft Maintenance Recovery Problem
AOG	Aircraft on Ground
ARP	Aircraft Recovery Problem
CY	Calendar Days
DD	Deferred Defect
DRL	Deep Reinforcement Learning
EO	Engineering Order
EPD	Expected Propagated Delay
FAP	Fleet Assignment Problem
FC	Flight Cycles
FH	Flight Hours
FTE	Full Time Equivalent
LOF	Line of Flight
MEL	Minimum Equipment List
MILP	Mixed Integer Linear Programming
MLSP	Multi-Label Shortest Path
MRI	Maintenance Requirement Item
MRP	Maintenance Recovery Problem
NSRE	Non-Safety Related Equipment
OCC	Operations Control Center
OTP	On-time Performance
PD	Propagated Delay
SDR	Structural Defect Report
TSN	Time-Space Network
WP	Workpackage

Introduction

Airline schedules suffer from disruptions daily. 30.1% of flights arrived more than 15 minutes late and 44.6% of all delay minutes resulted from an aircraft's late arrival in Europe in Q2 of 2023 according to [15]. In 2007 disruptions cost USA airlines $33B [40]. This raises interest in more stable network and maintenance plans for the day of operations to decrease disruptions.

In practice, network and maintenance planning for the day of operations are performed manually and separately, ignoring stability. This inefficiency leads to less stable sub-optimal plans, with increased variability in schedule performance and disruptions on the day of operations. This highlights the value of a decision support framework for integrated tail assignment and maintenance task scheduling to improve schedule efficiency and stability.

Within the literature, several researchers have tackled schedule stability for the day of operations. Predominantly, the focus has been reserved for stable network plans, however, often disregarding maintenance constraints. Most advanced models treat maintenance as a fixed time, fixed interval, non-aircraft-specific activity when creating stable plans. However, maintenance is highly aircraft-specific and hence requires maintenance tasks assignment for detailed modeling. If planned simultaneously, network and maintenance planning can potentially increase overall schedule stability for the day of operations.

This master thesis research presents an innovative formulation for integrated tail assignment and maintenance task scheduling the day before operations. A well-designed framework should provide feasible and real-time decision support, adherent to airline requirements, and have the potential to enhance schedule efficiency and stability. Therefore, the framework's objectives are:

- Improve schedule efficiency.
- Improve schedule stability.

Based on the defined objectives, the research question tackled with this research is formulated as follows:

To what extent can schedule efficiency and stability be improved using a decision support framework for integrated tail assignment and maintenance task scheduling the day before operations?

The thesis is divided into three main parts. Part I consists of the scientific article and is the main part of the report, explaining the framework's reasoning, construction, results, takeaways, and recommendations. Part II presents an extensive literature study, performed at the start of the project to understand the state-of-the-art research in the context of airline operations planning and disruption management, and find a research gap. Lastly, Part III contains additional work that supports the scientific article, such as model verification, model validation, the description of the framework's innovative slot generation process, and a policy sensitivity analysis.

I

Scientific Paper

Integrated Tail Assignment and Maintenance Task Scheduling: A Decision Support Framework for Airline Schedule Efficiency and Stability

Luigi Valerio Luciano Pescio,*

Delft University of Technology, Delft, The Netherlands

Abstract

This study presents a new approach for integrated tail assignment and maintenance task scheduling decision support. Wasted maintenance resources, restricted fleet availability for schedule flexibility, inconsistent planning, and no emphasis on schedule resilience are all inefficiencies that now arise from airlines' separate and manual tail assignment and maintenance task scheduling. The framework addresses these issues, something not yet discussed in the literature. The framework's objective is to deliver feasible plans that increase schedule efficiency (no cancellations, high fleet availability, high fleet health, and optimal use of maintenance resources) and schedule stability (limited number of late arrival disruptions during operations) the day before operation. At its core, a mixed integer linear programming optimization model abides by airline-specific requirements and priorities. The schedule is modeled using an innovative two-space time-space network (TSN), with one space dedicated to maintenance and the other network activities. Stability is introduced with additional ground arcs positioned after flights that, if assigned, allow the creation of flight-specific buffers based on flights' historical arrival delay distributions. Task scheduling is done in registration specific slots, based on aircraft's individual maintenance needs, to improve schedule efficiency. The performance of the framework is tested using a case study provided by a major European single hub-to-spoke airline, with a heterogeneous fleet of over 50 wide-body aircraft. Comparison against the airline's plans over seven days shows that the proposed approach can provide real-time decision assistance and produce more efficient and stable plans. A 17% reduction in maintenance time was achieved, consequently resulting in a 10% increase in fleet availability on the day of operations. This was possible through higher labor and task interval utilization, suggesting that the framework is more efficient at scheduling maintenance tasks. Lastly, the framework's plans were more resilient to arrival delays, reducing the number of disruptions and propagated delay on the day of operations by over 40%.

1 Introduction

Airline schedules suffer from disruptions daily. Poor weather conditions, airport congestion, unavailable staff, and unplanned mechanical failures have led to 30.1% of flights arriving more than 15 minutes late in Europe in Q2 of 2023 according to [EUROCONTROL, 2023]. Moreover, 44.6% of all delay minutes result from an aircraft's late arrival on the previous flight. This emphasizes the need of robust plans that decrease the chance of disruptions and/or facilitate recovery.

Although highly intertwined, tail assignment and maintenance task scheduling, on the day before operations, are in practice performed manually and in separate teams with conflicting interests. This inefficiency leads to sub-optimal plans, with increased variability in schedule performance. Additionally, due to the complexity of the problem, involving a multitude of maintenance tasks, flights, and aircraft, all subject to stringent operational constraints, planners often end up overlooking robustness. This highlights the necessity for a framework that provides robust and integrated tail and maintenance task assignment decision support to airline planners.

To be suitable for implementation, a decision support framework must adhere to numerous requirements set by the airline. Ideally, it should offer solutions within 5 to 10 minutes, allowing planners to make adjustments as new information arrives, such as new maintenance tasks. Planners' priority is to schedule all flights to prevent any cancellations. Thus, the framework should do the same. Furthermore, the framework should create feasible plans ensure aircraft airworthiness and that adhere to the airline's tail restrictions, maintenance schedule, and resource availability.

Despite the growing interest of researchers in robust planning, extensive modeling of simultaneous tail assignment and maintenance task scheduling is lacking in current literature. Predominantly, focus has been reserved for stable and/or flexible network plans, however often disregarding maintenance constraints. Most

*Msc Student, Sustainable Air Transport, Faculty of Aerospace Engineering, Delft University of Technology advanced models, for example, [Z. Liang and Chaovalitwongse, 2015] and [Ahmed et al., 2017], treat maintenance as a fixed time, fixed interval, non-aircraft-specific activity when creating stable plans. While robust maintenance planning models, such as that presented by [Shahmoradi-Moghadam et al., 2021], disregard network operations altogether. In reality, network and maintenance are highly intertwined activities that if integrated can potentially increase schedule stability and efficiency. For instance, more efficient task planning may decrease maintenance time, thereby increasing useful buffer time, helpful for preventing disruptions from flight late arrivals.

This paper addresses these aspects. The innovative framework presented assigns tails and schedules maintenance tasks simultaneously. It incorporates stability using historical flights' arrival delay data. At the core of the framework is an efficient mixed integer linear programming (MILP) optimization model that follows a hierarchical structure set by airline planners. It ensures continuous aircraft airworthiness and respects airline-specific rules and schedules. The task assignment is constrained by the availability of material, machinery, method, and manpower (4M). Unlike literature and airline practices, maintenance slots are modeled as tail-specific activities based on individual aircraft needs, with the intention of improving schedule efficiency. Thereby, this study provides new insights into integrated tail assignment and maintenance task scheduling, aiming to create more efficient and stable plans in the process.

This paper follows the subsequent structure: first, a literature overview provides insights into network planning, maintenance planning, and robust planning research in section 2. Afterward, in section 3, the problem is defined. This is followed by the framework's methodology, which includes the time-space network (TSN), slot generation process, and MILP model, in section 4. Subsequently, section 5 presents results obtained by implementing the framework to a real airline case study, critically demonstrating its validity and benefits in practice. Lastly, conclusions and recommendations for future research end the paper in section 6.

2 Literature overview

Airline network and maintenance planning are complex and multi-stage processes. They both begin years before the actual day of operation. Both are highly intertwined from the start but are often treated separately in the literature and by airlines. Airlines may even outsource maintenance planning to external Maintenance Repair and Overhaul (MRO). This leads to conflicting interests between the network and maintenance planning departments.

Robust planning begins from the inception of network and maintenance planning up until the day before operations. For the most part, it has two objectives: to increase schedule stability, by decreasing disruptions on the day of operations; and/or increase schedule flexibility, by making disruptions easier to recover. The remaining part of this section will dive deeper into network, maintenance, and robust planning. Focusing on how network and maintenance planning are approached by airlines and addressed in the literature, and how robustness is integrated into the planning.

2.1 Network planning

Due to the complexity and size of airline network planning, network schedules are created in sequential steps, both in practice and in the literature. Flight schedule design begins many months before the day of operation and is the process in which airline marketing and network divisions use demand forecasts to decide which O-D pairs to fly ([Cadarso and Marín, 2011]). Subsequently, aircraft fleets are assigned to individual flights according to passenger demands, operating costs, etcetera, such that total profit is maximized ([Abara, 1989, Hane et al., 1995]). A few days before operations, flights are assigned to individual registrations, ensuring that maintenance requirements are respected, with often the objective of minimizing operating costs ([Barnhart et al., 1998, Sriram and Haghani, 2003]). Lastly, crew scheduling is carried out to find the best crew pairings.

Maintenance has in large part been disregarded from network planning literature. Maintenance is only considered at a strategic level in the tail assignment problem to ensure feasible plans. Often it is treated as a fixed interval, fixed duration activity, for example by assigning Lines of Flights (LOFs) that start and end at maintenance stations ([Barnhart et al., 1998]) or by assuming overnight maintenance when aircraft visit maintenance stations ([Sriram and Haghani, 2003])). However, maintenance is aircraft-specific, as each aircraft has different maintenance requirements and tasks. Attempts at modeling aircraft-specific maintenance in the tail assignment are few, such as [Sarac et al., 2003], but are too abstract to consider maintenance tasks assignment.

2.2 Maintenance planning

Like network planning, maintenance planning is a multi-stage problem. The process commences with check scheduling, or letter check scheduling, in which long-term maintenance activities are assigned to tails, often for multiple years ([Deng et al., 2020, Deng and Santos, 2022]). Next, during the flight schedule design phase, time blocks, called maintenance slots, are reserved for maintenance. A few days before the day of operations, alongside tail assignment, maintenance tasks are assigned to maintenance slots, while ensuring the 4M constraints ([Marseguerra and Zio, 2000, Witteman et al., 2021, Papakostas et al., 2010]). Lastly, maintenance resources (mechanics and equipment) scheduling is carried out.

In search of a higher level of detail, researchers have focused on the task assignment problem as a stand-alone, leaving the tail assignment out of scope. For example, [Shaukat et al., 2020, Witteman et al., 2021, Papakostas et al., 2010] solve the task assignment problem at the strategic or operational phase, with similar objectives (minimization of costs and task interval waste), and have some sort of uncertainty given by the arrival of tasks. However, they all disregard network considerations. [Lagos et al., 2020] managed to integrate the tail assignment and maintenance task scheduling. They created an algorithm that uses Integer Programming, in combination with a Markov decision process to schedule LOFs and maintenance tasks to nighttime slots. They prevent tasks from going due and differentiate between critical and non-critical tasks. However, compared to other task assignment literature, they disregard key aspects, such as the availability of resources.

2.3 Robust planning

A collect of airline operation planning research has been looking into robust planning with the objective of facilitating and alleviating pressure on schedule recovery on the day of operation. This relatively new concept in airline operations planning research has, for the most part, been divided into proactive absorption and proactive recovery [Clausen et al., 2010].

Proactive absorption

The goal of proactive absorption is to design plans that are resistant to disruptions, and thus ensure schedule stability. In the past, researchers have achieved schedule stability in multiple ways, the most common being: time buffers, flight retiming, and delay robust routing.

[AhmadBeygi et al., 2010] proposed a model that assigns slack with the objective of minimizing flight delay during the schedule design phase. Later, a mathematical model for integrated fleet assignment and flight retiming, which determines flight departure times by maximizing a surrogate robustness measure, was developed ([Aloulou et al., 2010]). Most recently, [Ahmed et al., 2017] built a two-stage model for robust multi-fleet aircraft routing. It first makes use of a MILP to find optimal aircraft routes, and then through a heuristic approach based on Monte Carlo simulation lights are re-timed with the aim of maximizing on-time performance.

Delay robust routing involves using historical delay data to create more stable aircraft routes. [Lan et al., 2006] were among the first to study this concept. Using a column generation algorithm, they assigned LOFs to aircraft intending to minimize the expected propagated delay, determined from historical airline data. Aircraft routing was then followed by a MILP retiming model that minimizes passengers' misconnections. A similar model was developed by [Z. Liang and Chaovalitwongse, 2015], who also considered maintenance, planned overnight every three days. Researchers have created stability in maintenance plans in comparable ways. [Weide et al., 2022] developed a robust check-scheduling model, running ten scenarios with check duration uncertainty, while [Shahmoradi-Moghadam et al., 2021] used Monte Carlo simulations to take uncertainty samples from task duration sets in their robust maintenance task scheduling model. Nonetheless, detailed maintenance and network planning are solved separately.

Proactive recovery

The aim of proactive recovery is to design plans that are easily adjusted when disruptions occur, and thus ensure schedule flexibility. The most common method to promote schedule flexibility has been increasing aircraft swapping opportunities. However, overall, methods are few and not widely researched.

[Ageeva, Y. V., 2000] pioneered the concept of using swapping opportunities to increase schedule flexibility. They created a LOF-based aircraft routing model with the objective of maximizing swapping opportunities. The method was re-proposed by [Burke et al., 2010], who solved the tail assignment problem with a multi-meme memetic algorithm by integrating multiple robustness strategies, for both stability and flexibility. They re-timed flights to optimize stability, given by the probability that each flight departs on time, and flexibility, given by the probability that each flight involved in a swap departs on time. Lastly, [Lapp and Cohn, 2012] presented an optimization algorithm that suggests the best swapping opportunities between LOFs of aircraft. Their objective was to minimize aircraft requiring maintenance but assigned to a LOF with no maintenance opportunities. Nonetheless, they treated maintenance as a fixed interval activity.

Few other methods for proactive recovery have been researched. One is partitioning the schedule into sub-networks to safeguard high-revenue itineraries ([Kang, L. S., 2004]). Another is promoting station purity (i.e. reducing the number of aircraft types visiting an airport) to facilitate aircraft swapping ([Smith and Johnson, 2006]). Lastly, the idea behind short cycle scheduling together with hub-isolation (i.e. LOFs with few flights which include only one hub airport) is to prevent disruptions from spreading across hubs and affecting many later flights ([Rosenberger et al., 2004]). Overall, most of these methods are not applicable to all types of airline networks, are less effective at creating flexibility, and still disregard maintenance.

Paper	**SD**	**FA**	**TA**	**PA**	**PR**	**Maintenance planning**
[Ageeva, Y. V., 2000]			✓		SO	
[Kang, L. S., 2004]		✓	✓		SN	
[Rosenberger et al., 2004]		✓			SC	
[Lan et al., 2006]			✓	FR, DRR		
[Smith and Johnson, 2006]		✓			SP	
[AhmadBeygi et al., 2010]	✓			FR		
[Aloulou et al., 2010]		✓		FR		
[Burke et al., 2010]			✓	FR, DRR	SO	Fixed interval activity
[Lapp and Cohn, 2012]			✓		SO	Fixed interval activity
[Dunbar et al., 2014]			✓	FR, DRR		
[Z. Liang and Chaovalitwongse, 2015]			✓	FR, DRR		Fixed interval activity
[Ahmed et al., 2017]			✓	FR		Fixed interval activity
[Shahmoradi-Moghadam et al., 2021]				MSU		Task scheduling
[Weide et al., 2022]				MSU		Check scheduling
This paper			✓	DRR		Task scheduling

Table 1: Overview of airline robust planning literature. Abbreviations in the table: SD: Schedule design; FA: Fleet assignment; TA: Tail assignment; PA: Proactive absorption; PR: Proactive recovery; FR: Buffer and/or flight retiming; DRR: Delay robust routing; MSU: Maintenance scheduling with uncertainty; SO: Swapping opportunities; SN: Sub-networks; SP: Station purity; SC: Short cycles.

Based on this literature overview, it is concluded that robust planning that integrates tail assignment and maintenance task scheduling is currently not addressed in the literature. Building stable plans in the tail assignment phase, often by re-timing flights and/or using historical delay data, has been the focus. For the most part maintenance in the tail assignment problem has either been excluded or treated as a fixed time, fixed interval, non-aircraft specific activity. However, maintenance is aircraft-specific. Simultaneous tail assignment and task scheduling can potentially decrease airline operating costs (schedule efficiency) and recovery costs (schedule stability). This research, therefore, aims to solve the tail and task assignment simultaneously.

3 Problem definition

New arriving information requires adjustment to tail and maintenance task assignments. Commonly, airline planners do this manually, which is complex and time-consuming. Moreover, the tail and task assignment are often the responsibilities of two different teams with conflicting interests. Therefore, the airline's habitual planning approach can result in wasted resources, inconsistent planning, and no emphasis on schedule resilience. Implementing a framework for simultaneous tail assignment and maintenance task scheduling can support airline planners in the decision-making process. Such a framework can reduce compromises airline planners face, such as the trade-off between useful buffer time and maintenance time. A well-designed framework has the potential to create more efficient, stable plans that reduce operating costs and disruptions on the day of operations.

Therefore, to assist airline planners, the framework presented in this paper has the following objectives:

1. Improve schedule efficiency.

2. Improve schedule stability.

The models' objectives are defined by sub-objectives, as shown in Figure 1. Schedule efficiency induces lower operating costs for an airline. It is characterized by no cancellations, high fleet availability, high fleet health, and optimal use of maintenance resources. On the other hand, improved schedule stability lowers disruptions recovery costs, reduces workload on the day of operations, and leads to indirect revenue gains from more satisfied passengers. In this paper, improved stability is defined by a reduction in aircraft's subsequent jobs starting late due to the late arrival of their previous flight. In the framework, six unique decision levers enable the achievement of the sub-objectives. These levers will compose the framework's objective function.

Some of the framework's sub-objectives need to be defined. Ground time, apart from time on the ground for maintenance, is divided into fleet availability and ground time waste. Fleet availability provides flexibility to the airline in the form of potential aircraft-swapping opportunities. In contrast, ground time waste, which is ground time before maintenance slots, offers no flexibility. This is because aircraft often cannot be swapped before planned maintenance. On a given day, fleet health equals the number of days before the first maintenance task of an aircraft goes due, averaged over the entire fleet. When fleet health is high, most of the fleet does not require maintenance in the next days, thereby offering greater flexibility to network. Lastly, an optimal use of resources is characterized by improved task interval utilization, reduced maintenance time, and higher labor utilization. This ought to reduce airline maintenance costs.

The framework intends to provide tail assignment and maintenance task scheduling support on the day before operations. As such it can assign and cancel flights, but cannot retime them due to the vicinity of the departure date. However, for a fixed number of flights per day it can speed-up the turn around process. The framework plans maintenance slots, which are time slots reserved for maintenance activity, to registrations. Fleet health and maintenance resource utilization are optimized by scheduling maintenance tasks in maintenance slots at the preferred time. Alternatively, the framework can also defer maintenance tasks. A days-clean target, i.e. the target aircraft health after a maintenance slot, is used to decide which maintenance tasks to schedule and which to defer. Lastly, it selects when and on which tails to plan ground time to increase fleet availability and schedule stability. These are the decision levers that enable the framework to achieve its objectives, as shown in Figure 1.

The tail assignment is constrained by airline-specific rules, aircraft airworthiness constraints, and aircraft balance. Each airline has rules restricting certain registrations from flying to certain destinations, or limiting the number of daily quick turnarounds. Airworthiness constraints connect the tail assignment and task scheduling problems. These ensure that an aircraft is assigned to a flight only if has no outstanding maintenance task going due during the flight. Flow balance must be guaranteed as tails cannot be assigned to two overlapping flights. Lastly, it is assumed that the framework's tail assignment is not restricted by decisions taken the previous day. Every day it can create a tail assignment from scratch.

The process of scheduling maintenance tasks is constrained by the 4M requirements. These include scheduling a task before its due date, only when material becomes available, and in a sufficiently long maintenance slot at the right location (hangar or platform). The last 4M requirement, which regards assigning the correct mechanics skills required to perform a task, is not modeled in the framework. Hence it is assumed that the right skills to perform tasks are always available. As only relatively simple maintenance tasks, which require skills possessed by most mechanics, are scheduled by the framework, this assumption is non-limiting.

Besides the 4M requirements, task scheduling is constrained by rules that limit the assignment of aircraft to maintenance slots. As airline mechanics are often certified for one aircraft type, maintenance schedules are created per aircraft type (i.e. Boeing 787). Thus it is assumed that maintenance slots are not interchangeable across aircraft types. Moreover, like the tail assignment, the framework does not have to respect the task assignment planned the previous day. Hence, it can create a new assignment each day.

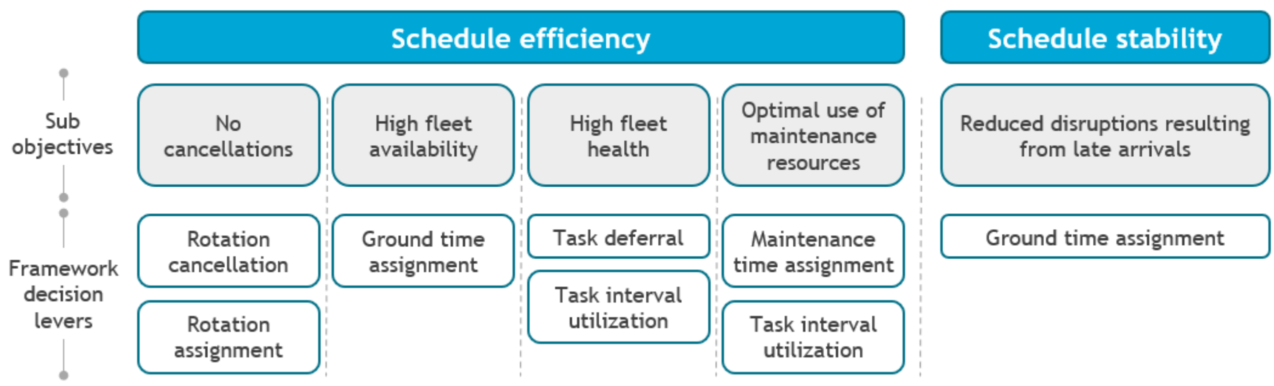

Figure 1: Breakdown of the framework's objectives into sub-objectives and decision levers.

4 Modelling framework

Based on the problem definition, the framework is structures as shown in Figure 2. On the day before operations, with inputs shown on the left, airline planners assign tails and schedule maintenance tasks. They do so often for the upcoming two to three days. At the end of the day, planners deliver their final plan to the operations control team. They will manage disruptions and revise the tail and task assignments on the day of operations.

The developed framework uses the same inputs to create the final plan on the day before operations. Tail assignment and maintenance task scheduling are integrated using innovative techniques. The remaining parts of this section will cover the framework's formulation in detail.

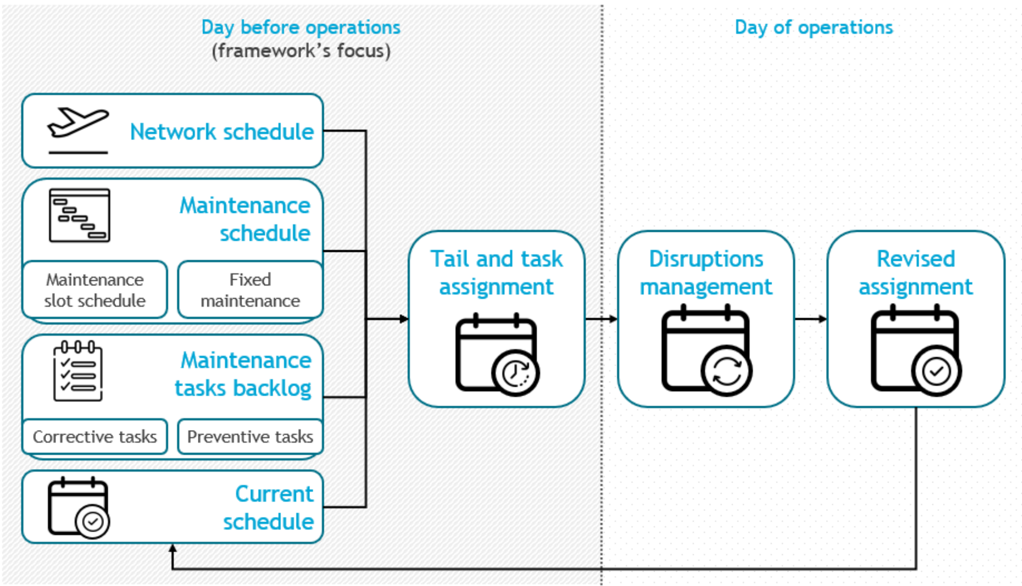

Figure 2: Overview of the tail assignment and maintenance task scheduling framework (all icons from Flaticon).

4.1 Model inputs

Airline planners assign tails and schedule maintenance tasks using generally three inputs: a network schedule, a maintenance schedule, and a maintenance tasks backlog. These same inputs are used by the framework, and are defined as follows:

- *Network schedule*: The network schedule is composed of flights. In the framework, flights have been combined in short sequences of flights (often just two) starting and ending at the same airport called rotations. This suits the network of hub-and-spoke carriers, with mostly only flights from and to their hubs.
- *Maintenance schedule*: The maintenance schedule provides insight into the maintenance slot schedule and fixed maintenance activities. Each is explained in further detail below:
  - *Maintenance slot schedule*: The maintenance slot schedule is a plan, often one week long, outlining maintenance scheduling opportunities. From it, one can obtain maintenance slots start and end times, work location (hangar or platform), assigned full time equivalents (FTEs) per hour, duration, and suitable aircraft type.
  - *Fixed maintenance*: Fixed maintenance activities include long-term letter checks, modifications, and other maintenance activities scheduled several weeks in advance. In the framework, all fixed maintenance activity must be scheduled, and tails cannot be assigned to overlapping jobs during these checks.
- *Maintenance tasks backlog*: This is a backlog of aircraft-specific maintenance tasks ready for scheduling. Following common airline practice, maintenance tasks are subdivided into two categories: preventive and corrective tasks. Differences are as follows:
  - *Preventive tasks*: Recurring tasks executed following interval requirements imposed by regulatory agencies, like the European Aviation Safety Agency. Airlines also have the opportunity to add additional preventive tasks. Once a task is executed the due date is reset. An example is the visual inspection of aircraft brake wear, which repeats at a fixed number of take-off and landing cycles.
  - *Corrective tasks*: One-off tasks executed only once before their due date. Corrective tasks also derive from either regulatory agencies (e.g., Minimum Equipment List (MEL)) or the airline (e.g., Non-Safety Related Equipment (NSRE)). The replacement of brake pads after a preventive check is an example of a MEL, while a broken tray table is an example of an NSRE.

  These tasks are further divided into mandatory and deferrable tasks. This division is based on the task type and the harshness of its due date. Table 2 provides an overview of the task division.

		Preventive	Corrective
Mandatory	Tasks with fixed due dates	Requirements: Minimum Required Item (MRI), Engineering Order (EO), and Structural Defect Report (SDR)	Ad-hoc, Minimum Equipment List (MEL)
Deferrable	Tasks with soft due dates		Non Safety Related Equipment (NSRE)

Table 2: Common subdivision of maintenance tasks used in practice and in the modeling framework.

4.2 Maintenance slots

The framework creates its maintenance slots based on aircraft-specific needs. This has advantages over current state-of-the-art approaches which make use of generic fixed times, fixed duration maintenance slots from the airline's maintenance schedule, often suitable for the entire type fleet. For instance, the framework creates maintenance slots specific to one registration, whose duration is based on the aircraft's individual maintenance needs. This should result in higher labor utilization, less maintenance time, greater fleet availability, and hence improved schedule efficiency. Nonetheless, this methodology increases complexity in the model as it introduces additional decisions. Whereas in practice it is based on the assumption that towing services are available at any time and it requires a revolution in current maintenance slot planning approach.

Maintenance slots in the framework are created using a two-step process. First, on a given day the model determines aircraft-specific required maintenance time based on their task backlogs and a days-clean target. Subsequently, using the calculated maintenance time duration, the framework creates maintenance slots that can only be assigned to the corresponding registration. It ensures that slots are created within the confines of the existing maintenance slot schedule to respect the airline's resource limitations.

On rare occasions, to prevent infeasibility, off-hour maintenance slots are created. These do not abide by the airline's maintenance slot schedule, and thus in practice would require quick alterations to mechanics' work shifts. These slots are created when the aircraft has tasks going due the following day but there are no suitable maintenance opportunities in the schedule. Otherwise, also when the aircraft's required maintenance time is longer than the longest slot in the schedule. Maintenance planners also create off-hour slots when similar issues arise.

4.3 Time-space network

To assign tails and schedule maintenance tasks simultaneously, the framework uses parallel time-space networks (TSN). The TSN construction enables the optimization of schedule efficiency and stability. Advantages of parallel TSNs include individual aircraft routing and ground time assignment. Parallel TSNs have been commonly used in literature for tail assignment, such as demonstrated by [Vink et al., 2020]. However, the novelty stands in solving the maintenance task assignment within the TSN.

Figure 3 depicts a simplified example of an aircraft's parallel TSN employed by the framework, whereby each aircraft has a separate network with independent ground arcs and maintenance slots. Coverage constraints for rotations are used to connect all networks. To decrease the network's size, time is discretized in non-homogeneous time steps. These steps depend on the start and end times of the aircraft's potential jobs (rotations, maintenance slots, and fixed maintenance slots). A Job's duration is rounded to the next nearest minimum time step defined by the airline's network and maintenance schedules. Moreover, rotation arcs are extended to accommodate for a fixed minimum turnaround time (TAT) before departure and after arrival. Maintenance slot arcs are also elongated with the location-dependent fixed tow time before the start and after the end of the slot.

Schedule efficiency

Part of the TSN's novelty lies in its utilization of two distinct spaces: the air space and the maintenance space. The assignment of rotations takes place at the air space, while scheduling maintenance tasks and maintenance slots is confined to the maintenance space. This dual-space approach serves the purpose of distinguishing between fleet availability and ground time waste. All air space ground arcs account for fleet availability, whereas all those of the maintenance space contribute to ground time waste.

Using connection arcs, aircraft can travel between the two spaces. Aircraft can travel to the maintenance space at their source node or only directly after the rotation they just flew. While aircraft can connect back to the air space at the end of all maintenance slots, including fixed maintenance slots. A side effect of the TSN formulation is that it results in a trade-off between schedule efficiency and schedule stability. Reducing ground time waste before a maintenance slot increases schedule efficiency. However, it also deteriorates schedule stability as the chances of the previous rotation's arrival delay propagating into the slot increase. Next, it is explained how schedule stability is optimized in the TSN.

Schedule stability

Stability is incorporated by using additional, non-mandatory ground arcs in the TSN, referred to as robust ground arcs. These are based only on rotations' historical arrival delays, excluding other sources of disruptions. The objective of robust ground arcs is to reduce the chances of rotations' arrival delay from propagating into the aircraft's next jobs. However, delay mitigation should not compromise the framework's other objectives. This approach enables to simultaneously optimize schedule stability and efficiency, thereby enabling a trade-off between the two.

Every rotation has an air space and a maintenance space robust ground arc. Each starts at the node corresponding to the rotation's end time at the respective space. The duration of both arcs is set using a delay mitigation parameter that represents the rotation's Nth percentile arrival delay determined from historical airline delay data. For instance, a delay mitigation parameter of 90% implies that 90% of the rotation's historical delays are mitigated if the robust ground arc were to be used. A trade-off analysis to determine the delay mitigation parameter will follow in section 5. Lastly, a rotation's robust ground arc can only be assigned to the same aircraft assigned to fly the rotation. This is enforced by constraints in the MILP model.

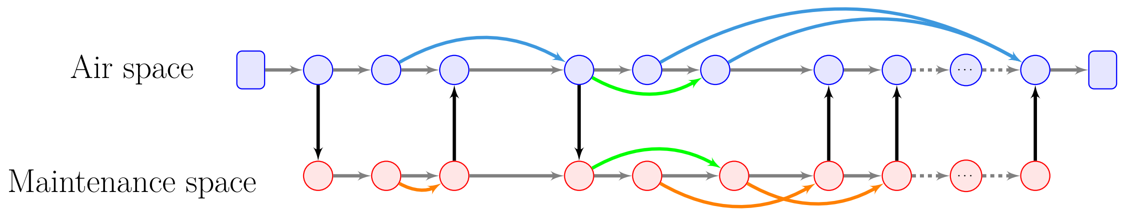

Figure 3: Parallel time-space network used in the framework. Rectangular nodes represent the aircraft's source and sink nodes. In light blue: rotations; in orange: maintenance space jobs (maintenance slots and fixed maintenance activities); in grey: ground arcs; in black: connection arcs; and in green: robust ground arcs.

4.4 MILP optimization model formulation

Core to the modeling framework is the MILP optimization model that simultaneously assigns tails and schedules maintenance tasks to optimize schedule efficiency and stability. The model's sets, parameters, decision variables, objective function, and constraints are discussed in this section.

4.4.1 Notations

The model's notation is introduced first. First, definitions for all the model's indices and sets can be found in Table 3. Then, the model's parameters are presented in Table 4, and can have either values that are fixed or dependent on the properties of model sets. Lastly, the model's decision decision variables are found in Table 5. Tail assignment is characterized by two decision variable types, one to assign tails to arcs and the other to cancel rotations. Similarly, maintenance task scheduling is possible through two variables, one to assign tasks to maintenance slots and the other to defer tasks.

4.4.2 Objective function

The objective function is formulated as a cost-minimization weighted sum objective. It takes inspiration from common airline objectives preferences. An explanation of the formulation of weight functions will follow in section 5. The MILP mathematical formulation of the objective function is given by Equation 1. It is composed of six terms, one for each decision lever presented in section 3.

$$\begin{aligned} Min : \sum_{f\in F} \delta_{Canx,f} W_{Canx,f} + \sum_{f\in F}\sum_{a\in A_f} \delta_{a,f} W_{a,f} + \sum_{t\in T\neg T_{Due}} \delta_{Defer,t} W_{Defer,t} C_{Type,t} + \\ \sum_{a\in A}\sum_{m\in M_a} \delta_{a,m} W_{a,m} + \sum_{m\in M}\sum_{t\in T_m} \delta_{t,m} W_{t,m} C_{Type,t} + \sum_{a\in A}\sum_{g\in G_a} \delta_{a,g} W_{a,g} \end{aligned} \quad (1)$$

In more detail, the objective function terms are:

- *Rotation cancellation*: First, cancellations are minimized. Network planners can decide to cancel a rotation if it cannot be operated or to safeguard the future schedule. However, this induces high costs to the airline and is often a remedy of last resort. This severity is represented in the model with the highest weight, $W_{Canx,f}$.

Set	Definition
$s \in S$	Set of all TSN arcs s (rotations, maintenance slots, ground arcs, etcetera)
$f \in F \subset S$	Set of rotations f
$m \in M \subset S$	Set of maintenance slots m
$u \in U \subset S$	Set of fixed maintenance slots u
$g \in G \subset S$	Set of ground arcs g
$b \in B \subset G$	Set of air space ground arcs (fleet availability) b
$w \in W \subset G$	Set of maintenance space ground arcs (ground time waste) w
$c \in C \subset S$	Set of connection arcs c
$d \in D$	Set of days d in the planning window
$pd \in PD_d \subset D$	Set of days preceding day d including day d
$s \in S_d \subset S$	Set of arcs s on day d
$k \in K$	Set of aircraft types k (e.g. Boeing 777)
$a \in A$	Set of aircraft a
$a \in A_s \subset A$	Set of aircraft a that can be assigned to arc s
$a \in A_k \subset A$	Set of aircraft a of type k
$s \in S_a \subset S$	Set of arcs s that can be assigned to aircraft a
$t \in T$	Set of tasks t open at the start of the planning time
$t \in T_{Due} \subset T$	Set of open mandatory tasks t due on the first day of the plan (i.e. day of operation)
$t \in T_a \subset T$	Set of open tasks t of aircraft a
$t \in T_m \subset T$	Set of open tasks t that can be scheduled in maintenance slot m
$t \in T_d \subset T$	Set of open mandatory tasks t going due on or after day d
$n \in N$	Set of nodes n
$n \in MN \subset N$	Set of maintenance space nodes n
$n \in AN_a \subset N$	Set of air space nodes n with a terminating air space ground arc g and an originating connection arc c for aircraft a
$l \in L$	Set of maintenance locations l (hangar and platform)
$m \in M_t \subset M$	Set of maintenance slots m in which task t can be scheduled
$m \in M_{d,l} \subset M$	Set of non-off-hours maintenance slots m at maintenance location l on day d
$m \in M_{n,l,k} \subset M$	Set of non-off-hours maintenance slots m suitable for aircraft subtype k at maintenance location l passing through maintenance node n
$m \in M_a \subset M$	Set of maintenance slots m that can be assigned to aircraft a
$m \in G_a \subset G$	Set of ground arcs g that can be assigned to aircraft a
$g \in RG_{a,r} \subset G$	Set of robust ground arcs g of rotation f that can be assigned to aircraft a
$s \in O_{n,a} \subset S$	Set of arcs s originating from node n that can be assigned to aircraft a
$s \in T_{n,a} \subset S$	Set of arcs s terminating at node n that can be assigned to aircraft a
$f \in QTA_d$	Set of quick turnaround rotations on day d

Table 3: Definitions of model's sets and indices

Parameter	Unit	Definition
f_{QTA}	[-]	Quick turn around copy of rotation f
$FB_{n,a}$	[-]	Flow balance at node n for aircraft a: 1 at source nodes, -1 at sink nodes, and 0 at intermediate nodes
FTE_m	[/h]	FTE required for maintenance slot m
$FTE_{n,l,k}$	[/h]	FTE available at node n at location l for aircraft type k
$MaxM_{d,l,k}$	[-]	Maximum number of slots at location l and suitable for aircraft type k that can be assigned on day d
$MaxQTA$	[-]	Maximum number of daily quick turnaround rotations
L_t	[hr]	Labor hours required to complete task t
$L_{max,m}$	[hr]	Maximum labor hours that can be assigned in maintenance slot m
$BigM$	[-]	A relatively large number
$c_{a,n}$	[-]	Connection arc c originating from air space node n for aircraft a
$b_{a,n}$	[-]	Air space ground arc b terminating at air space node n for aircraft a
$DL_{t,d}$	[days]	Number of days remaining from day d until task t is due
H_f	[days]	Minimum required aircraft health to operate rotation f (e.g., for a rotation leaving today and returning tomorrow $H_f = 1$)
$C_{Type,t}$	[-]	Task type criticality coefficient of task t
$W_{a,f}$	[-]	Cost of assigning aircraft a to rotation f
$W_{a,m}$	[-]	Cost of assigning aircraft a to maintenance slot m
$W_{a,g}$	[-]	Cost of assigning aircraft a to ground arc g
$W_{Canx,f}$	[-]	Cost of cancelling rotation f
$W_{t,m}$	[-]	Cost of assigning task t to maintenance slot m
$W_{Defer,t}$	[-]	Cost of deferring task t

Table 4: Definitions of model's parameters

Decision variable	Definition
$\delta_{a,s}$	Binary variable equal to 1 if aircraft a is assigned to arc s, 0 otherwise
$\delta_{Canx,f}$	Binary variable equal to 1 if rotation f is cancelled, 0 otherwise
$\delta_{t,m}$	Binary variable equal to 1 if task t is assigned to maintenance slot m, 0 otherwise
$\delta_{Defer,t}$	Binary variable equal to 1 if task t is deferred, 0 otherwise

Table 5: Definitions of model's decision variables

- *Rotation assignment*: Secondly, flight operating costs are minimized ($W_{a,f}$). In practice and in the model this decision depends on the aircraft's fuel efficiency and the rotation's block time.

- *Task deferral*: The next objective is to minimize the cost of deferring maintenance tasks ($W_{Defer,t}$) to optimize fleet health. Maintenance resources are limited and/or performing maintenance may mean canceling a flight. Therefore sometimes the best choice is to defer a task. This decision depends on the task hierarchy, implemented with $C_{Type,t}$ as done by [Van Kessel et al., 2023], and on the interval utilization (i.e., the moment in time a task is executed relative to its due date). Regarding the latter, the following two distinctions are made:

  - For mandatory tasks, the deferring cost increases close to the due date, as diminishing scheduling opportunities lead to a higher risk of grounding the aircraft.
  - For deferrable tasks, the deferring cost is constant, as there is no risk of grounding the aircraft as the task can be postponed.

- *Maintenance time assignment*: To optimally use maintenance resources, the fourth objective is to minimize maintenance time. Thus, the assignment of maintenance slots is penalized with $W_{a,m}$. The cost of maintenance depends on the slot's duration. Additionally, extra costs are incurred for off-hours maintenance slots, as these require changes to mechanics rosters.

- *Task interval utilization*: The following objective is to schedule maintenance tasks at the preferable moment in time to optimize maintenance resources, specifically interval utilization and fleet health. Maintenance planners can schedule a task in multiple maintenance slots. Choosing the most suitable slot is a decision that depends on the task hierarchy and interval utilization. Hierarchy between tasks is achieved again with $C_{Type,t}$, while, when it comes to interval utilization, the following two distinctions are made:

  - Preventive tasks should be scheduled as close as possible to their due date to minimize interval waste.
  - Corrective tasks should be scheduled as soon as possible to not hinder future maintenance plans.

- *Ground time assignment*: The last objective is to optimize ground time allocation for higher fleet availability and improved schedule stability. Fleet availability is maximized, while ground time waste is minimized through $W_{a,g}$. The importance given to fleet availability as opposed to ground time waste is airline-specific. Moreover, using $W_{a,g}$, robust ground arcs are assigned with the intent of improving schedule stability.

4.4.3 Constraints

The constraints of the MILP model are formulated as follows:

$$\sum_{a \in A_f} \delta_{a,f} + \sum_{a \in A_{f_{QTA}}} \delta_{a,f_{QTA}} + \delta_{Canx,f} = 1,\ \forall f \in F \tag{2}$$

$$\sum_{a \in A_u} \delta_{a,u} = 1,\ \forall u \in U \tag{3}$$

$$\sum_{s \in O_{n,a}} \delta_{a,s} - \sum_{s \in T_{n,a}} \delta_{a,s} = FB_{n,a},\ \forall n \in N, a \in A \tag{4}$$

$$\sum_{t \in T_m \cap T_a} \delta_{t,m} \leq Big_M \delta_{a,m},\ \forall a \in A, m \in M_a \tag{5}$$

$$\sum_{a \in A_m} \delta_{a,m} \leq \sum_{t \in T_m} \delta_{t,m},\ \forall m \in M \tag{6}$$

$$\sum_{t \in T_m} L_t \delta_{t,m} \leq L_{max,m},\ \forall m \in M \tag{7}$$

$$\sum_{m \in M_t} \delta_{t,m} + \delta_{Defer,t} = 1,\ \forall t \in T \neg T_{Due} \tag{8}$$

$$\sum_{m \in M_t} \delta_{t,m} = 1,\ \forall t \in T_{Due} \tag{9}$$

$$\sum_{m \in M_{n,l,k}} \sum_{a \in A_m} FTE_m \delta_{a,m} \leq FTE_{n,l,k},\ \forall k \in K, n \in MN, l \in L \tag{10}$$

$$\sum_{a\in A_k}\sum_{m\in M_a\cap M_{d,l}} \delta_{a,m} \leq MaxM_{d,l,k},\ \forall k \in K, d \in D, l \in L \tag{11}$$

$$\sum_{m\in M_a\cap M_d} \delta_{a,m} \leq 1,\ \forall d \in D, a \in A \tag{12}$$

$$\sum_{f\in QTA_d}\sum_{a\in A_f} \delta_{a,f} \leq MaxQTA,\ \forall d \in D \tag{13}$$

$$\frac{1}{|M_t \cap M_{pd} \forall pd \in PD_d|}\sum_{pd\in PD_d}\sum_{m\in M_t\cap M_{pd}} (1+BigM\delta_{t,m})DL_{t,d} > H_f\delta_{a,f},\ \forall d \in D, a \in A, t \in T_a\cap T_d, f \in F_a\cap F_d \tag{14}$$

$$\delta_{a,c_{n,a}} + \delta_{a,b_{n,a}} \leq 1,\ \forall a \in A, n \in AN_a \tag{15}$$

$$\delta_{a,g} \leq \delta_{a,f},\ \forall f \in F, a \in A_f, g \in RG_{a,f} \tag{16}$$

$$\delta_{a,s} \in \{0,1\},\ \forall s \in S, a \in A_s \tag{17}$$

$$\delta_{Canx,f} \in \{0,1\},\ \forall f \in F \tag{18}$$

$$\delta_{t,m} \in \{0,1\},\ \forall m \in M, t \in T_m \tag{19}$$

$$\delta_{Defer,t} \in \{0,1\},\ \forall t \in T \neg T_{Due} \tag{20}$$

Constraints (2) is a coverage constraint that ensures that either rotation f or its quick turnaround copy f_{QTA} are assigned to at most one aircraft or canceled. No other coverage constraints are required because all other arc types are aircraft-specific, i.e. can be assigned only to one tail. Constraints (3) ensure that all fixed maintenance slots are assigned. Flow balance is guaranteed with constraints (4) at every node and for every aircraft.

Using constraints (5) the model guarantees that a task is scheduled to a maintenance slot only if the corresponding aircraft is assigned to that same slot. Constraints (6) prohibit empty slots from being assigned, while constraints (7) prohibit the number of scheduled labor hours in a slot from exceeding that slot's maximum allowed. Tasks with due dates not on the first day of the plan should be scheduled in a maintenance slot or deferred, while tasks going due have to be scheduled. The model enforces the former with constraints (8) and the latter with constraints (9).

The following two constraints ensure that the airline's maintenance schedule is respected. The first, constraints (10), ensure that the available FTE for an aircraft type at a given maintenance location and node is never exceeded. The second, constraints (11), guarantee that on any given day the number of assigned maintenance slots at a given maintenance location for a particular aircraft type do not exceed that maximum in the schedule.

The next constraints, constraints (12), prohibit aircraft from being assigned to more than one maintenance slot a day. Constraints 13) ensure that the maximum number of daily quick turnarounds is not exceeded. Aircraft airworthiness is guaranteed with constraints (14). For every day and every tail, it checks that the days left in the intervals of all the tail's open mandatory tasks going due on or after the day in question are higher than the aircraft health required to operate a given rotation. Each task that does not satisfy this requirement should be scheduled in a maintenance slot before the rotation, otherwise that rotation cannot be operated by the tail in question.

Lastly, constraints (15) and constraints (16) prevent certain movements in the TSN. The former makes sure that aircraft can travel to the maintenance space only directly after the end of rotations they were assigned to. The latter ensures that a rotation's robust ground arc is assigned to the same aircraft that is assigned to the rotation. Constraints (17) - (20) define the decision variables as binary. Finally, the MILP model is solved using a commercial tool.

5 Case study

A case study is performed to understand if the framework can provide realistic and real-time decision support to airline planners the day before operations. Moreover, by comparing the framework's plans against those of the airline, it is assessed if the model can contribute to improved schedule efficiency and stability. Although planning is performed manually by the airline, the framework models the same method currently adopted in practice, thereby it can be a solid benchmark for comparison. This section follows the subsequent structure: first, the test procedure and case study are defined; then, model parameters are determined with the help of trade-offs; subsequently, schedule efficiency is assessed; and lastly, through stochastically generated disruption scenarios, schedule stability is evaluated. The latter analysis focuses only on aircraft late arrivals from late rotations.

5.1 Test procedure

The objective of the case study is to assess if the framework can create realistic plans in real time that improve schedule efficiency and stability. To benchmark the framework's performance, its results are compared to three reference models:

- *Airline*: The schedule produced manually by planners at a major European airline.
- *Fixed Tail Assignment (TA)*: The schedule produced by the MILP model when scheduling maintenance tasks only and fixing the tail assignment to the one created by the airline's planners.
- *Fixed Task Scheduling (TS)*: The schedule produced by the MILP model when assigning tails only and fixing the task assignment to the one created by the airline's planners.

The purpose of the latter two models is to assess the benefits, if any, of integrating tail assignment and maintenance task scheduling. Although airline planners assign tails and schedule tasks in separate teams, they still occasionally communicate and collaborate. Thus, in this respect the airline model is not the perfect benchmark. The Fixed TS and Fixed TA models are constructed by forcing decision variables based on the airline's decisions. Hence, leaving only the other part of the problem to be solved by MILP model. For example, for the Fixed TA model, the airline tail assignment is enforced by restricting decision variables, thereby only allowing the MILP model to change the maintenance task assignment.

The evaluation focuses on plans crafted for seven consecutive planning dates, spanning from 31/10/2023 to 06/11/2023. For each day within this timeframe, the framework generates stand-alone tail and task assignments for a predetermined number of subsequent days. Identical input information as the airline's planners have available at 5 PM of each day, shown in Table 6, is employed across all models. To ensure a fair comparison, only rotations planned by the airline's planners are scheduled, using the same tail restriction rules. Additionally, maintenance is planned seven days ahead the planning date considered, mirroring the practices of the airline's maintenance planners.

The case study airline is a major European single hub-to-spoke airline. It has a fleet of 54 wide-body aircraft of four subtypes and two types. The fleet assignment has been solved in advance and provided as an input. Therefore, type and subtype swaps are not allowed, meaning that rotations can only be assigned to the subtype fleet decided during the fleet assignment. Not all maintenance tasks are part of the case study. Short-interval turnaround maintenance is excluded and assumed to be scheduled during the aircraft's TAT, even if the aircraft's following job is a maintenance slot. Long-interval letter-check maintenance tasks are also excluded. These are planned in fixed maintenance slots, which are forcefully assigned by the MILP model. Moreover, fixed maintenance slots are assumed to be full, hence prohibiting the assignment of any further maintenance task in these slots. These decisions align with practices at the case study airline. Lastly, the maintenance schedule is also provided by the airline.

Planning date	Rotations	Open tasks	Fixed maintenance slots
31/10/2023	77	379	9
01/11/2023	90	379	8
02/11/2023	92	374	9
03/11/2023	93	344	12
04/11/2023	126	345	13
06/11/2023	94	363	12
07/11/2023	82	387	9

Table 6: Overview of the dataset.

5.2 Parameter definitions

As the case study involves comparing the airline's manual plan against that of the framework, parameters and weight functions are defined according to the airline's preferences. The following section outlines the definition and fine-tuning process of the objective weights, which were formulated as cost functions. However, first, the rules used to create a hierarchy between weights are explained. An overview of model parameters is provided in Table 8.

5.2.1 Objective function hierarchy

To reflect the airline planners' priorities, the framework's objectives are characterized by four hierarchies or rules found in practice. These hierarchies help set the weights' orders of magnitude relative to each other. First, airlines aim to cancel as few rotations as possible. Thus canceling a rotation is much more expensive than flying one. Second, deferring a task is more expensive than scheduling it in a maintenance slot, as continuous deferral of maintenance tasks can result in unplanned maintenance and/or too large task backlogs in the long run. The third rule reflects the trade-off between flying and performing maintenance. It states that canceling a flight is worse than deferring tasks. Lastly, ground time is never prioritized over rotations, maintenance activity, and task interval utilization, and is the by-product of the planners' tail and task assignments. Within the objective function, the rules translate as follows: $W_{Canx,f} >> W_{a,f}$, $W_{Defer,t} >> W_{a,m} + W_{t,m}$, $W_{Canx,f} >> W_{Defer,t}$, and $W_{a,f}, W_{a,m}, W_{t,m} >> W_{a,g}$ respectively.

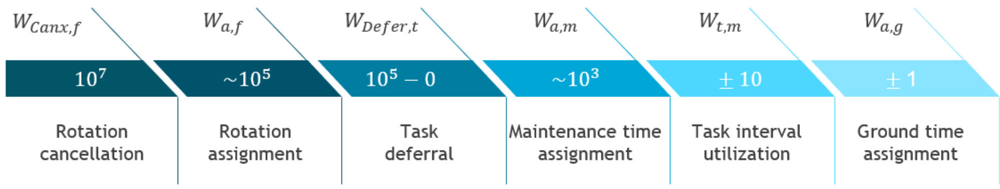

Figure 4: Objective function hierarchy based on airline planning priorities (image inspired by the work of [Van Kessel et al., 2023]).

5.2.2 Network planning costs

Costs associated with network planning are the cost of operating a rotation, and the cost of canceling a rotation. The cost of flying $W_{a,f}$ is based on the rotation's block time BT_f in hours, the aircraft fuel consumption FE_a in kg/hour, and the cost of fuel C_{fuel} in /kg, as follows:

$$W_{a,f} = BT_f FE_a C_{fuel}, \tag{21}$$

Where both the fuel cost and the aircraft-specific fuel consumptions are provided by the case study airline. Given the airline's data and rotations, flying a long-haul rotation typically costs around 10^5. All aircraft within a subtype fleet are assumed to have the same fuel efficiency, as planners ignore individual registration fuel efficiencies.

For quick TA rotations, $W_{a,f}$ is multiplied by a penalty, P_{QTA}. The case study airline allows a maximum of two quick TAs per day by removing 60 minutes of turnaround time at the start of the rotation. The selected penalty value ensures that the additional cost of the rotation is significantly higher than the benefit of scheduling an extra 60 minutes of maintenance or fleet availability. Thus, quick TA is only used to prevent cancellations, as it is in practice.

Canceling a rotation is the last resort decision taken by planers. To represent this in the framework, the cost of canceling a rotation, $W_{Canx,f}$, equals 10^7 or around 100 times higher than the rotation's operating cost. $W_{Canx,f}$ is the same for all rotations as the goal is to prevent cancellations, rather than optimally select which rotations to cancel.

$$W_{Canx,f} = 10^7 \tag{22}$$

5.2.3 Maintenance planning costs

Maintenance planning consists of three parts: task deferral, task scheduling, and maintenance time assignment. The cost of deferring a task, $W_{Defer,t}$, is directly linked to the cost of canceling a flight, $W_{Canx,t}$. This relation dictates the magnitude of $W_{Defer,t}$. It is ensured that the framework will always prefer task deferral over flight cancellation. Only if operating the rotation would lead to more than 100 maintenance tasks being deferred does flight cancellation become the more viable option. However, this is a highly unlikely situation.

Moreover, deferring a task becomes more expensive as its due date approaches. This is because fewer opportunities to schedule it will arise and the risk of Aircraft on Ground (AOG) increases. Therefore, $W_{Defer,t}$ is assumed to follow a linear function that increases with decreasing days left to the task's due date (DL) given by Equation 23. When the days until the task's due date are more than the days-clean target (δ) the deferral cost is zero, to prevent early scheduling of preventive tasks. Afterward, $W_{Defer,t}$ increases linearly, with the deferral cost drastically increasing when the days remaining are fewer than the minimum health target (H_{min}) to ensure that the task is scheduled.

$$W_{Defer,t} = \begin{cases} 10^5 - \frac{(10^5-10^4)DL}{H_{min}} & \text{if } DL \leq H_{min} \text{ and t is mandatory} \\ 10^4 - \frac{(10^4-10^3)(DL-H_{min})}{\delta - H_{min}} & \text{if } H_{min} < DL \leq \delta \text{ and t is mandatory} \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

The cost of scheduling a task has two components: the cost of task interval utilization ($W_{t,m}$) and the cost of maintenance time ($W_{a,m}$). The formulation of the former is inspired by the work of [Van Kessel et al., 2023]. It depends on the task's type and due date (Due_t) and the maintenance slot's start date ($Start_m$). $W_{t,m}$ is defined by a linearly decreasing function for preventive tasks, while a linearly increasing function for corrective tasks. This is to optimize the task interval (I_t) utilization. Following hierarchical rules, weights range from 10 to 0 for preventive tasks and from -10 to 0 for corrective tasks. The weight functions are given respectively in Equation 24 and Equation 25. Moreover, maintenance planners prefer not to schedule tasks too close to their due date (Due_t). Thus, a penalty (W_{AOG}) is implemented when the days left in the task's interval are less than the minimum health target. At last, Table 7 presents the values of $C_{Type,t}$, which define the hierarchy between task types. These are taken from the work of [Van Kessel et al., 2023] and are determined based on the airworthiness impact of the task type.

$$W_{Preventive\ t,m} = \begin{cases} 10\frac{Due_t - Start_m - H_{min}}{I_t - H_{min}} & \text{if } Start_m \geq H_{min} \\ W_{AOG}(1 - \frac{Due_t - Start_m}{H_{min}}) & \text{otherwise} \end{cases} \tag{24}$$

$$W_{Correcitve\ t,m} = \begin{cases} -10\frac{Due_t - Start_m - H_{min}}{I_t - H_{min}} & \text{if } Start_m \geq H_{min} \\ W_{AOG}(1 - \frac{Due_t - Start_m}{H_{min}}) & \text{otherwise} \end{cases} \tag{25}$$

The cost of assigning a maintenance slot depends on the slot's scheduled duration, D_m, and FTE, FTE_m, as well as the cost of one FTE hour. This is the same in the framework, where $W_{a,m}$ is given by Equation 26. The cost of one FTE hour C_{FTE} is a sum of labor, opportunity, and material costs. Off-hours maintenance slots are penalized by multiplying $W_{a,m}$ with $P_{off-hours\ FTE}$. The value of the penalty is chosen ensuring that task deferral remains more expensive.

$$W_{a,m} = FTE_m D_m C_{FTE} \tag{26}$$

The choice of the days-clean target significantly influences task deferral costs, maintenance time, and, consequently, fleet availability. To justify the selection of a days-clean target that optimizes schedule efficiency, a trade-off analysis was conducted. The results are presented in Figure 5. This trade-off analysis used a rolling horizon technique to simulate scenarios spanning five weeks (from 31/07/2023 to 04/09/2023) while varying the days-clean target. This approach allowed for an assessment of the long-term impact of scheduling with a specific target.

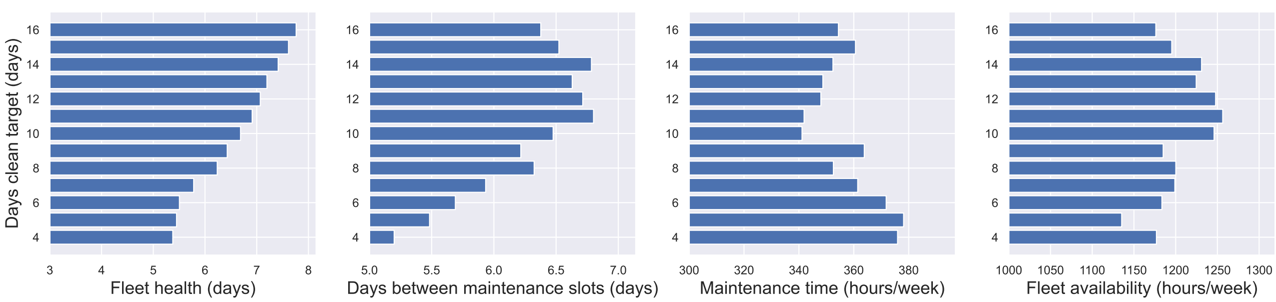

Figure 5: Average fleet health, days between maintenance slots, weekly maintenance time, and weekly fleet availability for several days-clean targets.

As anticipated, the choice of the days-clean target affects schedule efficiency, as it influences fleet health, maintenance time, and fleet availability. A decrease in the days-clean target results in lower fleet health due to reduced aircraft health after a maintenance slot. In the long term, lower fleet health corresponds to an increase

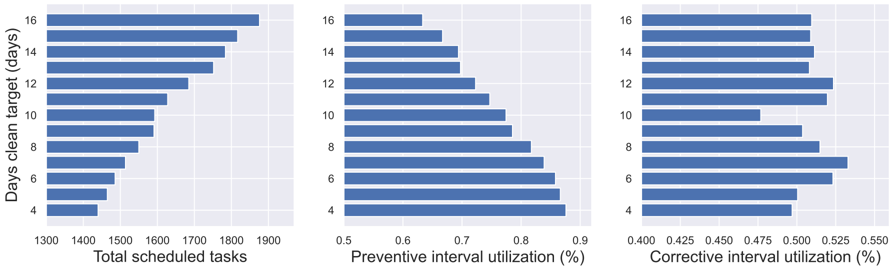

Figure 6: Scheduled tasks, average preventive interval utilization, and average corrective interval utilization for several days-clean targets.

in weekly maintenance time, as aircraft require maintenance more frequently. This, in turn, translates to a decrease in overall fleet availability. Conversely, higher fleet health in the long term results in more maintenance time. This is driven by fewer task deferrals, a higher number of scheduled tasks, and poorer task interval utilization, as illustrated in Figure 6. This leads to higher maintenance frequency necessary to achieve the demanding days-clean target. It is important to note that the utilization of corrective tasks' intervals remains largely unaffected. This is because they are typically scheduled well before the days to their the due date become fewer than the days-clean target. Thereby, the days-clean target has little effect on corrective task's interval utilization.

Schedule efficiency is optimized when planning between 10 and 14 days clean based on fleet health, maintenance time, and fleet availability. For the case study, the days-clean target is set to 10 days, as this falls within the optimal range and aligns with the airline's days-clean target.

5.2.4 Ground time costs

Airline planners are indifferent to where and when ground time is assigned as they prioritize flying and maintenance. Consequently, to reflect its low priority, ground time is associated with the lowest weight ($W_{a,g}$). However, unlike airline planners, the model optimizes ground time allocation, by maximizing fleet availability and schedule stability. Thus, a distinction is made between air space ground arcs (ground time waste) and maintenance space ground arcs (fleet availability), respectively costing 0/hour and 1/hour of ground time.

Schedule stability is implemented by assigning robust ground arcs. These arcs, when assigned, can help reduce disruptions. Hence, they have a cost of -1/hour of ground time, such that the framework actively seeks to assign them whenever possible. This cost formulation priorities schedule stability over fleet availability.

The delay mitigation parameter has direct implication on schedule stability. It affects the length and number of robust ground arcs. As the delay mitigation parameter increases, robust ground arcs are expected to become longer and more. The trade-off results presented in Figure 7 illustrate this dynamic. Nevertheless, there exists a threshold beyond which the arcs become excessively long, resulting in a reduction in their assignment. Ideally, to increase schedule stability, one should maximize the number of used robust ground arcs. Hence, the delay mitigation parameter is set to 95%. Note that this analysis is specific to the case study airline and its delay distributions. Hence, maximum stability is not always achieved with a 95% delay mitigation parameter.

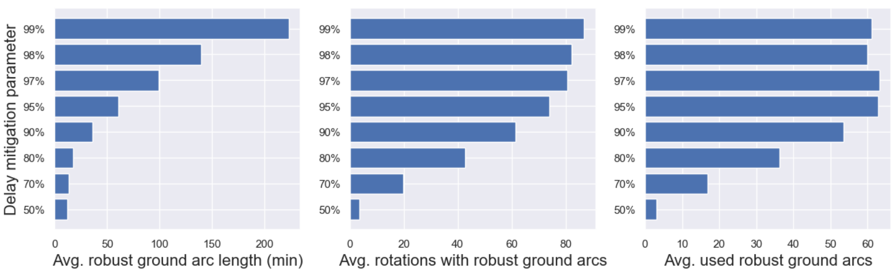

Figure 7: Average robust ground arc length, number of rotations with robust ground arcs, and number of assigned robust ground arcs for several delay mitigation parameters.

5.3 Schedule efficiency analysis

In this subsection, framework's plans are assessed concerning each of the schedule efficiency sub-objectives. The framework creates realistic plans, validated with the airline's planers. In fact, when presented with the frameworks plans alongside their own, without knowledge of which plan belonged to whom, the planners encountered

Task type	$C_{Type,t}$
Requirements	4
MEL	4
Adhoc	2
NSRE	1
Other	1

Table 7: Task type weighting factor taken from the work of [Van Kessel et al., 2023].

Parameter	Value
$MaxQTA$	2
$W_{Canx,f}$	10^7
W_{AOG}	100
C_{fuel}	1 /kg
C_{FTE}	100 /hour
P_{QTA}	2
$P_{off-hours\ FTE}$	2
H_{min}	3 days
Days-clean target, δ	10 days
Delay mitigation parameter	95%

Table 8: Overview of framework parameters

difficulty in distinguishing between the two. All rotations planned daily by the airline are assigned by the model. Only the Fixed TA model results in 12 cancellations as aircraft airworthiness could not be guaranteed. Hence, suggesting that the framework offers cancellation benefits over a fully separate planing approach. Furthermore, the average solution time of the framework is around 250 seconds for each day analyzed. Arguably less than what an airline planner takes to assign tails and schedule maintenance tasks. Hence, the model could be used in an operational environment.

The framework increases fleet availability by reducing maintenance time. Figure 8 summarizes the fleet availability and maintenance time planned in the subsequent one, two, and three days of a plan, respectively denoted by T-1, T-2 and T-3. Thanks to the framework's aircraft-specific maintenance slots, the average maintenance labor utilization increases from the airline's 67% to 81%, reducing wasted maintenance time. Consequently, compared to the airline, the framework plans on average 15 additional hours of fleet availability on the day of operations, or an increase of 10%. Notably, the minimum fleet availability planned by the framework is significantly higher than the airline's. Thus, the framework proves especially beneficial in dense plans, when fleet availability is limited but highly desired. However, there are operational constraints, such as mechanics skills assignment and sudden labor and/or towing services shortages, that are outside the scope of the framework. These could increase maintenance time and consequently decrease fleet availability.

The framework's ability to integrate tail assignment and maintenance task scheduling helps reduce maintenance time and increase fleet availability. Fixing the tail assignment restricts maintenance task scheduling in the Fixed TA model, resulting in additional ground time waste. As a result, average fleet availability falls even when maintenance time is decreased. The Fixed TS model implements the same task and slot assignment as the airline, and hence plans the same amount of maintenance time. But compared to the airline, it significantly increases fleet availability. This highlights the advantages of employing the optimization model for the tail assignment. Additionally, compared to the framework, it manages to slightly increase average fleet availability. Given that the Fixed TS approach plans more maintenance time, this may appear counterintuitive. But because in the framework task interval utilization is given priority over ground time assignment, the framework wastes more ground time in order to optimize task interval utilization. Consequently, this lowers fleet availability. Nonetheless, the framework's emphasis on task interval utilization will in the long term save maintenance time, increase fleet availability, and ultimately boost schedule efficiency.

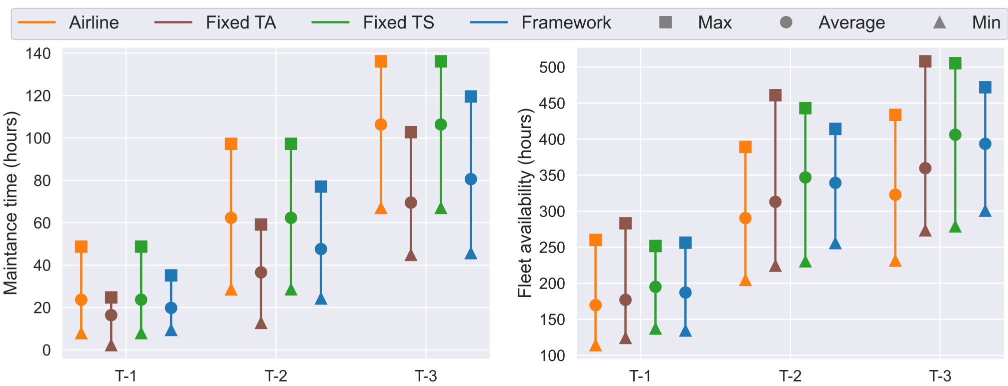

Figure 8: Planned fleet availability and maintenance time for T-1, T-2, and T-3 for all four models.

The location of fleet availability within the schedule is just as crucial as its quantity. Table 9 displays the distribution of ground time waste divided into short, medium and long duration based on the airline's preferences. The framework increases medium duration ground time waste because it prioritizes robust ground arcs and task interval utilization over minimizing ground time waste. This explains why the Fixed TS model, which reduced medium and long duration ground time waste, manages to increase fleet availability over the framework. Still, the framework reduced long duration ground time waste in comparison to the airline. This is particularly advantageous as long duration ground time waste represents the largest loss of flexibility. Moreover, Table 10 shows that the framework improves ground time allocation after rotations. The framework allocates ground time more evenly, represented by increased rotations followed by ground time between 2 and 8 hours long. Additionally, it increases rotations followed by more than 2 hours of ground time, which is useful to mitigate arrival delays. Both are preferences of the airline's planners.

	(0, 2]	(2, 12]	> 12
Airline	40.9%	36.4%	22.7%
Fixed TA	42.4%	31.8%	25.8%
Fixed TS	45.6%	35.3%	19.1%
Framework	39.5%	39.5%	21.0%

Table 9: Distribution of the ground time waste divided into short ((0,2]), medium ((2,12]), and long (>12) duration in hours for all four models.

	(2, 8]	> 2
Airline	23.7%	45.2%
Fixed TA	21.8%	47.7%
Fixed TS	32.4%	54.9%
Framework	32.4%	55.9%

Table 10: Distribution of the ground time planned after rotations in hours for all four models.

Although the model plans less maintenance time than the airline, on average 17% less, it schedules a similar amount of maintenance tasks each planning day. This indicates that the framework is more efficient at scheduling maintenance tasks. Indeed, it improves slot labor utilization and schedules preventive tasks closer to their due date, improving their interval utilization, as shown in Figure 9. However, no interval improvements are obtained for corrective tasks. This is likely due to their lower priority set with $C_{Type,t}$. The model's resulting improvements in the utilization of maintenance resources lowers airline operating maintenance costs. However, the framework's higher labor utilization could harm work packages' completion rate, resulting in more spilled tasks and additional unplanned maintenance time and costs.

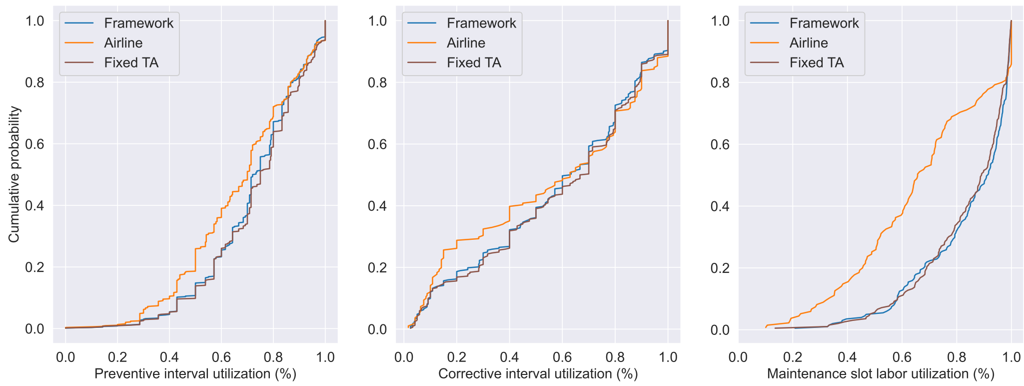

Figure 9: Cumulative distributions of preventive maintenance tasks' interval utilization, corrective maintenance tasks' interval utilization, and maintenance slots' labor utilization. Note that results for the Fixed TS model are not shown, as they are exactly the same as the Airline model.

Figure 10 displays the fleet health planned by each model for seven days from the planning date. The framework and the airline planers plan an almost identical average fleet health from D1 to D3. However, after D3 the framework plans more health. This is even more evident when comparing the framework with the Fixed TA model. Fixing the tail assignment greatly restricts maintenance task scheduling, deteriorating fleet health. A healthier fleet offers more flexibility to network operations as it means less maintenance time is needed in the subsequent days. Moreover, the decrease in the range between max and min planned fleet health, suggests that the framework is more consistent with the task assignment.

The case study has shown that the framework improves schedule efficiency over the airline. Firstly, like the airline, it does not cancel any rotation. Higher fleet availability increases useful buffer time for aircraft swapping, making the framework's plans easier to recover. Moreover, improved labor utilization reduces wasted maintenance time and wasted labor resources, reducing costs in the long term. Maintenance time, and thereby costs, are further reduced by the framework's higher fleet health and preventive task interval utilization. The framework's efficiency gains are attributable to the integration of tail assignment and maintenance task scheduling. Although the Fixed TS model slightly increases fleet availability, worse fleet health and utilization of maintenance resources will deteriorate this advantage in the long-term. Moreover, in the Fixed TA model restricting

Figure 10: Planned fleet health for the seven days from the planning date.

task scheduling results in cancellations, less fleet availability and worse fleet health.

5.4 Schedule stability analysis

In this subsection, the framework's impact on schedule stability is evaluated. To delve into schedule stability, plans from all four models are subject to aircraft late arrival disruptions. These disruptions are derived from historical rotation arrival delay data, including early arrivals, from August 2014 to November 2023 provided by the case study airline. The data set contains only instances of rotations inbound to the airline's hub. Rotations instances are grouped based on their departure airport and season (winter or summer), excluding flight numbers and/or flown aircraft subtypes as differentiators.

Each plan is tested with 100 different disruption scenarios. For each scenario, all planned rotations have a unique arrival delay randomly generated from their corresponding arrival delay distributions. On a given planning date, all four models are assessed with the same 100 disruption scenarios. To ensure a fair comparison, no maintenance disruptions are introduced, as the framework does not account for maintenance uncertainty while planning.

The aim is to reduce disruptions arising from late arrivals, subsequently reducing the necessity for schedule adjustments. Additionally, minimizing propagated delay is a critical factor in schedule stability. Figure 11 summarizes the number of disrupted LOFs and propagated delay in T-1 and T-2 for all four models. A LOF is disrupted when the arrival delay extends into the aircraft's subsequent job, be it a rotation or a maintenance slot.

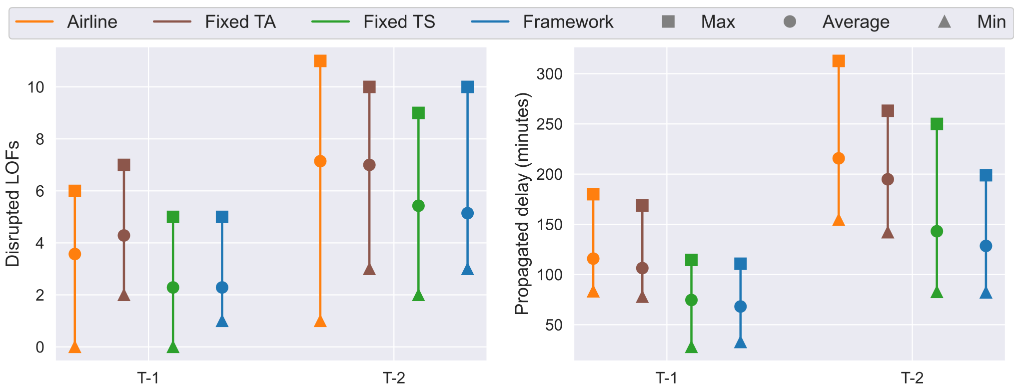

Figure 11: Disrupted LOFs and propagated delay in the subsequent day (T-1) and subsequent two days (T-2) for all four models.

On average, the framework creates plans that result in fewer disruptions and less propagated delay. Hence, it improves schedule stability. Compared to the airline, average disrupted LOFs are reduced by 42% on the day of operations, while 43% in the subsequent two days. Importantly, the maximum number of disruptions is lower, thereby reducing planner's maximum recovery workload. Moreover, the average propagated delay is decreased by 48 and 88 minutes in T-1 and T-2 respectively. The Fixed TS model comes closer to the framework's performance as it still has significant freedom in the ground time allocation. On the other hand, fixing the tail assignment, done in the Fixed TA model, greatly deteriorates stability. This is because most buffer time is pre-assigned. Lastly, although the framework's minimum number of disruptions is higher, the minimum propagated delay is lower. This hints at the model having shorter delays.

The distribution of propagated delay is presented in Table 11. The number of delayed subsequent jobs is reduced from the airline's 9.0% to less than 5.4%. Moreover, delays are reduced in each possible delay range, with the slight exception of delays between 30 and 60 minutes. Especially beneficial is the reduction of delays greater than 60 minutes, which are the most detrimental and the hardest to recover. Thus, framework plans require fewer adjustments, due to fewer disruptions, and are easier to recover, due to shorter delays. This diminishes recovery costs and has indirect revenue benefits from more satisfied passengers. Importantly, these advantages are not only obtained over the airline, but also over the Fixed TS and Fixed TA models. Hence this justifies the benefits of integrating tail and task assignment from a literature perspective.

	> 0	(0,30]	(30,60]	(60,90]	> 90
Airline	8.96%	5.46%	1.40%	1.82%	0.28%
Fixed TA	7.56%	4.18%	1.75%	1.08%	0.54%
Fixed TS	6.42%	3.35%	1.40%	1.40%	0.28%
Framework	5.43%	2.38%	1.59%	1.19%	0.26%

Table 11: Distribution of propagated delay in minutes.

As expected the framework decreases the proportion of expected long delays. This is shown in Table 12, which presents the distribution of the expected propagated delay (EPD) greater than zero. EPD is not influenced by the disruptions scenarios but is solely based on the arrival delay distribution of rotations and the scheduled ground time following them. The framework achieves no EPD greater than 30 minutes, contrasting with the other three models. Despite the model's improvement in the EPD distribution, the framework's average EPD, while 38% lower than the airline's, is only reduced by 5.2 minutes. This is not a significant reduction. Further testing of the model in a denser network, hence with a higher potential for delay propagation, would be intriguing.

	(0,15]	(15,30]	(30,60]	> 60	avg. EPD
Airline	75.3%	16.9%	3.4%	4.9%	13.6 min
Fixed TA	75.3%	16.4%	2.7%	5.5%	13.9 min
Fixed TS	81.3%	14.6%	2.1%	2.1%	10.2 min
Framework	86.5%	13.5%	0%	0%	8.4 min

Table 12: Distribution of the expected propagated delay (EPD) when disruptions occur in minutes.

Although the framework improves network stability, these improvements must not compromise maintenance stability. Table 13 displays the number of new faults arriving in T-1 and T-2 that are deferred. These deferrals occur when the faults cannot be scheduled within the current plan. This may happen either because the aircraft was not assigned a maintenance slot before the fault's due date or because the assigned slot/slots is/are already full. Unlike network stability, the framework does not incorporate any forms of robustness in the task assignment, such as additional maintenance time buffers for unexpected non-routine work during the slot. Thus, the primary objective is not to outperform the other models. Rather it is to achieve a similar number of deferrals. This is indeed the case, as average new fault deferrals are the same in T-1 and only one additional fault is deferred in T-2.

	T-1 deferred faults	T-2 deferred faults
Airline	15	25
Fixed TA	15	25
Fixed TS	15	25
Framework	15	26

Table 13: Average deferred new faults for the subsequent day (T-1) and subsequent two days (T-2) for the four models.

6 Conclusions and recommendations

This paper presented a decision support framework for integrated tail assignment and maintenance task scheduling on the day before operations. Testing the framework within a major European airline case study proved that it is able of providing real-time decision support in an operational environment. Evaluating the framework's plans against those of the airline, showed the framework can improve schedule efficiency and stability. This was done respectively by increasing fleet availability, improving fleet health, and reducing maintenance time, and reducing late arrival disruptions. Secondly, the case study showed the value of simultaneously assigning tails and maintenance tasks. When either the task or tail assignment were fixed, the remaining part of the problem was constrained by it, resulting in overall less efficient and less stable plans.

A novel approach to maintenance slot modelling was proposed. Instead of using generic fixed duration maintenance slots, suitable for the entire fleet, the framework creates tail-specific slots, tailored to aircraft's maintenance needs. This had advantages, such as improved maintenance planning efficiency resulting from less maintenance time waste. Specifically, maintenance time was reduced by 17% as a result of labor utilization increasing from on average of 67% to 81%. Consequently, this benefited fleet availability, which on average increased by 10% on the day of operation. This has benefits in literature (a new approach to maintenance slots modeling) and in practice (improved schedule efficiency results in lower operating costs).

Simultaneous optimization of schedule efficiency and stability was made possible by modeling the schedule with an innovative TSN. The TSN uses two distinct spaces, one dedicated to network while the other to maintenance activities. Priority was given to the flexibility enhancing fleet availability over ground time waste. Moreover, stability was introduced with additional ground arcs positioned after flights that, if assigned, allow the creation of flight specific buffers. These buffers were based on flights historical arrival delay distributions. Owing to this, the framework reduced late arrival disruptions by 42% and propagated delay by 48 minutes on the day of operations. According to [Eurocontrol, 2004] delay cost airlines on average EUR 0.30 per minute per passenger. Assuming an average of 300 passengers on long-haul flights, the framework can save the airline more than EUR 1.5 millions per year.

Although the framework's results are promising, the case study had limitations. The case study involved only the airline's long-haul single hub-to-spoke network. Nonetheless, small adaptations can be made to the framework to accommodate short-haul operations and different network types, such as point-to-point. The latter would require slight modification to the TSN by adding an additional air space per airport and an additional maintenance space per airport at which maintenance can be performed. Additionally, only short-term maintenance checks and tasks were considered. A valuable improvement would be to treat fixed maintenance slots as additional maintenance opportunities for task assignment, given that space permits it, and add minor flexibility to their start times.

Further improvements can be made to the framework itself. The research focused exclusively on arrival delays, neglecting other potential sources of uncertainty, such as unexpected labor within maintenance slots and AOG situations. The former could be especially problematic, as the model leads to densely packed slots, increasing the risk of delayed slots. Consequently, there is a need to incorporate robustness against late maintenance slots, potentially through the use of similar robust ground arcs based on historical task duration data. Moreover, currently fleet health is modelled indirectly. Considering its importance as an indicator of airline maintenance planning efficiency, it would be worthwhile to model it with unique decision variables and a specific term in the objective function of the MILP model.

Despite these limitations this research provides value to current literature and an airline. To the best of my knowledge, this research presents the first example of a robust framework that integrates tail assignment and maintenance task scheduling in details. The computational capabilities of the framework could offer a resolution to airline's manual and separate tail and task assignment approach. In an operational environment, the framework has demonstrated its ability to improve schedule efficiency and stability, creating value for an airline.

References

[Abara, 1989] Abara, J. (1989). Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19(4):20–28.

[Ageeva, Y. V., 2000] Ageeva, Y. V. (2000). Apprroaches to Incorporating Robustness Into Airline Schedules. Technical report.

[AhmadBeygi et al., 2010] AhmadBeygi, S., Cohn, A., and Lapp, M. (2010). Decreasing airline delay propagation by reallocating scheduled slack. *IIE Transactions*, 42(7):478–489.

[Ahmed et al., 2017] Ahmed, M. B., Ghroubi, W., Haouari, M., and D., S. H. (2017). A hybrid optimization-simulation approach for robust weekly aircraft routing and retiming. *Transportation Research Part C: Emerging Technologies*, 84:1–20.

[Aloulou et al., 2010] Aloulou, M. A., Haouari, M., and Mansour, F. Z. (2010). Robust aircraft routing and flight retiming. *Electronic Notes in Discrete Mathematics*, 36:367–374.

[Barnhart et al., 1998] Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., and Shenoi, R. G. (1998). Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):208–220.

[Burke et al., 2010] Burke, E. K., Causmaecker, P. D., Maere, G. D., Mulder, J., Paelinck, M., and Berghe, G. V. d. (2010). A multi-objective approach for robust airline scheduling. *Computers Operations Research*, 37:822–832.

[Cadarso and Marín, 2011] Cadarso, L. and Marín, . (2011). Integrated robust airline schedule development. *Procedia Social and Behavioral Sciences*, 20:1041–1050.

[Clausen et al., 2010] Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industryconcepts, models and methods. *Computers Operations Research*, 37(5):809–821.

[Deng and Santos, 2022] Deng, Q. and Santos, B. F. (2022). Lookahead approximate dynamic programming for stochastic aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 299(3):814–833.

[Deng et al., 2020] Deng, Q., Santos, B. F., and Curran, R. (2020). A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281(2):256–273.

[Dunbar et al., 2014] Dunbar, M., Froyland, G., and Wu, C. L. (2014). An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers Operations Research*, 45:68–86.

[Eurocontrol, 2004] Eurocontrol (2004). Evaluating the true cost to airlines of one minute of airborne or ground delay. Technical report.

[EUROCONTROL, 2023] EUROCONTROL (2023). All-causes delays to air transport in europe - quarter 2 2023. Technical report.

[Hane et al., 1995] Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., and Sigismondi, G. (1995). The fleet assignment problem: 1 solving a large-scale integer program. Technical report.

[Kang, L. S., 2004] Kang, L. S. (2004). Degradable Airline Scheduling: an Approach to Improve Operational Robustness and Differentiate Service Quality. Technical report.

[Lagos et al., 2020] Lagos, C., Delgado, F., and Klappa, M. A. (2020). Dynamic optimization for airline maintenance operations. *Transportation Science*, 54(4):998–1015.

[Lan et al., 2006] Lan, S., Clarke, J. P., and Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1):15–28.

[Lapp and Cohn, 2012] Lapp, M. and Cohn, A. (2012). Modifying lines-of-flight in the planning process for improved maintenance robustness. *Computers Operations Research*, 39(9):2051–2062.

[Marseguerra and Zio, 2000] Marseguerra, M. and Zio, E. (2000). Optimizing maintenance and repair policies via a combination of genetic algorithms and monte carlo simulation. *Reliability Engineering and System Safety*, 68(1):69–83.

[Papakostas et al., 2010] Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., and Chryssolouris, G. (2010). An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48(4):604–612.

[Rosenberger et al., 2004] Rosenberger, J. M., Johnson, E. L., and Nemhauser, G. L. (2004). A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3):357–368.

[Sarac et al., 2003] Sarac, A., Batta, R., and Rump, C. M. (2003). A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, 175(3):1850–1869.

[Shahmoradi-Moghadam et al., 2021] Shahmoradi-Moghadam, H., Safaei, N., and Sadjadi, S. J. (2021). Robust maintenance scheduling of aircraft fleet: A hybrid simulationoptimization approach. *IEEE Access*, 9:17854–17865.

[Shaukat et al., 2020] Shaukat, S., Katscher, M., Wu, C. L., Delgado, F., and Larrain, H. (2020). Aircraft line maintenance scheduling and optimisation. *Journal of Air Transport Management*, 89.

[Smith and Johnson, 2006] Smith, B. C. and Johnson, E. L. (2006). Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516.

[Sriram and Haghani, 2003] Sriram, C. and Haghani, A. (2003). An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*, 37(1):29–48.

[Van Kessel et al., 2023] Van Kessel, P. J., Freeman, F. C., and Santos, B. F. (2023). Airline maintenance task rescheduling in a disruptive environment. *European Journal of Operational Research*, 308(2):605–621.

[Vink et al., 2020] Vink, J., Santos, B., Verhagen, W., Medeiros, I., and Filho, R. (2020). Dynamic aircraft recovery problem - an operational decision support framework. *Computers Operations Research*, 117.

[Weide et al., 2022] Weide, T. V. d., Deng, Q., and Santos, B. F. (2022). Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers Operations Research*, 141.

[Witteman et al., 2021] Witteman, M., Deng, Q., and Santos, B. F. (2021). A bin packing approach to solve the aircraft maintenance task allocation problem. *European Journal of Operational Research*, 294:365–376.

[Z. Liang and Chaovalitwongse, 2015] Z. Liang, Y. Feng, X. Z. T. W. and Chaovalitwongse, W. A. (2015). Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B: Methodological*, 78:238–259.

II

Literature Study
previously graded under AE4020

Introduction

In the introduction of this paper, the background to the problem is presented first in chapter II, followed by the research objectives and questions respectively in chapter II and chapter II.

Background

Poor weather conditions, airport congestion, unavailable staff, and unplanned mechanical failures are disruptive and costly to airline operations. According to EUROCONTROL [14], ~55% of flights arrived more than 5 minutes late in Europe in Q3 of 2022, while according to the Bureau of Transport Statistics [49] ~20% of flights were delayed, ~3% canceled, and no airline achieved an on-time performance greater than 90% in the USA in 2022. Disruptions to airline schedules are not only common, but also expansive, e.g., additional crew overtime, increased fuel usage, and passenger compensations and accommodations are all additional expenses associated with delays and cancellations. In 2007 disruptions costed USA airlines ~$33B [40], which can only be higher today.

Due to large costs related to disruptions, efficient disruption management, is very important to airlines. Airlines can tackle disruptions proactively and/or reactively. Proactive disruption management deals with constructing disruption-resistant schedules by optimizing robustness and operating costs. In contrast, reactive disruption management deals with schedule recovery after disruptions occur by optimizing for recovery costs. Both proactive and reactive disruption management are solved sequentially for aircraft, crew and passengers, or as an integration of any of the three.

Today airline operators manually mitigate disruptions by altering aircraft schedules, crew schedules, and passengers' itineraries. However, for large airlines with extensive networks and complicated schedules, an operator cannot find manually an optimal solution. Therefore, leading airlines are looking for fast operating tools for schedule recovery and strategic models to create robust schedules, all with the aim of facilitating disruption management. Since 1984 over 110 disruption management articles have been published, with over 50% in the last 10 years [35].

Research objective and context

After the COVID-19 pandemic, labor shortages have disrupted airline schedules and passenger itineraries. Airline hope to return to improve disruption management, both in the schedule construction phase, known as proactive disruption management, and during the operation phase, known as recovery disruption management. In the proactive phase, airlines can introduce stability, by designing schedules that are insensitive to disruptions, introduce flexibility, by designing schedules that are easily adjusted if disruptions occur, or a combination of both. This will result in robust, or smart, schedules that are easy to recover in the reactive phase. Given the broad scope of the problem, this thesis will only explore possibilities for smart aircraft schedules and aircraft recovery, hence disregarding passenger and crew scheduling and recovery. Therefore the first objective of this literature review is:

Investigate, from literature, possibilities for airlines to develop smart aircraft schedules with in-built stability and/or flexibility for improved aircraft recovery.

Today, airlines treat network and maintenance as separate entities, although they are highly dependent on each other and often in conflict. Therefore, the respective schedules are often optimal individually, but suboptimal as a whole. Modeling network and maintenance simultaneously would not only lead to an operations plan that is more optimal as a whole but also to more robust and easier-to-recover schedules. A maintenance schedule can provide robustness to a network schedule, for example, through high aircraft health, and vice versa. At the same time, maintenance operations, such as slot swaps, can provide relief to network schedules and improve the options an airline has to recover from disruptions. Henceforth, the second objective of this work is:

Explore how maintenance scheduling is integrated into aircraft disruption management literature, and how maintenance practices can be used to improve schedule robustness and aircraft recovery.

Research questions

Based on these research objectives, the research questions and relevant sub-question that will be addressed are:
How have stable and/or flexible airline schedules for improved schedule robustness and aircraft recovery been created in literature?

1. How are network planning and disruption management approached by airlines and researchers?
2. Which smart schedule strategies have been neglected?
3. What strategies best promote schedule stability and/or flexibility and thus result in better aircraft recovery?
4. What objectives are sought in literature when creating smart schedules?
5. How has aircraft recovery been modeled in literature?

How has maintenance planning been integrated into aircraft disruption management literature to improve schedule robustness and aircraft recovery solutions?

1. How is maintenance planning approached by airlines and researchers?
2. Which approaches have been used for maintenance disruption management (proactive & reactive)?
3. What role has maintenance played while creating robust network schedules?
4. What role has maintenance played in aircraft recovery?

Report structure

After establishing this study's objectives and questions, the problem can be tackled. The first chapter (chapter II) gives a summary of typical disruptions, an examination of how they are managed, and a literature-based breakdown of the problem, highlighting the differences between proactive and reactive methods. The process for network and maintenance planning is covered in the chapter that follows (chapter II). The topic of proactive disruption management in chapter II and reactive disruption management in chapter II are best started with these two chapters. The analysis of pertinent literature is completed with chapter II.

Airline disruption management

This chapter will examine the procedure for managing airline disruptions. In chapter II, which opens the chapter, common types of network and maintenance disruptions are covered. We will then examine how airlines carry out their disruption management approach in chapter II. The disruption management problem will be broken down in this chapter's final section, Figure II, using knowledge gained from previous research.

Disruptions to airline schedules

Airline network schedules are rarely, if ever, implemented exactly as planned. Two major disruption sources impeded the implementation of schedules as planned [5]:

- *Airline resources shortages* are caused by crew unavailability, due to illness or missed connection; aircraft unavailability, due to unplanned maintenance or late arrival of the previous flight; and longer than expected turnaround time, due to longer boarding and/or deboarding or unavailability of turnaround services like catering and baggage handlers.
- *Airport and airspace capacity shortages* are caused by extreme weather events, over-congested airports, and unexpected accidents.

When not enough slack is incorporated into a schedule, these disruptions will result in irregular operations. Under irregular operations airport and airline resources, such as crews, aircraft, and landing slots, become unavailable, and thus the planned schedule is not operable.

Like network schedules, maintenance schedules also suffer disruptions that prevent them from being implemented as planned. Disruptions to maintenance schedules include [52]:

- *Irregular arrival of new maintenance tasks* can be caused by faults during flight or postponement of faults found during preventive work.
- *Adjustments to the slot schedules* are caused by the late arrival of an aircraft, hence shortening the duration of the slot or even resulting in slot cancellation, or aircraft early dismissal to facilitate network disruptions.
- *Maintenance resources shortages* are the result of decreased available workforce due to a unforeseen absence, or lack of material or tooling to complete a task due to late delivery.

Although disruptions often originate locally, because aircraft, crews and passengers are all interconnected, disturbances propagate beyond the event that caused them, affecting later flights and their crews and passengers [46]. For example, disruption to one flight leg in the morning can have an impact throughout the day, and cause flights to be canceled in the evening.

Disruption management at the OCC

To deal with the complex disruption management process presented in the previous section, airlines have established Operation Control Centers (OCC) [39]. The role of the OCC is to manage safe and effective aircraft and crew operations, exchange information with regulators, and manage passengers. To do so, there are various functions at the OCC, such as aircraft controllers, crew trackers, and customer service coordinators [39][46].

Every day, to manage disruptions, operators at OCCs perform trade-offs between airline operating costs and passenger disruption costs to select the best recovery strategy [6]. The first objective of disruption management, airline operating costs, consists of the minimization of delays, tail swaps, cancellations, and the use of reserves. The second objective, passenger disruption costs, is comprised of hard costs, such as rescheduling and accommodation costs, and soft costs, such as passenger loss if the customer promise (i.e. get passengers and their luggage to their destination on time) is not delivered [35].

Kohl et al. [46] present the disruption management process as an ongoing operation, rather than a single problem (see Figure 1). At OCCs, workers constantly monitor operations, composed of planned (e.g., tail and crew assignment) and actual events (e.g., a missing aircraft due to late arrival on the previous flight). Actual events flow with high density, often more than one message per second, and some can indicate discrepancies from planned events and thus raise questions. In the case of such an event, operators identify recovery options and evaluate them from the aircraft, crew, and passenger perspectives, leading to the objective trade-off previously introduced. For example, from the passengers' perspective, delaying an outbound flight to ensure the possibility of transfers is preferred. However, one must also evaluate this option from the crew and aircraft perspective (e.g., satisfaction of crew duty requirements). With feasible, optimal options a decision is made and finally implemented.

Work at the OCC consists of solving disruptions as they occur, in a process known as reactive disruption management. However, research has also been performed on the design of robust schedules, with more stability and/or flexibility implemented, to ensure that the planned itineraries can be maintained or recovered with more ease. This process is known as proactive disruption management [46]. Like reactive disruption management, proactive disruptive management has still a lot of potential.

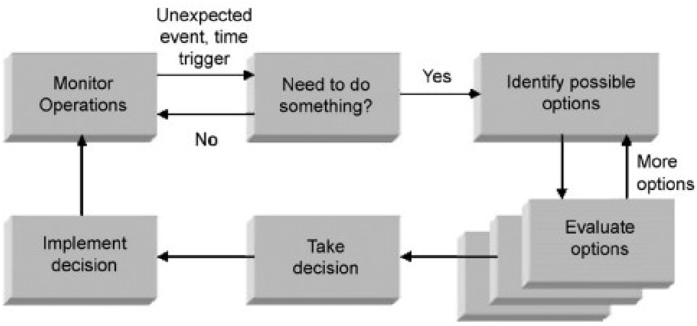

Figure 1: High-level view of the disruption management process [46].

Problem breakdown

This section presents a breakdown of the airline disruption management problem based on literature. Figure 2 depicts this breakdown. Papers often cover only one or two elements of the problem (e.g., absorption robustness, or aircraft and crew recovery). Disruption management in literature reflects the sequential process implemented by most airlines [39]. The first steps are aircraft scheduling and aircraft recovery, which involves aircraft routing/rerouting and/or flight cancellation and delay. The second step, crew scheduling and crew recovery is to create and adjust crew plans by assigning/reassigning flight personnel or using reserves. Finally, the third step consists of developing flight and/or accommodation plans for passengers and disrupted passengers. In literature, these subproblems are either addressed individually or integrated as one. **However, this thesis work will focus solely on the aircraft side of the problem.**

Airlines can solve disruptions proactively or reactively. To proactive distribution management, there are two approaches: absorption and recovery [22]. Both address the problem and create schedule robustness differently. *Absorption robustness* promotes stability as it aims to keep schedules feasible when disruptive events occur. On the network side, it does so by retiming flights, creating delay robust routing, and/or integrating recovery into network planning. On the maintenance side, stability is achieved through buffers and stochastic analysis of tasks durations in a process called maintenance uncertainty scheduling. On the other hand, *recovery robustness* promotes flexibility as it aims to design schedules that are easily adapted when disruptive events occur. It does so by facilitating aircraft swapping, creating station purity, using flexible maintenance slots, i.e. slots that can be assigned to all registrations, and/or planning in sub-networks. A review of proactive disruption management literature is presented in more detail later in chapter II.

When a smart schedule is built using proactive disruption management techniques, one can test its performance in a disruptive environment using reactive disruption management practices. Literature reviews on reactive disruption management (Hassen et al. [35], Clausen et al. [22] and Su et al. [66]) show that many have attempted the Aircraft Recovery Problem (ARP). The same actions implemented at airlines' OCCs for aircraft recovery are used in literature. These include flight delay and cancellation, aircraft swap, and maintenance slot swap. From the maintenance perspective, when disruption occurs, such as new incoming tasks and/or resource shortages, task, and slot rescheduling are common practices. A review of reactive disruption management literature is presented in more detail later in chapter II.

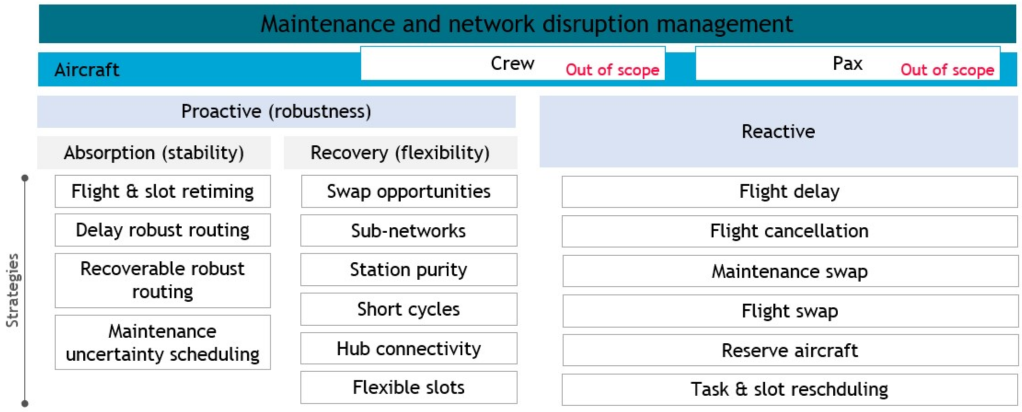

Figure 2: Breakdown of the airline disruption management problem.

Network and maintenance planning

Techniques for operations planning are described in this chapter. Starting with network planning, covered in chapter II, then maintenance planning, covered in chapter II. This is followed by chapter II, in which optimization techniques used to solve aircraft routing and/or maintenance scheduling problems are discussed. Lastly, in chapter II an overview of various network types used in pertinent literature are provided. The chapter is concluded with Figure II.

Network planning

Flight schedules are the core planning elements of an airline. It is composed of flights, including their departure and arrival times and airports, day of operations, and aircraft registrations, with the objective to match the airline's resources to passenger demand. Creating a schedule is a complex multi-part optimization problem, which is thus split into subproblems that are easier to solve individually [17]. Starting with flight schedule generation, destinations and flight times are decided from passenger demand data. This is followed by aircraft scheduling, containing fleet assignment and aircraft routing. Lastly, crew scheduling is performed.

In this section, an overview of the fleet assignment and the maintenance routing problems is presented. These two network scheduling subproblems are most relevant when it comes to disruption management. Note that one can begin disruption management practices already in the schedule design phase, proposed to an extent by Cadarso and Marín [11]. However, schedule design is already very complex, and in my opinion, one needs to first create a schedule that is at the forefront profitable and meets passenger demand, and then one can address disruptions, rather than starting from a schedule that is optimized for disruption management.

Fleet assignment problem

The Fleet Assignment Problem (FAP) is a tactical problem that deals with assigning a fleet type to each flight leg. It has been covered extensively in literature, including Abara [3] and Hane et al. [7]. Both models maximize aircraft profit while ensuring the feasibility of flight coverage, flow balance, and fleet count constants; however, Abara [3] used a time-space network, while Hane et al. [7] employed a connection network. Despite the fact that both models are capable of solving networks with more than 2500 daily flights, Hane et al.'s [7] technique does not account for aircraft rotations.

By including maintenance and personnel constraints, Clarke et al. [33] enhance earlier FAP work. FAP research presented so far considered only static demand. Barnhart et al. [9] present considerable advancement by suggesting a daily itinerary-based model that incorporates passenger flow and takes spilling and recapturing into account. In stochastic demand scenarios, their heuristic solution outperforms conventional models.

Maintenance routing problem

The Maintenance Routing Problem (MRP), like the FAP, is a tactical problem that deals with assigning flight legs and/or Lines of Flights (LOFs) to individual aircraft registrations. However, the problem's name is deceiving as the MRP is mostly focused on network operations, with limited attention given to maintenance. Nonetheless, maintenance has progressively received more thinking within the MRP. While early work only imposed maintenance at fixed intervals, later work include aircraft-specific requirements and tasks.

Aircraft routing within the MRP is often performed with a LOF-based approach, in which aircraft are assigned to predetermined strings of flight legs. Barnhart et al. [8] use a LOF-based approach to assign aircraft to flight strings that start and end at maintenance stations and hence considered maintenance feasibility constraints. The MILP model minimized the cost of assigning aircraft to LOFs, ensuring continuity and flight coverage constraints. Sriram and Haghani [62] developed a similar MILP model that considered only short and medium-haul flights and overnight maintenance. Both models are capable of finding solutions to large networks with more than 1000 daily flights and 50 aircraft. The challenge of LOF-based approaches is computing optimal and feasible flight strings. Solving the MRP with all feasible LOFs for large networks with many flights would be practically impossible. Therefore authors use methods to simplify their solution space, for example, a branch-and-price algorithm like Barnhart et al. [8] or a heuristic method like Sriram and Haghani [62].

Both works treat maintenance as a fixed interval activity, while later research considers maintenance as aircraft-specific. Sarac et al. [2] proposed an operational model that develops aircraft routings based on aircraft-specific maintenance requirements. Using a connection network, the authors created a MILP model, which they then solved using a branch-and-price algorithm. The model prevents registrations from exceeding their allotted flight time and ensures that aircraft with tasks due the next day make a night stop at a maintenance station. Lagos et al. [10] plan daily aircraft itineraries and assign dynamically incoming tasks to nighttime maintenance slots. More on their work later in chapter II.

Maintenance planning

Maintenance activity is divided into scheduled and unscheduled maintenance tasks. Scheduled tasks or routine work are requirements or Maintenance Requirement Items (MRIs) dictated by the authorities. They can be divided into three categories: hard time, executed at fixed Calendar Days (CY), Flight Hours (FH), and/or Flight Cycles (FC) intervals, whichever comes first; on condition, executed whenever a certain condition is met; and condition monitoring, executed when the state of health-prognostic controlled systems passed a threshold [30]. Routine work is scheduled as free-floating or is grouped together into blocks based on task intervals.

Unscheduled tasks or corrective work are faults executed based on necessity. They are divided into three main categories: Deferred Defects (DDs), non-routines, and ad-hoc. DDs, which are further divided into Minimum Equipment List (MEL) and Non-Safety Related Equipment (NSRE), are faults that occur during operations. The difference between NSREs and MELs is the source of their due date. NSREs are set by the airline, while MELs by the manufacturer and are thus more critical. Non-routine work originates from scheduled work and cannot be deferred. Lastly, ad-hoc tasks originate from the manufacturer, which also sets their due dates.

All maintenance tasks are performed in maintenance slots, which is a period of time assigned to aircraft registrations in which tasks can be completed. There are two types of slots, hangar and platform slots. Slots differ in duration, defined as the time between the start and end of the slot, and in Full Time Equivalent (FTE, or workforce per hour). These characteristics, together with blocks of requirements, define what is known in the industry as letter checks [59]. Letter checks, such as A and C, are performed at intervals and vary per airline.

Airlines do not generate any revenue from aircraft on the ground performing maintenance. Furthermore, maintenance expenses represent on average more than 10% of the total expenditures of an airline [21]. Therefore, efficient maintenance scheduling is of crucial importance. One should try to prevent tasks from going due and causing Aircraft on Ground (AOG), but at the same time try to maximize task interval utilization to reduce ground time. This complicated problem has received wide attention in literature and as van den Bergh et al. [27] present in their literature review, it can be divided into check scheduling and task scheduling.

Check scheduling

Check scheduling, or letter check scheduling, consists in assigning maintenance slots to registrations. In network-oriented literature, check scheduling using pre-defined thresholds is often used to incorporate maintenance. In maintenance-focused literature, check scheduling is done using intervals and used to create long-term schedules, often of multiple years.

Deng et al. [54] suggested a model that schedules A and C-Checks over a four-year planning horizon. Using dynamic programming, a sequential check schedule that minimizes interval loss is selected. Testing the model on a fleet of 45 aircraft, revealed a 7% reduction in the number of checks and a more robust schedule. Two years later, Deng and Santos [12] extended their work by considering slot duration and aircraft flight hours uncertainty. The authors present a Markov decision process to address the maintenance scheduling problem. Compared to their previous work, a negligible decrease in aircraft utilization is observed, while the number of slots, and hence ground time, significantly decrease.

Literature on maintenance-focused check scheduling has shown that the approach is best suited to long-term planning. Thus, due to the short-term nature of disruption management, check scheduling is not best suited to maintenance disruption management. Therefore, one needs to look at task scheduling which is more short-term oriented.

Task scheduling

In the past, researchers that have focused on fleet assignment and routing problems only minorly considered maintenance scheduling and often under the form of check scheduling. Task scheduling received interest in recent years, with authors taking different approaches following either strategic or operational logic. The problem consists of assigning maintenance tasks and registrations to slots. Thus, compared to check scheduling, this approach offers more complete schedules but is more suited to shorter time frames, due to its rapidly growing size with the number of tasks.

One of the earliest studies to cover maintenance task scheduling in depth was published by Marseguerra and Zio [44]. The document describes a variety of maintenance and repair procedures that take into account both safety and financial considerations. Both the cost of task scheduling and the cost of maintenance staff assignment are included. The authors use a genetic algorithm to reduce the computing time, while a Monte Carlo simulation simulates the arrival of aircraft faults.

Witteman et al. [43] approached task scheduling from a strategic perspective by proposing a heuristic method for optimally assigning routine work and DDs to predefined maintenance slots, using a bin-packing problem formulation. The heuristic prioritizes tasks with the same interval, then sorts the remaining tasks in decreasing order of cost. Tasks are assigned to bins by considering tasks' due dates and labor duration. The algorithm creates the next instances for recurring tasks once they are scheduled. The model is tested on a four-year maintenance slot plan for 45 aircraft and can provide, within 15 minutes, results within 5% of the optimal.

A task scheduling model for line maintenance is put forth by Shaukat et al. [60] from a shorter 3 to 4 weeks strategic viewpoint. Using fixed aircraft rotations as input, the MILP model considers workforce resource restrictions while scheduling recurring job with the objective of reducing the expenses related to intervals waste. The authors suggest two approaches to solving the problem: an ideal branch-and-bound algorithm and a heuristic strategy that mimics the strategy employed by a partner airline. Despite the smaller size of the studied examples (13 aircraft ad 3 maintenance stations), the optimal method still requires run times of up to an hour and delivers solutions with objective function averaging 3.5% better than the heuristics. The run time can be improved by grouping tasks into blocks, which the model does not. However, a big limitation of the model is that task duration is ignored.

Papakostas et al. [47] propose an operational model for line maintenance tasks assignment during the day of operations. The model chooses the task-station allocation with the highest aircraft utility from a set of potential task-station allocations found using Monte Carlo simulation. The aircraft utility measure is composed of costs, wasted usable interval, operational risk, and simulated delay. Although the model attempts task scheduling from an operational perspective, its complexity limits it. The model can only be applied to a single aircraft at a time and a small number of tasks as the number of task-station allocations grows significantly rapidly. Furthermore, their concept is only really applicable to health-monitored parts, and it does not consider all maintenance disruptions that could occur on the day of operation.

A notable development in the integration of aircraft routing and maintenance scheduling was accomplished by Lagos et al. [10]. In addition to planning daily aircraft itineraries, their model assigns incoming tasks to nighttime maintenance slots. With the objective of reducing overall costs modeled by AOG and missed tasks costs, they represented the issue as a Markov decision process, where the cost is the result of the state and the action that is selected. Several dynamic programming solutions were then proposed. Their approach includes the usage of resources, in the form of different skill levels of maintenance personnel, and it distinguishes between critical and non-critical tasks. Using a fleet of 30 domestic aircraft, they evaluated their model, and the results showed a 13% decrease in expired tasks. However, it assumes maintenance only occurs overnight, making the model only feasible in a short-haul context.

Optimization techniques

The airline network and maintenance planning problems are large and complex. This section will present various optimization techniques used in literature to create network and maintenance schedules, focusing especially on the advantages and the disadvantages, rather than the methodology. These include branch-and-bound, column generation, Bender decomposition, dynamic programming, reinforcement learning, and the multi-label shortest path algorithm.

Branch-and-bound

A fundamental optimization algorithm, branch-and-bound is used by many commercial solvers, such as Gurobi and Xpress, often used to solve the airline network and maintenance planning problems. It works by decomposing the problem into subspaces or branches (typically done by varying the value of a decision variable) and pruning branches that will not lead to the optimal solution.

The algorithm is established in the field and is particularly useful for solving MILP problems. It is used in combination with a rolling horizon by van Kessel et al. [52] in the aircraft maintenance recovery problem and by Abdelghany et al. [29] in the aircraft schedule recovery problem.

Branch-and-price and column generation

Suited to solving large-scale linear programming problems with many decision variables, branch-and-price is often implemented in network scheduling problems. Branch-and-price combines the branch-and-bound algorithm with the column generation technique. The basic idea behind it is to first solve the simpler restricted master problem, which includes only a reduced set of decision variables, to find dual variables. Next, a subproblem called the pricing problem is solved to generate new promising variables known as columns, for example LOFs, to add to the master problem. The pricing problem is solved repeatedly until all new variables with negative reduced costs have been found, after which the complete master problem is solved.

The formulation of column generation suits problems with many decision variables, of which only a small subset will be part of the final optimal solution. It helps reduce the solution space and consequently the computational time. Furthermore, this technique has already proven its worth in airline scheduling models, as it was the approach of choice for many researchers. Column generation, through the branch-and-price algorithm, is used to generate LOFs in aircraft routing, for example, by Lan et al. [58] and Liang et al. [71], or to create recovery plans, for example by Liang et al. [70] and Eggenberg et al. [45]. However, column generation can struggle to converge to an optimal solution. Additionally, determining the initial set of decision variables with which to solve the restricted master problem is very difficult and will affect the computational time.

Bender decomposition

Bender decomposition, like column generation, is an optimization technique used to solve large-scale linear programming problems. As the name suggests, the problem is decomposed into two subproblems, the master and the subproblem. The master contains a subset of the decision variable and constraints common between it and the subproblem, while the subproblem contains the remaining variables and constraints unique to it. Solving the subproblem using the cutting plane approach provides new constraints that set the objective function's upper bound in the master problem.

As mentioned Bender decomposition is especially helpful when solving large-scale problems as it improves convergence time. However, problems need to be easily decomposed, their variables divisible into distinct subsets, and their constraints separable into groups specific to the variable subsets. For example, Froyland et al. [16], who create a robust network schedule from a set of historical disruption scenarios, use Bender decomposition to create subproblems for each disruption scenario. Or, as Hassan et al. [35] have shown, Bender decomposition is more commonly used in models that integrate aircraft and crew recovery. Therefore, Bender decomposition is best suited to problems in which the decomposition is apparent, such as in an integrated recovery problem which can be decomposed between aircraft and crew recovery.

Dynamic programming

Dynamic programming is a mathematical optimization technique suited to solving problems that are easily divisible into subproblems. The central principle of dynamic programming states that an optimal solution to a problem can be established from the optimal solutions of its subproblems. Therefore, the algorithm is based on splitting the problem into smaller, simpler subproblems that are solved once and their solutions stored. The subproblems must be overlapping for dynamic programming to be useful.

Dynamic programming is well suited to resource-constrained shortest path problems, for example, aircraft recovery with maintenance restrictions [55]. In fact, it is used by Eggenberg et al. [45] to generate aircraft recovery plans that abide to maintenance requirements. Furthermore, also used for maintenance scheduling, for example, by Deng et al. [54], dynamic programming has the advantage of finding an optimal solution more efficiently by breaking the problem down into smaller subproblems. However, a drawback of this technique is that it can become very computationally expansive when there many subproblems whose solutions need to be stored.

Reinforcement learning
Reinforcement learning, a machine learning algorithm, can be applied to solve optimization problems that are complex or have dynamic characteristics. It involves formulating the problem as an interaction between an agent and an environment (in this context the optimization problem), where the agent learns to make decisions (change decision variables). The agent learns how to make decisions through a reward/penalty system with the goal of improving the optimization objective over time.

Reinforcement learning is one of the newest techniques implemented to solve the airline network and maintenance planning problems. It was implemented by Hassan [31] to create aircraft recovery plans, while by Tseremoglou et al. [20] in the aircraft maintenance recovery problem. Its advantages in solving complex, dynamic problems, were particularly put to practice by Lagos et al. [10], which used reinforcement learning to assign LOFs and perform maintenance tasks assignment. Nonetheless, reinforcement learning is relatively new in this field, and, as will be seen later, currently struggles to find solutions of the same quality as other more established optimization techniques. This has to do with the difficulty in tuning the algorithm's exploration strategies for an effective convergence to a meaningful solution.

Multi-label shortest path algorithm
The Multi-label shortest path (MLSP) algorithm is used to solve shortest path problems optimally in a graph with connections between nodes associated with multiple costs or labels. Each node in the graph is characterized by a set of labels. At each iteration, starting from a source node, the MLSP algorithm generates a new label set for each adjacent node by combining the labels of the previous node with the cost of the connection link. Then, non-dominant paths (i.e. paths with no label better than the best) to that node are excluded. The algorithm terminates when the sink node is reached, from which the optimal path can be reconstructed by backtracking following the labels with the lowest cost.

The MLSP algorithm, although not widely implemented in aircraft routing or maintenance scheduling compared to previously mentioned optimization techniques, it is particularly useful to graph problems where connections are associated with multiple costs and/or requirements. An example of such a problem is aircraft recovery with maintenance requirements, as implemented by Liang et al. [70]. In that case, connections between two flights were dependent on propagated delay, airport slot availability, and maintenance FH, FC, and CY requirements, hence requiring the MLSP algorithm. Considering that maintenance task scheduling has yet to be integrated into aircraft routing, the MLSP algorithm is a high-ranking candidate for the task. Nonetheless, the run time of the MLSP algorithm increases rapidly within graphs with high connectivity. However, this should not be a problem in a long-haul hub-and-spoke networks.

Network types
Multiple network types used in literature to model real-world airline schedules have been proposed. Clausen et al. [22] and Hassen et al. [35] analyzed airline disruption literature up until 2009 and between 2009 and 2018 respectively and found that the most commonly used networks are the connection network, time-space network (TSN), and the time-band network. These network representations are further detailed below.

Connection networks
The connection network is an activity-on-node network. Schedule information is used to construct the nodes and assess the feasibility of the connecting arcs. A node represents a single flight with the following characteristics: arrival and departure airports and times and assigned aircraft registration. Arcs represent the connection between two flight legs, i.e. two nodes. There are conditions that should be satisfied for two nodes to be connected by an arc. One, the arrival airport of the first flight leg (f_1) is the same as the departure airport of the second flight leg (f_2). Two, the scheduled arrival time of f_1 plus the minimum turnaround time is before the scheduled departure time of f_2. Three, f_1 and f_2 can be operated by the same aircraft. More constraints can be added to restrict a maximum delay and a maximum time gap between the arrival of f_1 and the departure of f_2. Figure 3 gives a visual representation of a connection network.

Connection networks work with paths, which are a series of connected nodes representing a feasible aircraft route (e.g., Liang et al. [70] in reactive and Froyland et al. [16] in proactive disruption management). Authors often use heuristic methods to generate a restricted set of preferred routes (e.g., Zhang [73]). Maintenance opportunities in the connection network can be represented by nodes with start and end times, but of course with the same arrival and destination airport. Alternatively, two flight nodes can be connected by a maintenance arc, in which maintenance activities can be carried out (e.g., Ben Ahmed et al. [38]).

The connection network has multiple advantages. Firstly, their path-based nature means that they have fewer decision variables which result in faster computational times. Secondly, time can be continuous, as implemented by Liang et al. [70] which resulted in recovery costs being ~ 40% lower compared to discretized time with 30 min intervals. Thirdly, maintenance opportunities are easily integrated into a connection network. However, the connection network also has drawbacks. Mainly that time is not represented visually in the network.

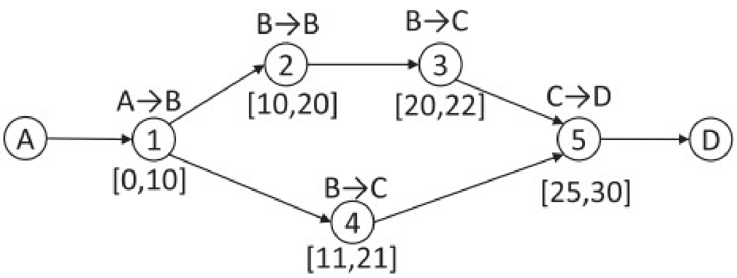

Figure 3: An example of a connection network [70]. A and D are the source and sink nodes respectively, the letters above the nodes are the departure and arrival airports of the flight, while the numbers below the nodes are the departure and arrival times. Note that node 2 is a maintenance opportunity.

Time space networks

The time-space network (TSN) is a two-dimensional network. On one axis time, discretized in time steps, on the other space i.e. airports. The most basic TSN is composed of nodes and three types of arcs: flight, ground, and overnight arcs. A node is characterized by a time and an airport. Therefore the maximum number of nodes in the TSN is the number of time steps in the recovery period times the number of destinations. Flight arcs, as the name suggests, represent a flight, and thus connect two nodes with different airports and times. The times and airports correspond to the flight departure and arrival times and airports. Ground arcs connect two nodes at the same airport, and thus represent the time between flights (e.g., turnaround time, maintenance time, or slack). Lastly, overnight arcs connect sink nodes back to source nodes and hence ensure the continuity of the schedule. Figure 4 gives a visual representation of a TSN.

The TSN has been widely implemented for both proactive (e.g., Vink et al. [28] and Yan and Yang [69]) and reactive disruption management (e.g., Lan et al. [58] and Liang et al. [71]). Flight delays and/or flight retiming in the TSN are often implemented using time-shifted copies for each original flight arc in the model and a new set of constraints preventing more than one copy to be selected. Maintenance in the TSN can be implemented using ground arcs, while multi-fleet representation requires parallel TSNs to model each aircraft individually, as shown by Vink et al. [28].

The TSN is widely implemented and offers a great visual representation of an airline schedule, but it has drawbacks. Firstly, the quality of the solution and computational intensity is dependent on the time step. Shorter time steps, result in a higher quality solution, but also in more decision variables, and hence longer run times. Moreover, compared to the time-band network, explained later, ground arcs add a lot of variables. Secondly, although it is possible to represent multiple fleets and perform aircraft routing, parallel networks are required, which complicates the problem.

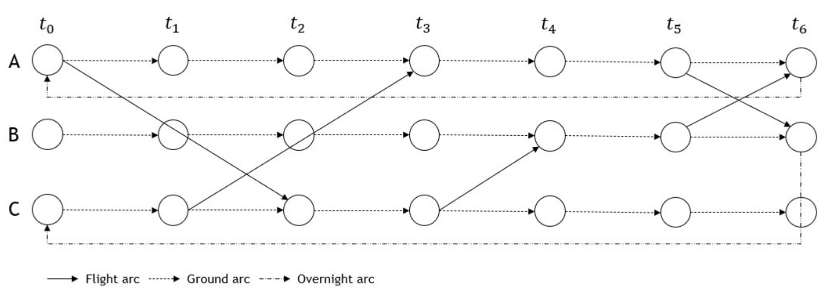

Figure 4: An example of a TSN. The y-axis represents space (airports A, B, and C), while the numbers x-axis time.

Time-band networks

The time-band network is similar to the TSN in the sense that it is a two-dimensional network. On one axis time, discretized in time steps, on the other space. However, ground activity is aggregated on one node during each of the sequential time intervals. Therefore, this removes the need for ground arcs. The time-band network is composed of two arc types: flight and termination arcs. Flight arcs are the same as in the TSN. While termination arcs connect each node to the sink node at the same airport. They ensure the termination of the aircraft route to the sink node destination after the last flight. Figure 5 gives a visual representation of a TSN.

The time-band network has mostly been implemented in reactive disruption management (e.g., Bard et al. [24] and Eggenberg et al. [45]). Like the TSN, flight copies are used to model flight delays. An advantage of the time-band network is the possibility to use varied time bands between hub and spoke airports. Bard et al. [24] implemented a finer time discretization at hub airports, where greater accuracy is required, while a less computationally intensive discretization at spokes. Maintenance opportunities can be integrated within nodes, as one of the ground activities, with the addition of maintenance termination arcs, which are essentially termination arcs with an integrated maintenance opportunity.

The time-band network has advantages over its close relative, the TSN. Firstly, the aggregation of nodes, which results in no ground arcs, reduces the problem size. Secondly, in the time-band network, it is possible to use different time discretization per node type, allowing for a better trade-off between solution accuracy and computational cost. Nonetheless, this network type also has disadvantages. It shares two of the same disadvantages as the TSN. One, that the solution quality depends on the time step, and two that multiple sub-networks are needed to represent individual aircraft. Furthermore, compared to the TSN, it offers a worse visual representation of ground activity, like maintenance, as they are built into the nodes.

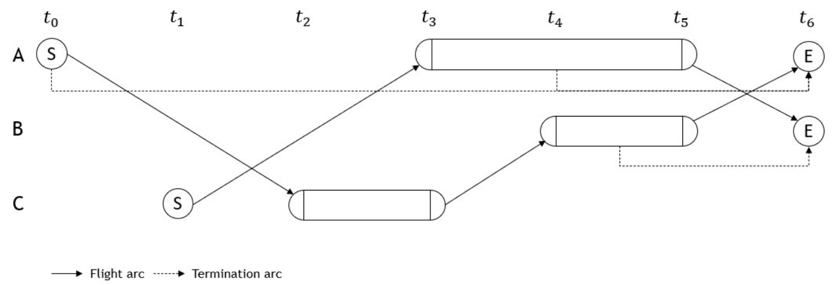

Figure 5: An example of a time-band network. The y-axis represents space (airports A, B, and C), the x-axis represents time, S are source nodes, and E are sink nodes.

Concluding remarks on literature

Network planning involves solving various subproblems, including the FAP and the MRP. Both problems have been extensively researched in literature, but maintenance is typically overlooked, treated as a fixed interval activity, and only [10] considering it as an aircraft-specific event. Similarly, maintenance scheduling disregards network is mostly solved in two approaches check scheduling or task scheduling. Check scheduling is more applicable for long-term planning and involves assigning aircraft registrations to specific letter-checks at fixed intervals. On the other hand, task scheduling is used for shorter-term planning and involves assigning maintenance tasks and aircraft to specific time slots.

Various optimization techniques were explored, each with their advantages and disadvantages. and Column generation and MLSP algorithm seem to be best suited to the problem at hand. Column generation has been widely implemented in airline scheduling problems, all be it not in maintenance, and is known to reduce computational time significantly. MLSP algorithm, due to its formulation, is suited to problems that integrate aircraft routing with maintenance requirements. These can also be combined as done by Liang et al. [70].

There are three types of networks are used to construct flights and slots schedules, namely the connection networks, the time-space networks, and the time-band networks. All three network types are suitable for disruption management and offer distinct methods for modeling maintenance, delays, and other pertinent factors, but connection and time-space networks are more widely implemented. The time-space network offers superior visualization capabilities, while the connection network permits continuous time modeling.

Proactive disruption management

Proactive approaches to handling disruptions are addressed in this chapter and are divided into two groups: proactive absorption (discussed in chapter II) and proactive recovery (discussed in Table II). A summary of proactive maintenance tactics is also provided in Table II. In Table II, a conclusion of the chapter is provided.

Proactive absorption

The goal of proactive absorption is to design schedules that are insensitive to small disruptions [22]. Therefore, proactive absorption ensures schedule stability [1]. In literature, stability was achieved in three ways: efficient allocation of buffer through flight departure retiming, optimal delay routing, and scenario-based recoverable robust routing. Ensuring schedule stability often introduces a trade-off between robustness and operating cost. For example, larger time buffers guarantee higher robustness, however, they result in inefficient use of resources, and thus higher operating costs (or lower operating profit) due to higher ground time.

Buffers and flight retiming

Many authors have addressed the problem of schedule stability by reallocating slack time in the schedule, often by retiming flights' departure times. When trying to implement schedule stability through buffers they all faced two issues: how much buffer time is needed, and where exactly should it be allocated in the schedule [22]. Wu [65] finds a solution to this problem by allocating buffer times using sequential optimization algorithms. Through airline historical data he constructed stochastic functions for a flight's en-route time and turnaround time, the latter modeled as three Semi-Markov Chains (passenger boarding/deboarding, cargo & baggage loading/unloading, and other independent activities (e.g., fueling, catering)). Wu [65] then performed 1000 Monte Carlo simulations to determine every flight's expected delay. He then established a schedule adjustment for each fight by comparing the expected delay to an input target delay. Based on the calculated schedule adjustment the turnaround time of flights is either increased or decreased. His method resulted in a 30% reduction in departure delay and an increase in schedule reliability from 37 to 52% but with the assumption of constant airport slot availability.

Wu [65] presented a simplified stochastic approach to buffer assignment with the sole objective of reducing delay with respect to a target, which can be very arbitrary. To improve on this, AhmadBeygy et al. [56] constructed a flight retiming model that assigns slack with the objective of minimizing flight delay propagation. They first present a single-layer model, in which delay propagation is considered only to the next subsequent flight. As delay often propagates past the next flight, they also developed a multi-layer model, which makes use of a delay propagation tree to consider delay propagation across the entire routing. In both models, slack between two subsequent flights is a decision variable, restricted between bounds to ensure the feasibility of aircraft routing and crew pairing. Propagated delay is another decision variable, and historical data is used to determine the probability of occurrence for each propagated delay. Although AhmadBeygy et al. [56] achieved between 5 to 50% propagated delay reduction compared to the original schedule, the models can only consider a worst-case scenario (due to the delay propagation tree construction) and do not allow changes to the aircraft routing.

Aloulou et al. [37] address the latter problem of AhmadBeygy et al. [56] research by integrating aircraft fleet assignment and flight retiming. Their model creates routes and determines flight departure times with the objective of maximizing a surrogate robustness measure, which implicitly captures the effects of disruptions. This measure of robustness is slack-based and has two parts: slack to ensure aircraft connection and passengers connection. Furthermore, unlike AhmadBeygy et al. [56] that determined a flight's new departure times using slack, Aloulou et al. [37] select a flight's new departure time from a set of possible times and thus treat it as a decision variable. This required a new constraint that ensures that exactly one departure time is assigned per flight.

Ben Ahmed et al. [38] proposed a two-stage model for multi-fleet aircraft routing and flight retiming. In the first phase, they solve a MILP that finds optimal aircraft routes (with established fleet assignment, thus routing is performed per aircraft type) with the aim of maximizing a simpler version of AhmadBeygy et al.'s [56] surrogate robustness measure. They implement a connection network with maintenance arcs (i.e. arcs between two connecting flight nodes in which maintenance activities can be performed). In the second phase, they implement a heuristic approach based on Monte Carlo simulations for flight retiming with the aim of maximizing On-Time Performance (OTP). After testing the model on a middle eastern airways fleet, they obtain between a 10 to 16% increase in OTP. There are limitations to their work. Firstly, as aircraft routing is often performed one week prior to operations, and well after tickets are sold, it might not be feasible to retime flight departure times. Furthermore, although their unique effort includes maintenance, the implementation is limited. Firstly, it is too simplified (every aircraft has a mandatory maintenance time to be achieved before accumulating a specific number of flight days), and secondly, it excludes the possibility of using maintenance for schedule stability. Especially the latter is an area yet to be explored.

Flight retiming and buffers are common methods implemented in literature and in practice to increase schedule stability, however, they introduce inefficiency in the schedule. A recurrent issue in literature is that slack is always maximized, or at least never minimized. For example, take Aloulou et al. [37] and Ben Ahmed et al. [38] research, in which routes are created for maximum robustness which is a proxy for maximum slack. In fact, buffer time reduces aircraft productivity, and according to Ball et al. [40] the cost of buffer time due to lost revenue was $ 3.7 billion in 2007. **Therefore, an interesting unexplored research direction is creating routes with the aim of minimizing slack and propagated delay simultaneously.**

Delay robust routing

A flight is associated with two types of delays: primary delay and propagated delay. Primary delays are not a function of routing and include mechanical issues, bad weather, safety issues, connecting passenger delays, etc. On the other hand, propagated delay is a function of routing, and it is described as the delay carried forward from previous flights which could not be absorbed in the buffer time between flights. Thus, propagated delay is dependent on the propagated and primary delays of previous flights in the same route. The process of delay robust routing involves constructing routes that minimize propagated delay to create schedule stability.

Lan et al. [58] implemented a single fleet aircraft routing model in sequence with an aircraft retiming model. Aircraft routing is performed with a LOF-based, stochastic model that minimizes the expected propagated delay (EPD). Using historical airline data they model flight arrival delay with a log-normal distribution, and together with aircraft historical routing, they separate primary and propagated delay. To reduce the problem size, column generation is used to generate routes with already low propagated delays. After aircraft routing, the selected optimal LOFs serve as input in the MILP retiming model, which aims to minimize the number of disrupted passengers. Lan et al. [58] use flight copies for flight retiming with a constraint that prevents selecting more than one copy. The end result is a one-day schedule that results in a 1.6% improvement in OTP and a 40% reduction in passenger misconnections.

Although outside the scope of Lan et al. [58] research, due to their short one-day scheduling window, maintenance can have a profound impact on aircraft routing. Liang et al. [71] also created a robust aircraft routing model that minimizes the EPD, however, their model schedules over one week period and incorporates maintenance, planned overnight every three days. Furthermore, they introduce a more accurate, but more computationally expensive, function to determine the EPD of a LOF. Like Lan et al. [58], column generation is used to generate LOFs. While generating LOFs, they also allocate buffer time efficiently using a stochastic method. Overall, they show that using the conventional formula for EPD (used by Lan et al. [58]) can underestimate a LOF's EPD by up to 40%. However, their model is again suited to only one aircraft type and, like other proactive models so far, maintenance is overshadowed.

Yan and Kung [68] deviated from the common approach of minimizing EPD. Instead, their robust aircraft routing problem minimizes the maximum propagated delay by assuming that flight delays lie in an uncertainty set. This approach allowed the authors to model the interconnection of primary delays, however, at the expense of overestimated propagated delays. They compared their results against those of Dunbar et al. [41] (EPD minimization delay robust routing) on two networks and on both historical and simulated delay data. They found that their approach resulted almost always in less volatile propagated delays and lower worst-case propagated delay, and ~ 50% of the time in a lower average propagated delay. However, the article fails to compare the total slack time in the LOFs of the two approaches. Therefore, they fail to present the cost of the extra ground time required when considering the maximum rather than the expected propagated delay.

Delay robust routing models make use of historical delay data to create delay probability distributions and often require a heuristic method, such as column generation, to simplify the problem, which otherwise is too large. However, all approaches presented so far are applicable only to a single fleet, thus requiring fleet assignment to be performed beforehand. No model has integrated fleet assignment and aircraft routing in the propagated delay minimization problem. Furthermore, maintenance is often not included in the problem, and when it is, like in Liang et al. [71], its implementation is oversimplified. Additions in the field could be integrating maintenance task scheduling and/or adding maintenance slot duration uncertainty as an additional delay source. **Thus, a multi-fleet model that minimizes propagated delay resulting from both flights and maintenance activities would be a valuable addition to literature.**

Scenario-based recoverable robust routing

Schedule stability methods thus far do not consider recovery directly, but indirectly by reallocating buffers and routing to minimize delays. Scenario-based recoverable robust routing is an approach by which aircraft are routed based on minimizing recovery costs from a set of historical disruption scenarios.

In 2014, Froyland et al. [16] introduced this novel technique to aircraft routing. Opposed to conventional robust optimization, their recoverable robust model aims to minimize planning costs and recovery costs. They consider 102 different disruption scenarios of two types, airport closures and aircraft groundings, and implement three recovery actions: flight delay, flight cancellation, and aircraft swap. The model's objective function selects the routes that minimize the total recovery costs of all 102 disruption scenarios. An issue of recoverable robust routing is a vast amount of decision variables, which the authors reduce by using firstly, Benders' decomposition to separate the disruption scenarios and secondly, column generation to create LOFs for each scenario. The model's performance is assessed by comparing the recovery costs of the LOFs selected based on planning and recovery costs to the recovery costs of the LOFs selected based on only planning costs. The results show between ~ 0 and ~ 15% reduction in recovery costs when recovery costs are considered in the planning. However, the authors fail to compare recovery costs with routes created using other proactive absorption methods (e.g., minimization of EPD or flight retiming) to truly see the added value of this method compared to others.

Being the first of its kind, Froyland et al.'s [16] model can be enhanced. Firstly, more recovery actions can be considered, such as reserve aircraft. Secondly, maintenance is not considered in the LOFs and because of this maintenance recovery actions, such as maintenance slot swapping, are also ignored. Thirdly, the model is a single fleet model, and thus cannot consider swapping aircraft of two different types, which is often performed and is more expensive. Lastly, a problem specific to recoverable robust routing is the fact that routes are optimized based on a set of scenarios, and can thus be sub-optimal in the event of new disruptions. One can increase the number of scenarios to cover more disruption events but at the expense of computational time. Nonetheless, this approach has potential, as it implements stability by looking directly at reducing recovery costs, which is the goal of stability in the first place. **It can thus be further explored with the integration of maintenance and more recovery actions.**

Table 1 presents an overview of proactive absorption literature. In general, extensive research has been done on flight retiming and minimizing propagated delay, while recoverable robust routing remains a new topic. Many authors considered aircraft routing, but neglect the fleet assignment problem. This makes sense for delay and recoverable robust routing, but less so for flight retiming, as when aircraft are routed tickets have often been sold already. Furthermore, maintenance is commonly excluded. It could be used to introduce robustness opportunities (e.g., maintenance swaps in recoverable robust routing) and sources of disruptions (e.g., delays in delay robust routing). Focus on any of these aforementioned fields would contribute to proactive absorption literature.

	Proactive absorption			**Model characteristics**			
Paper	**B&FR**	**DRR**	**RRR**	**FAP**	**MRP**	**NT**	**Objective**
AhmadBeygy et al. [56]	✓						Minimize delay propagation
Aloulou et al. [37]	✓			✓			Maximize robustness
Ben Ahmed et al. [38]	✓				✓	Connection	Maximize robustness & OTP
Dunbar et al. [41]	✓	✓			✓		Minimize delay cost
Froyland et al. [16]			✓		✓	Connection	Minimize planning & recovery costs
Lan et al. [58]	✓	✓			✓	Connection	Minimize EPD & disrupted pax
Liang et al. [71]	✓	✓			✓	TSN	Minimize EPD
Wu [65]	✓						Minimize estimated delay
Yan and Kung [68]		✓			✓	Connection	Minimize max propagated delay

Table 1: Overview of proactive absorption literature. Abbreviations in the table: B&FR: Buffer & flight retiming; DRR: Delay robust routing; RRR: Recoverable robust routing; FAP: Fleet Assignment Problem; MRP: Mainteance Routing Problem; NT: Network type.

Proactive recovery

The goal of proactive recovery is to design schedules that are easily adjusted if disruptions occur [22]. Therefore, proactive recovery ensures schedule flexibility [1]. Multiple strategies were proposed in literature to achieve schedule flexibility. Increasing swapping opportunities is the most common. Other methods include scheduling in sub-networks, decreasing hub connectivity, enforcing station purity (i.e. one aircraft type per airport), and canceling flights on the same aircraft route. Like proactive absorption, proactive recovery yields plans that are more expansive when only considering planning costs, but become cheaper when one also considers recovery costs. However, compared to stability, flexibility does not add costly, inefficient buffer times to the schedule. Nonetheless, an airline opting for flexibility over stability needs to be aware that their recovery process might be more intense and delivering the customer promise more difficult.

Swapping opportunities

Aircraft swapping is common practice at airlines OCC for schedule recovery. It often enables a cheaper solution to a disruption compared to flight delay and cancellation. Therefore, creating schedules that increase swapping opportunities is encouraged. This is achieved by maximizing the overlap in scheduled ground time between routes.

Ageeeva [67], in her thesis, first introduced the concept of using swapping opportunities for schedule flexibility. She created a two-stage aircraft routing model with the objective of maximizing swapping opportunities. In the first phase, she solves a conventional LOF-based aircraft routing model disregarding swapping opportunities. In the second phase, she finds alternative optimal solutions and assesses their flexibility, which is a function of the time two aircraft have overlapping time at an airport. The best solutions (i.e. the ones with the highest flexibility) are selected for the final solution. Although her solution yields a 35% higher opportunity index (i.e. ratio between actual and potential intersecting routing), she does not simulate disruptions in a recovery phase, and thus it cannot be concluded if her routes are more robust.

The method is re-proposed by Burke et al. [13], who this time do compare their new schedule to KLM's original 2006 summer and winter schedules and find up to ~ 7% improvement in OTP. They propose a multi-objective aircraft routing model that integrates multiple robustness strategies: swapping opportunities, flight retiming, and reserve aircraft. A TSN, with flight copies for flight retiming (same method as Lan et al. [58]) and separate reserve aircraft arcs, is used for routing. The authors also integrate maintenance, which is scheduled every 60 flight hours. The model has two objectives: reliability and flexibility. Reliability is represented by the sum of the probabilities that each flight in the schedule departs late, while flexibility is given by the sum of the probabilities that each flight involved in a swap departs on time. The probabilities are determined using historical distributions for turnaround and en-route time. The model is solved using a multi-meme memetic algorithm, that combines a genetic algorithm with three Pareto frontiers.

A limitation of both pieces of research is that they disregard fleet assignment. Consequently, they do not facilitate swapping opportunities between aircraft of different types, which are performed in practice and are significantly more expensive than same aircraft type swaps (due to different seating capacities). Furthermore, both do not consider the swapping of maintenance slots and hence do not optimize for this. **Thus, a valuable addition to literature would be a model that maximizes opportunities for flight and maintenance slot swaps across all aircraft types in the fleet through flight retiming**

Other proactive recovery methods

Aside from aircraft swapping opportunities, other methods to enhance schedule flexibility exist. One of these, partitioning an airline schedule into sub-networks, was explored by Kang [32]. The motivation behind this approach is to shield higher revenue passenger itineraries from disruptions by prioritizing the recovery of the sub-networks containing these itineraries. Furthermore, aircraft swapping is only allowed within sub-networks, thus reducing the size of the recovery problem. Kang [32] considers dividing flights into sub-networks at different stages of the planning procedure (schedule design, fleet assignment, and aircraft routing). The objective of the models is to maximize revenue by assigning high revenue-generating itineraries to higher-priority sub-networks, with a constraint that limits the number of flights per sub-network. Using an event-based simulation, MEANS, they simulate weather disruptions by reducing hub airports' capacity. Contrary to the theory, during bad weather, their schedule resulted in more cancellations, more passenger missed connections, longer passenger and flight average delays, and more variation in flight delays (i.e. larger standard deviation). Deteriorating performance is driven by lower-priority sub-networks, which experience many large delays. Therefore, the performance gain in priority sub-networks does not outweigh the performance loss in the lower priority sub-networks. Additionally, this approach is not suited to a network with a single hub with hub-to-spoke flights, but only to airlines with multiple hubs.

Smith and Johnson [61] proposed an alternative to schedule flexibility that is also suited to single hub networks: station purity. The objective of station purity is to reduce lonely fleets (i.e. situations in which a fleet serves an airport only once or twice during the scheduling horizon), which should in turn facilitate the swapping of aircraft of the same type. They proposed a standard fleet assignment problem that maximizes operating profit with an additional station purity constraint that limits the number of fleet types that can serve each airport in the network. Airports, depending on their level of daily activity, receive different purity restrictions: 1 or 2 fleet types, or no purity restrictions. Due to the purity constraints, computational time increased by 500%, forcing the authors to develop an alternative. The station decomposition model assumes that there exist flights only between hubs or from hubs to spokes, and it does so by either converting spoke airports to hubs or by grouping spoke airports. The results are a greatly improved computational time, but also lower profitability (11% lower) compared with the original schedule. However, this is to be expected, as the airline cannot adapt to changing demand levels during the day and week, for example by operating a larger aircraft during demand peaks. Nonetheless, the number of fleet/airport combinations and the number of lonely fleets, greatly improve, suggesting that recovery costs of the purity-optimized schedule would be lower, but the authors do not show this. In fact, the model does not include elements of recovery, both as a simulation to test the schedule's recovery performance against the original's and in the objective by minimizing both planning and recovery costs. One could also integrate flight retiming together with station purity to further increase swapping opportunities at spokes.

Another solution to schedule flexibility is short cycles, or short LOFs, together with hub isolation. A short cycle is a LOF with few flights, while the process of hub isolation involves creating LOFs that include only 1 hub airport. This approach is limited from the start as it is not suited to all airlines, for example, airlines that only fly back and forth from one single hub. Nonetheless, for specific airline networks short cycles would result in fewer cancellations, as operators when canceling a flight often have to cancel the whole cycle, while reducing hub connectivity (same as promoting hub isolation), defined as the number of flights in a LOF that start and end at different hubs, should avoid disruption from spreading across hubs.

Rosenberger et al. [26] decided to explore the short cycles and hub isolation concept by building two fleet assignment models with a TSN, one that minimizes costs and constraints hub connectivity, and the other that minimizes hub connectivity and constraints costs. However, they first prove analytically that reducing hub connectivity results in shorter cycles. The model is then tested using SimAir to run a 500 days airline operations simulation on three schedules and different fleet scenarios. When compared to the original schedule produced with the fleet assignment model of Barnhart et al. [8], the reduced hub connectivity schedule resulted in fewer cancellations and swaps, and in an improved OTP. Nonetheless, this concept cannot be implemented in networks with only one hub.

Overall, the presented solutions to create schedule flexibility are not applicable to all airline network types, or at least best suited to one type. Furthermore, only station purity and short cycles proved to be effective in creating flexibility, but the former was not tested in a disruptive environment. In general, these methods for schedule flexibility are not as effective at creating robustness as maximizing swapping opportunities.

Table 2 presents an overview of proactive recovery literature. When compared to proactive absorption, there have been significantly fewer attempts at proactive recovery. Methods to create schedule flexibility are fewer, less researched, not applicable to all airline business models, and as a result less effective at creating robustness compared to methods for schedule stability. However, this should not be the case as promoting flexibility could in theory be the better option, as fewer resources are wasted compared to stability approaches. **Therefore, literature should focus on developing existing flexibility approaches, by testing them in more contexts, and deriving new methods for flexibility, by combining current ones or developing whole new ones.**

Proactive maintenance scheduling

Thus far proactive disruption management has been presented from the perspective of network. This section will present literature that explores maintenance schedule stability and flexibility. Compared to network robustness, this field has received significantly less attention. Examples of stability and flexibility in a maintenance schedule are slot swapping, the use of historical data to model scheduling uncertainty, and flexible aircraft slots.

Paper	Proactive recovery SO	SN	SP	SC	Model characteristics FAP	MRP	NT	Objective
Ageeeva [67]	✓					✓	Connection	Maximize swapping opportunities
Burke et al. [13]	✓					✓	TSN	Maximize reliability & flexibility
Kang [32]		✓			✓	✓	TSN	Maximize pax itinerary revenue
Rosenberger et al. [26]				✓	✓		TSN	Minimize hub connectivity or costs
Smith and Johnson [61]			✓		✓		TSN	Maximize operating profit

Table 2: Overview proactive recovery literature. Abbreviations in the table: SO: Swapping opportunities; SN: Sub-networks; SP: Station purity; SC: Short cycles; FAP: Fleet assignment problem; MRP: Maintenance routing problem; NT: Network type.

Historical data on checks duration and aircraft utilization, together with scenario scheduling, is an approach to increase maintenance schedule stability. van der Weide et al. [63] constructed a model for robust heavy aircraft maintenance check scheduling under uncertainty using scenarios built with historical data on check duration and aircraft utilization. They schedule only long-term heavy maintenance performed in checks with 18 to 36 months intervals, therefore well outside the time horizon of fleet assignment and aircraft routing. Nonetheless, their reasoning can be applied to smaller, more frequent checks as well. The objective was to maximize interval usage, such to maximize flight time and penalize violations and the need for extra slots. Instead of using heuristics, the authors use a genetic algorithm to simplify the solution space. Running 10 uncertainty scenarios the algorithm produced interesting results, including a 7% reduction in heavy maintenance and a corresponding 4.4% increase in aircraft availability. However, it is not possible to determine how the added fleet availability translates into extra flights as network is not considered in the mdoel.

Shahmoradi-Moghadam et al. [18] adopted a similar approach to maintenance task scheduling for a military aircraft fleet. They work with a rolling horizon and consider both scheduled tasks and unexpected repair jobs, which present themselves with a fault detection probability. The model aims to optimize fleet availability and is constrained by the availability of skilled labor. Monte Carlo simulations are used to take uncertainty samples from task duration sets. Like van der Weide et al.'s [63] work, Shahmoradi-Moghadam et al. [18] do not consider two other types of uncertainty which are causes of disruption in maintenance: aircraft late arrival and changes in labor availability.

The two works presented solely focused on the maintenance perspective, disregarding network. Lapp and Cohn [34] introduce an approach to increase maintenance flexibility by modifying LOFs and as such present a novel work that integrates maintenance with network. Their research objective was to create flexible LOFs by swapping aircraft from regular LOFs to maintenance LOFs, as to facilitate recovery from maintenance disruptions and minimize deviation from the original maintenance plan. Given an input set of LOFs from a major American airline, they first find all swapping opportunities based on 5 requirements, of which one is that aircraft must be of the same type. Using the swapping opportunities as input, they present an optimization algorithm that selects the best swapping opportunities to implement to minimize the total number of expected maintenance miss-alignments. A maintenance miss-assignment is defined as a LOF that does not terminate at a maintenance station but is assigned to an aircraft that requires maintenance at the end of the LOF. The number of expected miss-alignment is estimated using a binomial distribution, for every airport, and assuming the probability of an aircraft requiring maintenance at the end of the day as 1/7 (each aircraft must undergo a maintenance check every 7 days). They compared their model's revised LOFs with the airline's original and found that the number of expected miss-alignments reduced by 88%.

Although Lapp's and Cohn's [34] model successfully increases flexibility by integrating network and maintenance, it has areas of improvement. Firstly, using the original LOFs as a start solution might not lead to the optimal solution. Rather one could implement their methodology directly to fleet assignment and/or aircraft routing. Secondly, they only consider check scheduling and not task scheduling. **Nonetheless, they presented a novel technique for maintenance flexibility that can be extended to include fleet assignment and/or aircraft routing.**

Concluding remarks on literature

In literature, proactive disruption management has been addressed with two approaches. The first proactive absorption aimed at schedule stability, and the second proactive recovery aimed at schedule flexibility. Table 3 gives an overview of proactive literature, dividing research between absorption, recovery, and different levels of maintenance considerations. Network-focused models often overlook maintenance, and when they do consider it, they only treat check scheduling. To address this, task scheduling should be integrated with proactive network scheduling to create robust network and maintenance schedules simultaneously and understand the impact of maintenance scheduling decisions on flight schedule robustness.

Building robust maintenance schedules has received significantly less attention, as research has primarily focused on network. One promising approach would be to enhance the approach proposed by Froyland et al. [16] to consider, on top of the standard network recovery actions, maintenance recovery actions (e.g., slot swapping and cancellation) as well as maintenance disruption scenarios. Another option, again following Froyland et al. [16] method, would be to develop a recoverable robust maintenance scheduler that assigns tasks and slots based on a set of input maintenance disruption scenarios. Alternatively, methods that add flexibility to a maintenance schedule, for example through slot retiming and/or planning using a minimum number of days without any due tasks after a check, could be explored.

To ensure robust schedules, researchers often focused on stability, for example by minimizing propagated delays or disrupted passengers when rerouting and reallocating buffer times. Fewer researchers have approached schedule robustness through the lens of flexibility. In the case of network this was realistically only successfully done by creating LOFs that facilitate aircraft swapping, while in the case of maintenance, flexibility is still to be explored.

	Proactive absorption				**Proactive recovery**				**Maintenance**		
Paper	**B&FRT**	**DRR**	**RRR**	**MSU**	**SO**	**SN**	**SP**	**SC**	**CS**	**TS**	**S&F**
Ageeva [67]					✓						
AhmadBeygy et al. [56]	✓										
Aloulou et al. [37]	✓										
Ben Ahmed et al. [38]	✓								✓		
Burke et al. [13]	✓	✓			✓				✓		
Dunbar et al. [41]	✓	✓									
Froyland et al. [16]			✓								
Kang [32]						✓					
Lan et al. [58]	✓	✓									
Lapp and Cohn [34]					✓				✓		✓
Liang et al. [71]	✓	✓							✓		
Rosenberger et al. [26]								✓			
Shahmoradi-Moghadam et al. [18]				✓						✓	✓
Smith and Johnson [61]							✓				
van der Weide et al. [63]				✓					✓		✓
Wu [65]	✓										
Yan and Kung [68]		✓									

Table 3: Overview of proactive disruption management literature. Abbreviations in the table: B&FR: Buffer & flight retiming; DRR: Delay robust routing; RRR: Recoverable robust routing; MSU: Maintenance scheduling with uncertainty; SO: Swapping opportunities; SN: Subnetworks; SP: Station purity; SC: Short cycles; CS: Checks scheduling; TS: Tasks scheduling; S&F: Maintenance stability and/or flexibility.

Reactive disruption management

In literature, reactive disruption management is addressed through the Aircraft Recovery Problem (ARP). The ARP is introduced in chapter II followed by chapter II, in which recovery strategies and their implementation into the ARP are addressed. In Table II three types of solution approach to the ARP are presented: exact, heuristics, and other methods. Maintenance recovery and the integration of maintenance into the ARP are discussed in Table II. Lastly, the chapter is concluded in Table II.

The ARP

When one talks about reactive disruption management from the perspective of the aircraft, one is referring to the Aircraft Recovery Problem (ARP). The ARP, given a flight schedule and a set of disruptions, determines which flights to delay or cancel, which aircraft to swap, etc. to minimize the disruption costs. The disruption costs are composed of both operating costs and passenger compensation costs. The ARP is most of the time formulated as a cost minimization problem rather than a profit maximization problem since by the time it is solved tickets have been sold and revenue is thus fixed [35].

A defining characteristic of the ARP is that it needs to be solved quickly, in a couple of minutes, to provide aircraft controllers with rapid schedule recovery solutions. Therefore, ARP models often make use of heuristics and/or other methods to simplify the problem and make the tool suited for operations. Furthermore, there exist two types of distinct ARP models: static and dynamic. A static approach assumes that all disruptions are known before the simulation starts. This is the most common type of ARP. On the other hand, dynamic models find a new solution every time a new disruption arrives by building upon previous solutions. Vos et al. [19] introduced the dynamic ARP, which was later improved by Vink et al. [28].

Most studies consider two types of disruptions in the ARP: aircraft unavailability and airport capacity reductions. Aircraft unavailability can last for many hours or even days, due to unexpected failures resulting in AOG, or it can last minutes due to delay propagating from previous flights. Airport disruptions are the consequence of ground staff shortages and/or adverse weather events. In literature, they are modeled in two ways: most often through airport closure for a period of time, for example by Wu et al. [72], or as reduced flow as a percentage of the total capacity during normal operations, for example by Liang et al. [70]. Two more, less common disruption types are found in ARP literature: flight delays, for example by Aktürk et al. [42] and Bard et al. [24], and flight cancellations, for example by Vos et al. [19]. In most models, disruptions are grouped into scenarios, as by Vink et al. [28], and added to the model as constraints.

ARP models are formulated to recover from disruptions within a recovery period for which a network is built using for example a connection network, a TSN, or a time-band network. A recovery period starts when a disruption arrives and ends at a proposed recovery time. The start of the recovery period is characterized by source nodes, while the end sink nodes. The idea is that schedules within the recovery period are modified to cope with the disruption, while outside they are left untouched.

Recovery strategies in the ARP

Authors of ARP models implement the same recovery strategies used by aircraft controllers at OCCs. The most common strategies in literature are also the ones most widely used by airlines: delaying and/or canceling flights, and aircraft swapping. Other recovery actions include maintenance slot swaps, ferry and reserve aircraft, cruise speed control, and turnaround operations. The three subsections below give details on how these recovery strategies are modeled in literature.

Modeling flight delays and cancellations

While flight cancellations are treated by most as decision variables penalized by high costs in the objective function (e.g., Vink et al. [28] and Liang et al. [70]), Su et al.'s [66] ARP literature review shows that five distinct approaches are used to model aircraft delays: flight copies, constructed in paths, adaptive delays, decision variables and decided by ready time. Using flight copies, first applied by Yan and Yang [69], is common. The method consists in selecting one flight copy from a set of discrete copies, with a predetermined size and copies interval. This approach requires additional constraints that prohibit selecting more than one copy. There are two advantages of this delay modeling option. Firstly for its ease of implementation, it offers great delay variation. Secondly, it has been widely used in TSNs and time-band networks, for example, Vos et al. [19] and Vink et al. [28] in the TSN, and Eggenberg et al. [45] in the time-band network. However, the disadvantage is that computational time increases significantly when one adds copies due to extra decision variables. Therefore, there is a strict trade-off between the number of delay options (i.e. number of flight copies) and the problem size. Nonetheless, Lan et al. [58], who implemented flight copies for flight retiming (very similar to flight delays except that flights can also depart earlier than scheduled), showed that using a narrow interval between copies often fails to generate substantially better solutions.

Flight copies are implemented in time-space and time-band networks, however, construction in paths is a more common implementation of delays in connection networks (e.g., Zhang [73] and Wu et al. [74]). Paths are constructed from flight nodes and connection arcs, using for example a depth-first search algorithm, following two requirements. One of these requirements is that the time difference between two connected flight nodes is within a fixed time frame $[T_{min}, T_{max}]$, with T_{min} dictating the maximum possible delay. This approach is easy to implement and one can have many potential delay/path options depending on T_{min}. Furthermore, the delay does not have to be discrete. However, similarly to flight copies, the more delay options one allows for (i.e. a more negative T_{min}) the more paths there will be, and thus the longer the time to solve the problem. Thus, choosing T_{min} can be a hard decision.

The approaches presented so far both have the issue that adding delay options greatly affects computational time. Adaptive delay, implemented by Liang et al. [70], and delay as a decision variable, implemented by Aktürk et al. [42], go around this problem and offer higher delay flexibility. Liang et al. [70] use a multi-label shortest path algorithm to construct LOFs, with one of the two labels being flight node delay given its predecessor flight nodes. This enabled them to model flight delays as continuous or adaptively, rather than discrete. Instead, Aktürk et al. [42] propose an approach in which departure delay and cruise time are decision variables, however in doing so the model is non-linear. These options are harder to implement but result in a more precise representation of flight delays and better estimates of recovery costs. For example, Liang et al. [70] showed that using discrete flight copies can greatly overestimate recovery costs compared to using continuous adaptive delay.

Modeling flight delays using ready time, used by Bard et al. [24], is a version of flight copies. While constructing the time-band network, they position nodes in the time-space based on flights' departure times and airports. Then, all flights scheduled to depart from airport A are added to all nodes of airport A, not depending on the flights' scheduled departure times. For example, take two flights leaving A, f_1 scheduled for 9:00 and f_2 scheduled for 10:00, then node $(A, 10:00)$ will contain a copy of f_1 with a delay of one hour and the original f_2. Therefore, delays are based on aircraft ready time and dependent on the position of the nodes. Although this approach is simple to implement, it doesn't offer enough delay options and is, to the best of my knowledge, used only by Bard et al. [24]. An overview of delay modeling approaches is provided in Table 4.

Delay approach	Example	Pros	Cons
Flight copies	Yan and Yang [69]	Widely implemented Suited to all network types	Discrete delay Can be computationally expensive
Constructed in paths	Zhang [73]	Continuous delay Many possible delay paths	Determining T_{min} Can be computationally expensive
Adaptive delays	Liang et al. [70]	Continuous delay Delay optimization	Only suited to path-based models
Decision variables	Aktürk et al. [42]	Continuous delay Delay optimization	Model is non-linear Extra decision variables
Ready time	Bard et al. [24]	Cheap implementation Continuous delay	Not enough delay options

Table 4: Overview of delay modeling approaches found in literature.

Modeling aircraft swaps

Operators can perform aircraft swaps and/or maintenance swaps. Aircraft swaps are very common in ARP models. Approaches to implement aircraft swaps vary depending on the formulation of the ARP: flight-based or LOF-based. In flight-based models, two penalties can be applied: one for flight change and another for routing change. Aktürk et al. [42] add a penalty for a flight change directly to the optimization function, while Vink et al. [28] and Bisallon et al. [57] add a penalty to the operating cost of a flight when that flight is operated by an aircraft registration different than scheduled. Vink et al. [28] and Thengvall et al. [4] also add a penalty for a routing change to protect the consecutive operation of flights by the scheduled aircraft. In all flight-based models flight changes and/or routing changes are determined using decision variables that activate the respective penalties.

In path-based models (e.g., Liang et al. [70] and Zhang [73]), LOFs with fewer flight swaps have a lower cost and are thus preferred during LOF selection. Zhang [73] used LOF purity, defined as the number of different scheduled aircraft in one LOF, as a measure of a LOF's swap cost. Furthermore, some authors (e.g., Bratu et al. [6]) allow swaps only between aircraft of the same type, while others (e.g., Eggenberg et al. [45] and Vink et al. [28]) also between different types as they formulate their problem as multi-fleet. In the latter case, cost matrices can be used to assess the feasibility and costs of swapping between different aircraft types.

Swapping planned maintenance is often not considered in ARP models. Eggenberg et al. [45] and Liang et al. [70] are the only authors that make use of planned maintenance swaps as a recovery strategy. In their frameworks, registrations are allowed to swap maintenance if they are of the same type and if all FH, FC, and CY requirements are satisfied after changing the slots.

Modeling other recovery strategies

Other recovery strategies include reserve aircraft (e.g., Arikan et al. [64]), aircraft ferry (e.g., Yan and Yang [69]), cruise speed control (e.g., Akturk et al. [42]) and turnaround operations (e.g., Evler et al. [23]). Although expensive, sparing resources, crew, or aircraft, is a very common practice for airlines. Arikan et al. [64] developed an integrated aircraft, crew, and passenger recovery model in which reserve entities (i.e. crew and/or aircraft) can be used for disruption management. In their model, reserve aircraft registrations are predetermined and modeled as normal aircraft that have no scheduled flights during the recovery window. Although Arikan et al.'s [64] goal was not to find the optimal number of reserve aircraft, doing so is not hard and has been tackled before. **However, it would be worthwhile to create a model that determines the optimal amount of reserve time, rather than reserve blocks, and where to allocate it.** This has the potential to improve literature.

Aircraft ferrying is very expensive and is performed in rare cases. It consists in relocating reserve aircraft to serve flights originally scheduled by AOG aircraft. Yan and Yang [69] implement aircraft ferrying in the TSN by adding position arcs (i.e. flight arcs) from all airports to all other airports. Position arcs are selected with decision variables and have a high cost in the objective function. Carriers can choose suitable times and airports to which to add position arcs, thus reducing the problem size and computational time. Overall, aircraft ferrying is often disregarded in ARP models as it is so economically unviable to airlines. This approach is also similar to the one of Arikan et al. [64] who implement aircraft ferrying in a connection network. They create "external arcs" between flight nodes representing the ferrying of an aircraft to the origin of its next flight after completing its previous flight.

Akturk et al. [42], to the authors' best knowledge, were the first to implement cruise speed control as a recovery strategy (i.e. option to fly faster to recover delay). They modeled cruise speed and cruise time of flights as decision variables, and in the objective function they included an additional fuel and CO_2 cost incurred when flying faster. The objective function is again a recovery cost minimization that minimizes delay, swap, and cruise speed control costs. As both cruise speed and cruise time are decision variables, the model is non-linear and hence requires a conic quadratic optimization approach. The authors were also the first to model delay costs as a step function that increases at predetermined breakpoints, such as minimum times for missed baggage and missed passenger connections. The authors concluded that cruise speed control is an effective delay mitigation strategy, decreasing delay at times by as much as 50% compared to a solution without cruise speed control. However, cruise speed control is not always a viable option due to air traffic management, which is not considered in their work.

Lastly, to absorb delay disruptions, Evler et al. [23] considered executing ground operations faster or partially in an integrated aircraft, crew and passengers cost-minimizing MILP. Turnaround recovery options considered by the authors were altering the sequence of the turnaround process (e.g., fueling, catering, ...) or accelerating a process by a fixed factor. Both of these options are subject to airport resources constraints. In the model, turnaround recovery options incur an additional cost in the objective function, which also considers delay and cancellation costs. This is a novel recovery process in the ARP, however, the authors do not consider routine maintenance checks between flights. **An improvment to the model would be to allow**

cancellation and/or faster execution of maintenance activity during turnaround, and at the same time ensure CY, FH, and FC requirements.

Solution approaches to the ARP

The ARP needs to be solved quickly for it to provide operational guidance to controllers when disruptions become known. Therefore, part of the literature on the ARP has focused on finding faster ways to solve the problem. Approaches can be categorized into three groups: exact, heuristics, and other methods. Hassan et al. [35] provide a more detailed overview of ARP literature from 2009 until 2020 divided per solution approach. The following subsections evaluate papers that pioneered an approach and/or have rare characteristics, such as multi-fleet.

Exact

One of the first to implement an exact method were Yan and Yang [69]. The authors created four evolutions of one MILP model that minimizes different recovery costs depending on the evolution. Using a TSN, their first model minimizes cancellations, the second cancellations and aircraft ferrying, the third cancellations and delays, and the fourth combines all previous plus reserve aircraft. The first two evaluations are small i.e. don't have many decision variables or constraints. Therefore, the first and second model evaluations are solved exactly. However, Lagrangian relaxation with subgradient methods was used to solve the two last evaluations, which are larger due to flight delays being modeled with flight copies. For a problem size of 15 airports, 319 flights, and 12 airplanes the authors could only solve two simpler model versions exactly, posing the main limitation of exact approaches.

Bard et al. [24] implemented an exact approach to solve the ARP based on the time-band network. Their cost-flow integer model minimizes the cost of delays and cancellations, resulting from two disruption types: aircraft unavailability and flight delays. The model first creates the time-band network from the original schedule and predetermined time-band duration, then the mathematical problem is relaxed to set a lower bound on the objective value. Based on the LP-solution, an integer-valued solution that respects the time-band network is created and its costs are assessed. The model is tested with the fleet of an American carrier, with 162 flights, 30 airports, and 27 aircraft, and for different time-band sizes and number of unavailable aircraft. The authors also allow different time-band sizes between hubs and spokes, with a finer resolution at hubs. Considering the relatively small problem size and the fact that the model doesn't apply many recovery options, like aircraft swapping, the problem has a high computational time, $\sim$ 550 seconds. However, it is worth mentioning that the processing time is 10 seconds with 30-minute time-bands, but this is deemed by the authors to overestimate recovery costs.

Akturk et al. [42] and Arikan et al. [64] implement conic quadratic mixed integer programming to solve the model exactly in reasonable times. Both works are the only ARP models that use cruise speed control as a recovery strategy, however, forcing the model to be non-linear and hence requiring the conic quadratic mixed integer approach. Akturk et al. [42] showed that cruise speed control together with the possibility of aircraft swapping does not deteriorate the run time, as the maximum computational time is $\sim$ 250 seconds. Arikan et al. [64] integrated aircraft and crew recovery with a wide range of recovery options at their disposal. Aircraft and crews are modeled as separate entities in a modified connection network which allows unit-specific constraints, such as maintenance and crew rest requirements. However, integrating crew and aircraft recovery affected the run time and solution quality. The model can only find approximate solutions to a large network with 1200 flights in 10 minutes.

Heuristics

A large variety of heuristic approaches have been implemented to solve the ARP. Zhang [73] proposed a two-stage heuristic algorithm to reduce the number of LOFs, in their LOF-based cost-minimizing ARP MILP model. Firstly, all possible paths are created using a depth-first search algorithm in a connection network. In the first heuristic stage, LOFs are scored based on the number of swaps (less is better) and the number of flight legs (more is better). Using a predefined threshold they select the LOFs with the best score. In the second stage, flow balance constraints are aggregated per airport for every aircraft, and the new model is solved to get a set of optimal LOFs. The score of the poorest LOF in the optimal solution is used as a threshold to include all other LOFs with better scores and originally outside the optimal solution to the optimal LOFs set. Reduced costs are then evaluated by solving the relaxed aggregated model, and the final LOFs are selected if their reduced cost is lower than the reduced cost of the threshold LOF. This methodology enabled the author to reduce the number of LOFs by more than 99% and solve a scenario with 638 flights and 44 aircraft in 150 seconds. However, this approach provided time-saving only for large scenarios (above 100 flights).

Liang et al. [70] also developed a LOF-based ARP model in a connection network, which however uses column generation to create LOFs and reduce the problem size. The MILP master problem, which minimizes operating costs (given by the cost of delays, swap, and maintenance swap) and cancellation costs, is solved to get dual variables of the initial LOFs. Then, subproblems are solved for each aircraft individually to find new routes with negative reduced costs to add to the master problem. Two subproblems are available: one with airport slots to model airport capacity and one with swappable planned maintenance. A multi-label shortest path algorithm is used to formulate the subproblems, which allows the authors to model multiple decision variables, such as delay, maintenance requirements, airport slot, and operating cost, when evaluating a connection between two flight nodes in the connection network. This formulation allows for continuous delays and maintenance requirements for maintenance swapping in the LOF generation phase, something unique to their research. Although all of its extensions, thanks to column generation the model solves a scenario with 44 aircraft and 638 flights in less than 360 seconds. However, the model is suited to one single fleet.

A column generation algorithm was also presented by Eggenberg et al. [45] who build upon the work and time-band network of Bard et al. [24] that used an exact formulation. Eggenberg et al.'s [45] problem formulation is unique, as it is suited for unit-specific constraints, unlike the multi-commodity approach which usually struggles with such constraints. A unit (an aircraft, a crew, or a passenger) is associated with a specific recovery network, a set of nodes and arcs, to represent a recovery path for the specific unit. Paths are generated for each unit using a dynamic programming algorithm that considers four arc types: flight arcs, termination arcs, maintenance arcs, and maintenance termination arcs. Once paths are formulated, a column generation algorithm is used to reduce the number of LOFs. This process allows the model to solve large instances very quickly, for example, a scenario with 100 aircraft and 760 flights is solved in just 63 seconds. However, the authors consider only one fleet type.

Khaled et al. [48] presented a multi-criteria recovery model for the tail assignment problem that minimizes the number of impacted airports, the number of flights changed, and the repair costs. This model is one of the few ARP models that optimizes for something other than costs. To generate multiple efficient solutions the authors used the ϵ-constraint method as a Pareto frontier. These solutions are obtained through a parametric variation in the objective function by increment ϵ. For example, 7 efficient solutions are found by forcing the number of impacted airports to 5,6,...,11. The novelty of Khaled et al.'s [48] model is that it requires direct input from the aircraft controller. The five solutions with the best objective value (based on pre-assigned weights for each of the three objectives) are shown to the aircraft controller, which then selects one. Based on the controller's selection, the model automatically adapts the objective weights such that the selected solution has the best objective value. Although the model is relatively quick (in 30 seconds it finds a solution to a test case with 111 flights and 10 aircraft), it has limitations. Firstly, the choice of ϵ is very difficult and will affect the run time and solution quality. Secondly, the ϵ-constraint method only really works when objectives take small integer values.

All models presented so far are static, Vos et al. [19] and later Vink et al. [28] developed dynamic ARP models using an aircraft selection algorithm. Dynamic models are more realistic as they handle disruptions as they happen by building upon solutions from previous disruptions. An aircraft selection algorithm is a rule-based process that reduces the solution space by selecting aircraft most suited for recovery from a disruption event. This is also common practice at OCCs where controllers only use sub-sets of aircraft to solve disruptions. Vos et al. [19] use ground time (the longer the better) as the sole indicator to select best-candidate aircraft. On the other hand, Vink et al. [28] argue that best-candidate aircraft are found based on location rather than ground time. If an aircraft visits both the airport of the disrupted flight and an airport it had previously visited earlier during the day it is considered a recovery candidate. The latter ensures the possibility for the aircraft to switch back to its original routing. Vink et al. [28] also modeled maintenance requirements and connecting passengers. The costs of passengers' missed connections are very smartly added to the cost of delaying a flight in the optimization function and thus require no additional decision variables or constraints. Furthermore, their work, unlike Vos et al.'s [28], provides an initial solution in seconds, which acts as a base for better solutions provided as time progress. Both authors presented complete models (e.g., multi-fleet, disrupted passengers, etcetera) capable of solving the ARP quickly and dynamically, and suited to multiple disruption types, however, the approach on which they are based, the aircraft selection algorithm, is rule-based and will need adjustment depending on the airline. Additionally, like many other ARP models, the models assume that crew is always available to perform flight in the recovery solution.

Other

Given the complexity of the ARP, several other approaches have been attempted. Many writers presented stochastic techniques in light of the inherent uncertainty of the ARP. A stochastic ARP was solved by Arias et al. [50] using a simulation technique in conjunction with constraint programming with the objective to minimize total flight delays and canceled flights while reestablishing the original flight schedule as much as possible. By contrasting the standard deviation from the simulation results with the variation of the probability distribution that was used to create the stochastic delays and the expected propagation, the resilience of the solutions is determined. Recently, Lee et al. [25] suggested a creative combination of reactive and proactive measures to address the ARP. Their study improved recovery strategies that address both actual disruptions and expected future disruptions by forecasting systematic delays at hub airports. The authors integrate a stochastic queuing model, a tool to predict airport congestion, with an integer programming solution to create the disruption recovery schedule. Stochastic approaches are suitable for combining proactive measures into the ARP, as well as combining simulation and optimization methods to address the ARP.

Recently, simulation-based approaches have come into the picture. Rhodes-Leader et al. [36] proposed a symbiotic simulation approach. That is using high-fidelity models to guide the search of the optimization low-fidelity integer program. The low-fidelity model, solved with a commercial optimizer or with the ϵ-constraint method, is used to generate the first set of solutions. These are then the initial solutions to the high-fidelity simulation, in which the schedules are subject to variations in turn-around time, flight times, and maintenance activities.

One last approach to solve the ARP is machine learning. In his thesis work, Hassan [31] developed a decision support tool for the recovery process capable of delivering a solution in less than 120 seconds. The model is based on the MILP models and aircraft selection algorithms developed by Vos et al. [19] and Vink et al. [28]. The novelty of his work is that best-candidate aircraft for disruption recovery are chosen with machine learning, rather than predetermined rules. The author compared the results of the model with the optimal solution obtained without aircraft selection, using a case study with 2200 flights, 827 aircraft, and 8 fleets. The results show that the aircraft selection algorithm decreases computational time by 45% while still find the optimal solution 99% of the time. However, the author fails to compare the machine learning based aircraft selection with the rule-based aircraft selection algorithm. From Vink et al. [28] it was already determined that the aircraft selection algorithm is an effective approach to the ARP, while it was its rule-based formulation that was limiting.

An overview of the solution approaches is presented in Table 5. In general, it is observed that exact approaches cannot provide solutions to large problems quickly enough and are thus suited only to small problems. Furthermore, for the most part, they use few recovery strategies. Other solution approaches, like stochastic and simulation, are less focused on problem size and computational performance. Thus, heuristic models are best suited for quick run times. In particular, the aircraft selection algorithms by Vos et al. [19] and Vink et al. [28] perform incredibly well in this respect. They also model multiple fleets, something not common in ARP literature. Lastly, most authors optimize for delay, aircraft swap and cancellation costs, while very few consider maintenance cancellation and maintenance swap costs. The latter two can greatly improve the recovery solution and flexibility of the recovery strategy; more on this in Table II. Based on these observations, **a valuable addition to literature is a heuristic-based ARP model, either with column generation or an aircraft selection algorithm, that considers maintenance swap and cancellation as recovery actions.**

Maintenance recovery and integration into the ARP

Reactive management of maintenance disruptions, such as new incoming tasks and maintenance resource shortages, are addressed in the Aircraft Maintenance Recovery Problem (AMRP). The AMRP, given a set of disruptions and a slot schedule, determines the aircraft slot assignment and the slot task assignment to minimize disruption costs. Compared to the ARP, the AMRP received significantly less attention in literature with just a few attempts.

Attempts with limited focus on maintenance disruption management are proposed by Yuan et al. [53] and Callewaert et al. [51]. Yuan et al. [53] constructed an operations support tool for task scheduling that considers disruptions by adding an objective to minimize earliness and tardiness of tasks. Callewaert et al. [51] developed a decision support framework for maintenance scheduling that suggests adding or postponing tasks when operations experience delays or are ahead of schedule. Additional tasks are added to workpackages when the timetable is ahead of schedule. Both these works are not full representations of the AMRP as they do not consider task and slot rescheduling or any disruptions.

Paper	**SA**	**Solving method**	**Model size (flights/fleets/ac)**	**CPU sec**	**Objective to minimize**
Akturk et al. [42]	EX	Conic quadratic prog.	207 / 6 / 60	250	D, SW, CSC
Arias et al. [50]	O	Stochastic	51 / NA / 11	NA	D, C, SW
Arikan et al. [64]	EX	Conic quadratic prog.	1429 / 6 / NA	600	D, C, SW, CSC, R, F
Bard et al. [24]	EX		162 / 1 / 27	550	D, C
Eggenberg et al. [45]	HE	Column generation	760 / 1 / 100	NA	D, C, SW, MS
Hassan [31]	O	Machine learning	2200 / 8 / 827	180	D, C, SW
Khaled et al. [48]	HE	ϵ-constraint	111 / 1 / 10	30	Impacted flights & airports
Lee et al. [25]	O	Stochastic	852 / 3 / NA	300	C, R, CSC
Liang et al. [70]	HE	Column generation	638 / 1 / 44	360	D, C, SW, MS
Rhodes-Leader et al. [36]	O	Simulation	83 / NA / 5	50	D, C, SW
Vink et al. [28]	HE	Aircraft selection alg.	600 / >1 / 100	50	D, C, SW, MC, DP
Vos et al. [19]	HE	Aircraft selection alg.	760 / >1 / 100	60	D, C, SW
Yan and Yang [69]	EX		319 / 1 / 12	330	D, F
Zhang [73]	HE	Two-stage heuristic alg.	638 / 1 / 44	150	D, C, SW, MC

Table 5: Overview of solution approaches implemented in ARP literature. Abbreviations in the table: SA: Solution approach; EX: Exact; HE: Heuristics; O: Other solution approaches; D: delay; C: Cancellation; SW: Swaps; MS: Maintenance swaps; MC: Maintenance cancellation; CSC: Cruise speed control; R: Reserve aircraft; F: Ferry aircraft; DP: Disrupted passengers

True pioneers of the AMRP, van Kessel et al. [52], created a MILP model for task and slot rescheduling in a disruptive environment to see how it would compare to airline rescheduling practices. The multi-objective function has a hierarchical structure, based on pre-determined weights, and gives the highest priority to task execution, followed by ground time, schedule changes, aircraft days clean, and interval utilization in this order. Solving the model is facilitated with a heuristic method: a 10-day rolling horizon over a period of 120 days. The model was tested using maintenance data from an established European airline and self-generated scenarios, in which the unexpected arrival of tasks is simulated as a stochastic process. When compared with the airline schedule, the 10-day model achieved a reduction in schedule changes, especially close to the day of operations, improve interval utilization, and a 17% reduction in ground time. Although the results were promising, the model has limitations. Firstly, it only considers hangar tasks. Secondly, it completely disregards network. For example, it could be that a maintenance swap proposed by the model cannot actually be implemented because the aircraft involved has a scheduled flight at the same time. This would require another aircraft to operate that flight, thus a tail swap, or creating a new maintenance slot suited to the aircraft routing. Therefore, based on this observation the increased fleet availability is overestimated. Furthermore, it is impossible to deduce how the extra fleet availability translates into more flights or additional buffers for aircraft recovery.

Tseremoglou et al. [20] assessed the performance of van Kessel et al.'s [52] MILP model against that of a deep reinforcement learning (DRL) algorithm in a disruptive maintenance environment. On top of the corrective and preventive tasks considered by van Kessel et al.'s [52], Tseremoglou et al. [20] took into account prognostic-driven tasks for a set of systems as additional disruptions. The DRL algorithm works by considering each maintenance opportunity and deciding which aircraft to schedule based on the highest value function and using the same constraints as the MILP model. Whereas the MILP model produced more stable maintenance schedules (i.e. with fewer schedule changes) and induced less ground time, the DRL made better use of prognostic-driven tasks utilization and was less computationally intensive. Considering that prognostic-driven systems are few and that the computation gain is at most 18 seconds, the MILP model is a better option to solve the AMRP when one considers typical airline maintenance objectives. Nonetheless, this decision will depend on the airline. However, the DRL does suffer from the same limitation of the MILP model: it fails to consider network.

All AMRP models failed to consider network constraints. Some papers on the ARP do consider maintenance requirements and/or use maintenance as a recovery strategy. The role that maintenance takes in ARP literature spans from it being completely excluded to it being used for recovery as long as maintenance requirements (i.e. FC, FH, and CY requirements) are satisfied. Zhang [73] implemented maintenance in a rudimentary state by including fixed unplanned activities as nodes in the connection network. Maintenance nodes are characterized by a start and end time, an airport, and an aircraft registration. When LOFs are created, maintenance nodes are connected to flight nodes ensuring two constraints: maintenance cannot be delayed, and minimum turnaround time with the two connecting flights. In the LOF selection and tail assignment, constraints guarantee that an aircraft is assigned to a LOF that contains its maintenance activity. Zhang's [73] methodology requires maintenance scheduling (such as through the AMRP) to be solved before the ARP, as such aircraft recovery depends on a fixed maintenance schedule and is thus sub-optimal.

Vink et al.'s [28] ARP model, like Zhang's [73], requires a maintenance schedule to be solved beforehand, however, they allow some maintenance flexibility during aircraft recovery through maintenance cancellation and postponement. The authors make a distinction between fixed and flexible maintenance activities. Fixed activities are harder to reschedule, and are thus more expensive to cancel, while flexible activities can start at different prespecified times and can thus be postponed. Both types of activities are constructed using sets of sequential ground arcs that are preassigned to the respective registration of the maintenance activity. During recovery, failing to assign all ground arcs associated with an aircraft maintenance activity will incur a penalty in the objective function. This formulation allows the authors to model maintenance without any additional decision variable for greater computation speed. A large limitation of this model is that once a maintenance activity is canceled, for example, to favor a flight, it is not rescheduled. This means that the recovery plans found by the model might be infeasible as they can violate maintenance requirements. Furthermore, the authors do not assess the benefits of maintenance cancellations on aircraft recovery.

The main setback of Vink et al.'s [28] and Zhang's [73] ARP models is that maintenance is a fixed activity i.e. it is not scheduled within the ARP. This leads to either sub-optimal solutions with little flexibility, in Zhang's [73] case, or potentially infeasible solutions, in Vink et al.'s [28] case. This limitation is addressed by Khaled et al. [48] who address maintenance as a renewal process of maintenance resources (FH, FC, and CY). In the tail assignment problem, maintenance is scheduled overnight at maintenance stations and is assigned based on three constraints: every airplane should be maintained every D_{max} days, or every FH_{max}, or every FC_{max}, whichever comes first. Maintenance station capacity constraints are also considered, and costs are incurred in the objective function when maintenance is scheduled. This method ensures that maintenance can be scheduled within the tail assignment problem but doesn't provide any flexibility in the ARP nor functions as a recovery strategy.

Very similarly, Bratu et al. [6] integrated aircraft and crew recovery and ensured that maintenance requirements are satisfied when flights are delayed and/or canceled, aircraft are swapped, and reserve aircraft are used. To limit the problem size and attain real-time solutions, maintenance constraints are not enforced directly into the ARP. Instead, given the recovery schedule, it is checked that maintenance critical aircraft (i.e. aircraft that require maintenance at the end of the day) are positioned at maintenance stations at the end of the day. If that is not the case for a maintenance critical aircraft, constraints prohibiting rerouting of this aircraft are added to the ARP, which is then solved again. This process continues until all maintenance critical aircraft are routed appropriately for their end-of-the-day check. Like Khaled et al. [48], this approach integrates maintenance check schedules with the ARP, however, it again fails to provide maintenance flexibility as a recovery action nor does it include maintenance disruptions.

Both Eggenberg et al. [45] and Liang et al. [70] integrated maintenance check scheduling in their ARP models and considered maintenance swapping as a recovery strategy. Liang et al. [70] presented the most complete integration of maintenance into the ARP thus far. In their column generation approach, subproblems are solved for each aircraft individually to find new maintenance feasible routes. In the subproblems formulation, a multi-label shortest-path algorithm allows keeping track of the elapsed FH, FC, and CY since the last maintenance check. Therefore, FH, FC, and CY constraints are enforced when assessing the connection between two flight nodes, and a flight node with a maintenance node. Once a maintenance node is visited the elapsed FH, FC, and CY are zeroed. If a maintenance node is visited by a different registration than planned, a maintenance swap cost is incurred in the connection. The effect of maintenance swaps is impressive. A reduction in recovery cost by up to 60% is observed, however, achieved with a maintenance swap cost of zero which is not entirely true. Although the sizable results, Liang et al. [70] do not consider other forms of maintenance flexibility, such as maintenance cancellation and maintenance postponement, nor do they assign checks based on open tasks, but rather use thresholds.

An overview of the literature discussed in this section is shown in Table 6. The following observations can be made. Firstly, maintenance flexibility for aircraft recovery, through maintenance slot swaps, postponement, and/or cancellation, has not been modeled to its full potential. There is currently no ARP model that makes use of multiple maintenance recovery strategies, and including them could lead to better recovery solutions. Secondly, only maintenance check scheduling has been integrated into the ARP, either as a fixed requirement or using predetermined FC, FH, and CY thresholds. Incorporating task scheduling, instead of check scheduling, is a complete representation that could result in less ground time for maintenance and more for recovery purposes. Thirdly, no model has yet incorporated the ARP and the AMRP. A model that would consider all types of disruptions to create an optimal network and maintenance recovery plan simultaneously might find better solutions. Based on these observations, **a very valuable addition to literature is an ARP model that considers all maintenance recovery strategies and disruptions, and uses task scheduling theory to schedule maintenance opportunities, and in so doing integrates the AMRP.**

	Model domain		Maintenance characteristics				
Paper	**ARP**	**AMRP**	**CS**	**TS**	**MC**	**MS**	**MP**
Bratu et al. [6]	✓		✓				
Callewaert et al. [51]		✓		✓			
Eggenberg et al. [45]	✓		✓			✓	
Khaled et al. [48]	✓		✓				
van Kessel et al. [52]		✓		✓	✓	✓	
Liang et al. [70]	✓		✓			✓	
Tseremoglou et al. [20]		✓		✓	✓	✓	
Vink et al. [28]	✓		✓		✓		✓
Yuan et al. [53]		✓		✓			
Zhang [73]	✓		✓				

Table 6: Overview of literature on reactive recovery with maintenance consideration. Abbreviations in the table: ARP: Aircraft recovery problem; AMRP: Aircraft maintenance recovery problem; CS: Check scheduling; TS: Task scheduling; MC: Maintenance cancellation; MS: Maintenance swap; MP: Maintenance postponement.

Concluding remarks on literature

Reactive disruption management, from the lens of network, is represented in literature by the ARP. The ARP is a relatively old problem, as it was first attempted in the 1980s, and thus most disruption types and network recovery actions have been modeled. Airport restrictions and aircraft unavailability are the most common disruption types, while flight delays, cancellations, and swaps are popular recovery strategies. Recent research has focused on improving computational performance and transitioning from static to dynamic ARP models. This has led to the exploration of a vast set of heuristics and other approaches. For example, the aircraft selection algorithm, as a heuristic method, has proven very effective, but it is limited by its rule-based nature.

Like in proactive disruption management, maintenance is often treated separately from aircraft recovery. While various network-based recovery strategies have been implemented, maintenance strategies like slot cancellation, postponement, and swap have yet to be fully utilized, and maintenance disruptions like DDs and resource shortages are not considered in ARP models. Similarly, AMRP models fail to consider network constraints. Therefore, there is currently no link between network disruption management and maintenance task scheduling through an integrated ARP and AMRP model. An ARP model that considers both maintenance and network disruptions and recovery actions and integrates maintenance task scheduling would provide a more complete representation of recovery costs, making it a promising avenue for future research.

Conclusion

The airline industry is prone to disruptions, with few flights operated as scheduled. Today, labor shortages and other factors, make this problem ever more relevant to airlines throughout the world. Therefore, airlines are increasingly interested in disruption management models to build stable, flexible, and recoverable schedules. This research aimed to explore existing literature to investigate current and new possibilities for airlines to develop smart aircraft schedules with built-in stability and flexibility for improved aircraft recovery. The second objective of this study was to examine how maintenance scheduling can be integrated into aircraft disruption management, and how maintenance flexibility can contribute to smart schedules and improved aircraft recovery.

Disruption management literature is divided into proactive and reactive disruption management. Proactive disruption management focuses on constructing disruption-resistant schedules that are optimized for robustness rather than operating profits, while reactive disruption management focuses on schedule recovery after disruptions occur optimized for recovery costs.

Smart schedules are built using proactive disruption management strategies which are categorized into two approaches: proactive absorption, which aims for schedule stability, and proactive recovery, which aims for schedule flexibility. Literature is primarily focused on creating schedule stability through delay robust routing and/or flight retiming, rather than flexibility through facilitating swaps. Furthermore, apart from Burke et al. [13], possibilities to combine stability and flexibility, for example by reallocating buffer time to maximize swapping opportunities, have been overlooked. Nonetheless, the choice between schedule stability and flexibility is often airline-specific. Increasing stability, for example through the use of buffer time, results in inefficiency but stable operations. On the other hand, flexibility, such as facilitating swapping opportunities, results in a more intense recovery phase but more efficient resource utilization.

Maintenance is often overlooked in the context of proactive disruption management. Frequently treated as a fixed interval activity, maintenance is overshadowed by network operations that take center stage. Aside from Lagos et al. [10], maintenance task scheduling is never modeled together with network resource allocations. Integrating the two could allow the evaluation of maintenance schedule design choices (e.g., check frequency and the minimum number of days without any due tasks after a check) on the robustness of a network schedule. Furthermore, maintenance flexibility methods, such as cancellation, swap, and retiming of slots, are frequently ignored despite their potential to enhance schedule robustness.

In the context of maintenance robustness, like for network, the objective has often been to ensure stability. This was done through task duration uncertainty scheduling and minimization of ground time and scheduling costs. Rather a flexibility objective, for example setting a minimum number of days without any due tasks after a check (clean days) or balancing slot workload to facilitate swapping, has to the best of my knowledge never been tested.

Like in proactive disruption management, maintenance is often ignored or treated as a fixed interval activity during aircraft recovery. Efforts have concentrated on developing new solution approaches, such as the aircraft selection algorithm, to make operational models that can be solved dynamically and quickly, rather than the inclusion of maintenance for completion and/or as a recovery outlet. Indeed, most ARP models only use network-related recovery strategies like aircraft swaps, flight cancellations, and flight delays, while they ignore maintenance-related recovery actions such as the postponement and swapping of slots. On the other hand, in maintenance recovery models, network requirements are disregarded. As a result, maintenance and network recovery are presently disconnected, highlighting the need for a model that bridges this gap.

This literature review sheds light on a research gap. A tactical tool capable of performing task scheduling and aircraft tail assignment simultaneously to create a robust and cost-optimal schedule that promotes flexibility. Robustness, assessed through fleet health and buffer time, can be introduced through flexibility in the schedule, for example, done with a days clean target and/or a minimum buffer time requirement. Thus, the thesis objective is:

Developing a tactical tool that allocates network and maintenance resources simultaneously in order to assess the impact of flexibility on airline schedule robustness.

The corresponding research question to the research objective, and relevant sub-questions, are:

To what extent can airline schedule robustness be improved by means of flexibility using a tactical tool that simultaneously allocates network and maintenance resources?

1. How are network and maintenance resources allocated by airline planners?
2. How is flexibility currently embedded in network and maintenance resource allocation?
3. What methodology can be implemented to integrate maintenance and network scheduling?
4. What are the main objectives to promote flexibility during network and maintenance resource allocation?
5. To what extent can a tool that integrates maintenance and network resource allocation provide feasible airline schedules?

III

Supporting work

1

Maintenance slots generation

This appendix explains the methodology employed by the framework to generate maintenance slots, a process derived from the aircraft maintenance tasks backlog and the airline's maintenance schedule. The decision to create maintenance slots rather than directly utilizing those from the airline's schedule is driven by the need for flexibility. This allows for the adjustment of maintenance time by varying the days-clean target, subsequently influencing fleet availability as well. The implementation of this two-step process, as illustrated in Figure 1.1, involves the determination of the required maintenance time for each aircraft on any given day within the scheduling window. Subsequently, using the calculated maintenance time, the framework creates maintenance slots. Further insights into both sub-processes will be given in the subsequent sections.

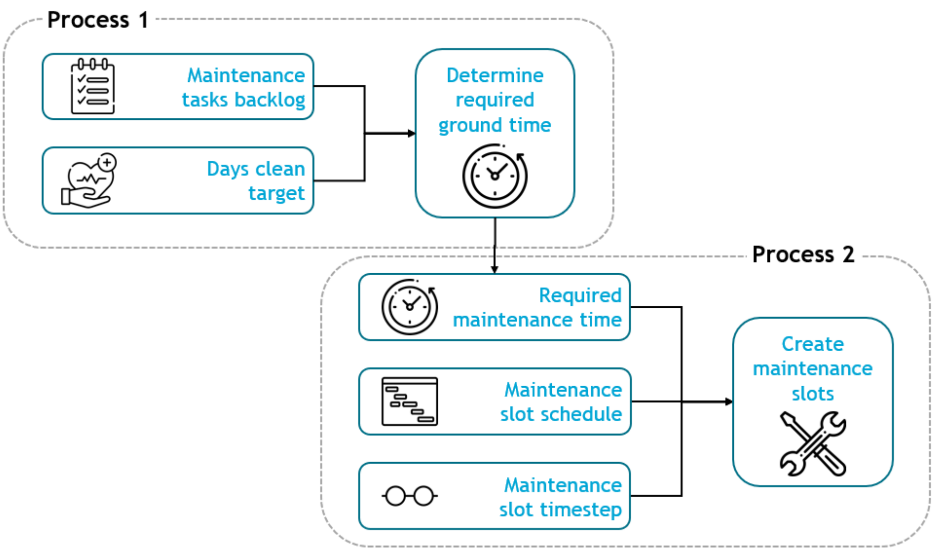

Figure 1.1: Illustration of the slot generation process implemented by the framework.

1.1. Required maintenance time

By using the aircraft's tasks backlog and adhering to a days-clean target, the model calculates the necessary maintenance time for the aircraft on any given day within the maintenance scheduling window. This process ensures that priority tasks, tasks with fewer days left from their due date than the days-clean target, are factored into the daily ground time calculation. This approach, coupled with the high deferral costs beyond the days-clean target, enables the framework to allocate sufficient space for all priority tasks when planning maintenance, ultimately working towards achieving the target health after the execution of the maintenance slot.

This defers slightly for corrective tasks such as MELs and NSREs. Given their objective of being scheduled as promptly as possible, these tasks are treated as priority tasks as soon as they arrive and are ready for scheduling. Unlike preventive tasks, waiting for the days left in the task's interval to be less than the days-clean target would go against corrective scheduling aims.

The framework computes the required maintenance time based on the list of priority tasks. The required ground time is the largest between the sum of all priority tasks' labor hours divided by the slot scheduled FTE (3/hour and 4/hour for hangar and platform slots respectively) or the longest task duration from all priority tasks. A detailed example calculation is given in Table 1.1.

Open task	Task type	Work type	Arrival day	Due day	Schedule from day	Priority task	Labor	Duration
T1	MRI	H	-20	0	-10	True	4 h	4 h
T2	MRI	P	-31	1	-9	True	1 h	1 h
T3	MEL	P	-95	5	-95	True	0.5 h	2 h
T4	MRI	P	-20	10	0	True	1 h	1 h
T5	MRI	P	0	10	0	True	0.5 h	0.5 h
T6	NSRE	H	-20	10	-20	True	4 h	10 h
T7	MRI	H	-100	20	10	False	6 h	6 h
T8	MEL	H	-60	40	-60	True	6 h	10 h

Table 1.1: Example of how the framework calculates the required maintenance time. Given that today is day 0, the days-clean target is 10 days, the available FTE per hour is 3, and the open tasks T1 through T8 with specified characteristics, the required hanger maintenance time is 10 hours ($max((4+1+0.5+1+0.5+4+6)/3, max(4,1,2,1,0.5,10,10))$, while the required platform maintenance time is 0 hours, as a platform slot would not guarantee the days-clean target as some tasks require the hangar.

1.2. Maintenance slots generation

Utilizing the computed maintenance time and the airline's slot schedule, of which an example is shown in Figure 1.2, the framework generates maintenance slots tailored to a specific registration. On any given day, the slot durations match the registration's required maintenance time. These slots are created within the confines of the airline's existing slots that could have been assigned to the same registration. For example, on a Monday the model can create a 4-hour 787 maintenance slot within slots 3, 5, or 6, or alternatively, it can create a 16-hour 787 slot by merging slots 3, 4, and 6 as these are overlapping. Note that slots can be merged only if they are at the same location and for the same aircraft type. Within the existing schedule, slot copies are spaced every twenty minutes, offering a good compromise of flexibility and model run performance.

Additionally, the framework can generate slots outside the airline's predefined schedule under two conditions. Firstly, when the aircraft's required maintenance time results in a slot that cannot fit into the schedule, even by merging slots. Planners typically address this by either creating a distinct, longer slot or occasionally extending existing slots. Secondly, when an aircraft has tasks going due the following day, but no slot exists in the airline's schedule to plan these tasks. This is a measure against infeasible solutions, which mirrors the practices of planners. These off-hour slots, created outside the standard schedule, do not abide by schedule constraints and incur a penalty in the framework.

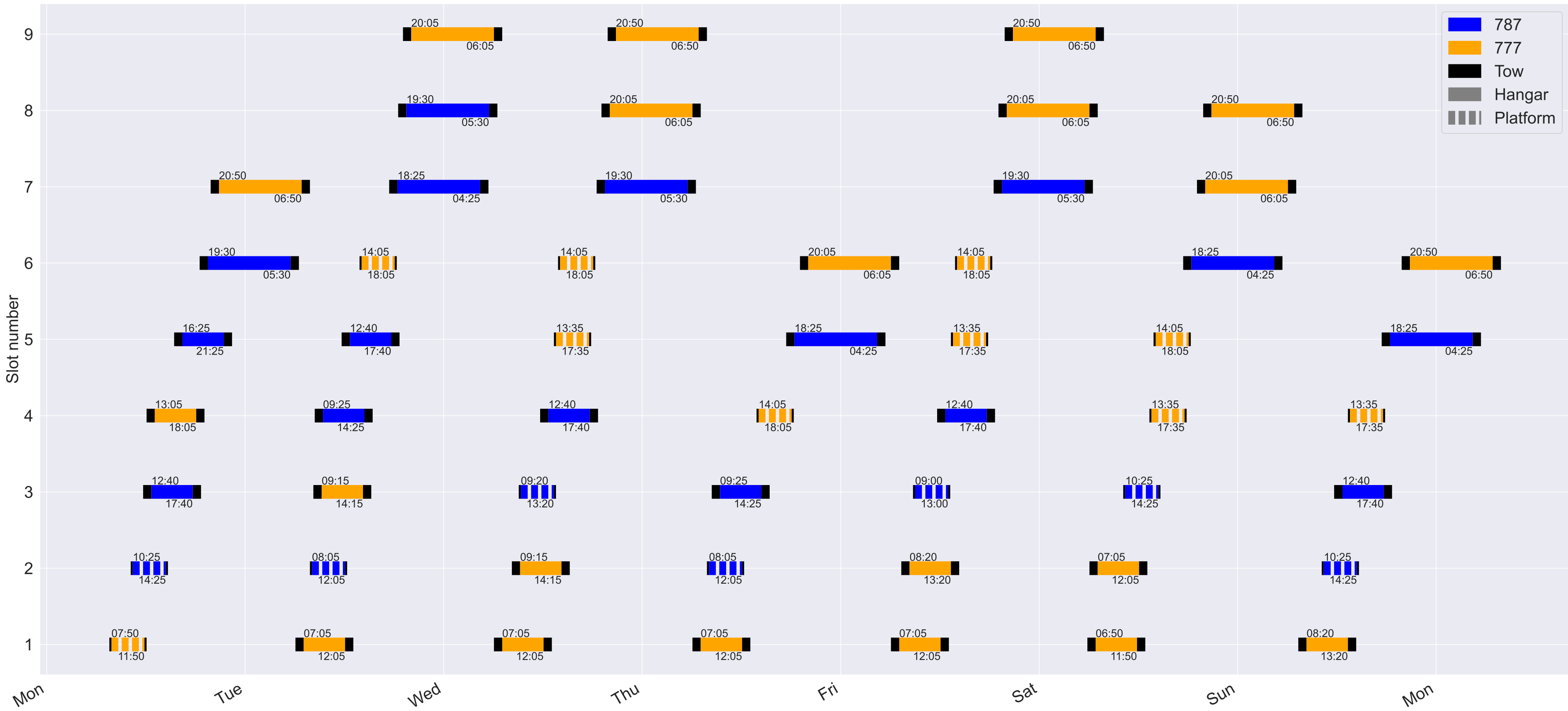

Figure 1.2: Example slot schedule provided by the case study airline.

2

Model verification

Verification is essential to guarantee that the model makes decisions as designed. Therefore, this appendix chapter delves into model verification, treating the framework's tail and task assignments decision process. Verification is performed by running simplified tests, to which the optimal solution is trivial, in order to partition all different scheduling rules and test them independently. The following sections touch upon individual tests, aimed at different scheduling rules, which include rotations cancellations, quick turnaround, aircraft airworthiness, the trade-off between maintenance and flying, and task interval utilization.

2.1. Rotation cancellation

The framework, as a last resort option, can cancel rotations. The following test is primarily aimed at guaranteeing that the model only cancels a rotation if and only if it has no other option. Moreover, it shows that rotations either have to be assigned to a tail or canceled and that flow balance has to be respected, hence an aircraft cannot be assigned to overlapping jobs.

The simple test involves only one aircraft and two overlapping rotations. Consequently, the model is forced to cancel one rotation. The resulting plan, presented in Figure 2.1, shows that the model correctly assigns the BOM rotation and cancels the PTY rotation, the longest, and hence more expensive of the two. Therefore, the model cancels rotations only if there is no other alternative, prevents overlaps, and either assigns or cancels all rotations.

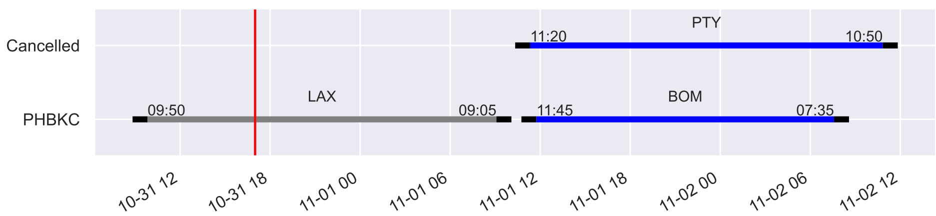

Figure 2.1: Plan created by the model showing that cancellations are made only as a last resort option and that flow balance is respected.

2.2. Quick turnaround

The model, like airline planners, can speed up the turnaround process of a rotation to prevent cancellations. However, both are restricted to a maximum of two quick turnarounds per day. The subsequent test shows that the model only assigns quick turnaround rotation to prevent a cancellation, and not to gain fleet availability, and that is restricted to a maximum of two per day.

The test is composed of four aircraft, all arriving at 9:00 AM, and four rotations, three of which require a quick turnaround to be assigned as they depart before 11:00 AM (2 hours of minimum TAT). As shown by the plan in Figure 2.2, the model correctly assigns two quick turnaround rotations (JFK and DEL) but is forced to cancel the SFO, the longest of the three rotations, as otherwise, it would exceed the maximum daily allowed quick turnarounds. Moreover, no quick turnaround is used for the ORD rotation as it is not required as it departs at 11:00 AM, hence showing that the model does not speed up the turnaround process just to gain fleet availability.

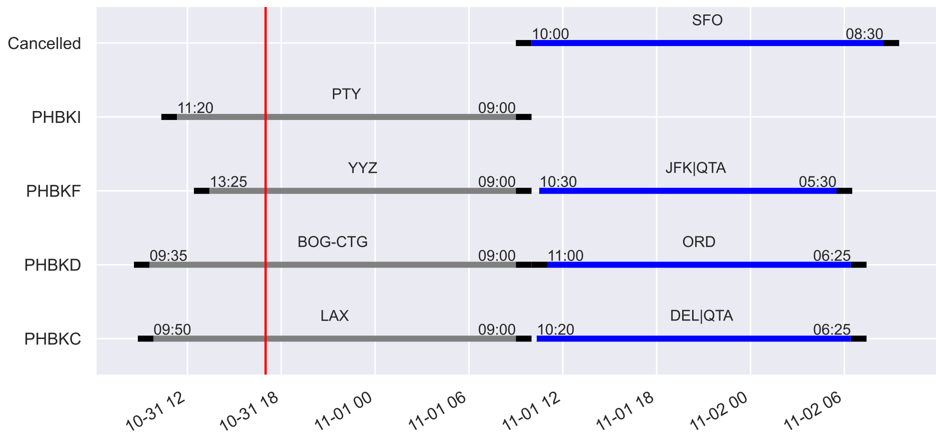

Figure 2.2: Plan created by the model showing that a maximum of two quick turnaround rotations per day are assigned and only to prevent cancellations.

2.3. Aircraft airworthiness

An aircraft cannot be assigned to a rotation if it has any outstanding tasks that go due before the arrival time of that rotation unless those tasks are scheduled in a maintenance slot before the rotation's departure time. This constraint guarantees the airworthiness of the aircraft. If maintenance cannot be scheduled, the framework is forced to cancel all subsequent rotations.

The following verification test, shown in Figure 2.7, guarantees that the aircraft is always airworthy. For it only one aircraft, three consecutive rotations (from 1/11/2023 - 3/11/2023), and one MEL of 5 hours, whose due date will be varied, are considered. If the task is due on November 1, as in Figure 2.3, the model must schedule it and cancel the PTY rotation as it does not fit in the schedule. Note that the model uses the first maintenance opportunity to minimize ground time waste.

Changing the due date of the task to November 2 results in the same schedule, as seen in Figure 2.4. The task must still be scheduled to guarantee the aircraft's airworthiness for the JFK and SFO rotations, and it is scheduled on November 1 as it is cheaper to cancel the PTY rotation and to plan corrective tasks as soon as possible. The plan changes when the due date is set to November 5, in which case the task is planned after all rotations to prevent any cancellations, as shown in Figure 2.5. Lastly, in Figure 2.6, if the task due date is set back to November 2, but all maintenance opportunities are removed, the framework is forced to cancel all subsequent rotations (JFK and SFO) as airworthiness cannot be guaranteed.

2.4. Maintenance-flying trade-off

Airlines prioritize flying over maintenance unless maintenance is strictly required, either to maintain aircraft airworthiness or to prevent too large task backlogs. The same rule is implemented in the framework, as rotation cancellation is significantly more expensive than task deferral. The following test checks if this rule is implemented correctly.

The test involves one aircraft, one rotation, and one task due on the second day of the plan, however with only one scheduling opportunity on the first day. As shown in Figure 2.8, the framework correctly prefers to schedule the rotation and defer the task. However, if the task is divided into 1000 small tasks, with again the same single maintenance opportunity, the model plans the maintenance slot and cancels the rotation as it is cheaper than deferring all tasks, as presented in Figure 2.9. Therefore, the model prevents tasks from accumulating and applies the correct logic when trading flying with maintenance.

2.5. Task interval utilization

One of the framework's objectives is to optimize task interval utilization, by scheduling preventive tasks as close to their due dates, while corrective tasks as soon as possible. Additionally, like airline maintenance

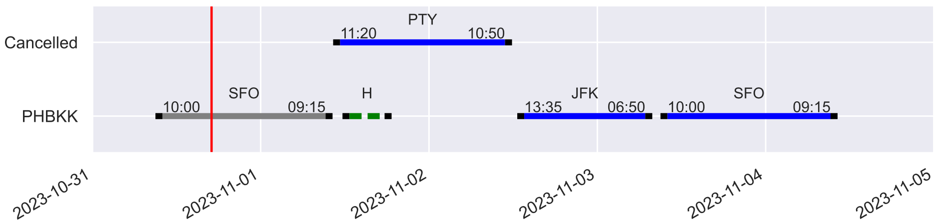

Figure 2.3: Task due date is set to November 1, forcing the model to cancel the PTY rotation to schedule the task and guarantee airworthiness for the subsequent rotations.

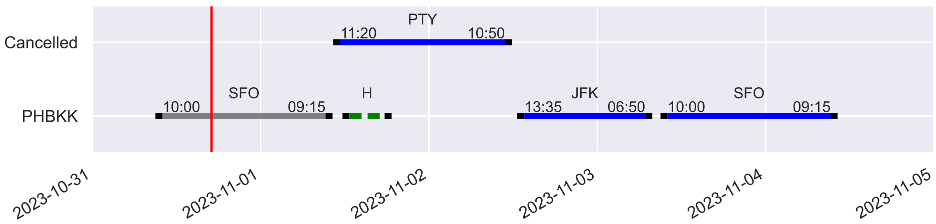

Figure 2.4: Task due date is set to November 2, forcing the model to cancel the PTY rotation to schedule the task and guarantee airworthiness for the subsequent rotations.

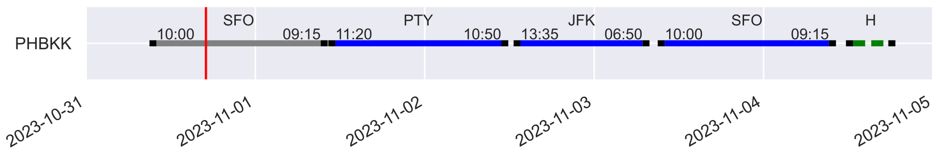

Figure 2.5: Task due date is set to November 5, hence the model can schedule the task after all rotations.

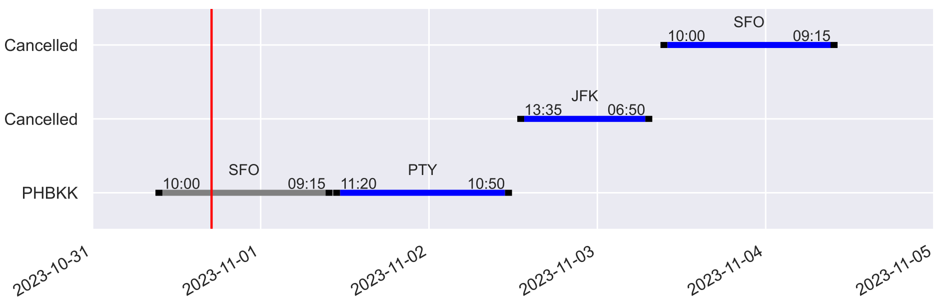

Figure 2.6: Task due date is set to November 2, however, all maintenance opportunities are removed, forcing cancellations of the two subsequent rotations, JFK and SFO, as the aircraft is not airworthy.

Figure 2.7: Framework plans to show that aircraft airworthiness is guaranteed.

planners, the framework implements a hierarchy to prioritize certain task types over others. The following task shows that the model optimizes the task interval utilization and currently enforces the task hierarchy, which is set through the $C_{Type,t}$ coefficient.

For this test only one aircraft and two maintenance platform tasks are needed, an MRI (a high-priority preventive task) due on November 2 and an NSRE (a low-priority deferrable corrective task) due on November 10. For simplicity, it is assumed that it is optimal to schedule preventive tasks at their due dates rather than at H_{min} days before the due date. The test's results, shown Figure 2.11, reveal that the framework correctly prioritizes the MRI task by assigning a maintenance slot on November 2. However, as shown in Figure 2.12, if the MRI task is removed, the model plans the maintenance slot as soon as possible to optimize the interval

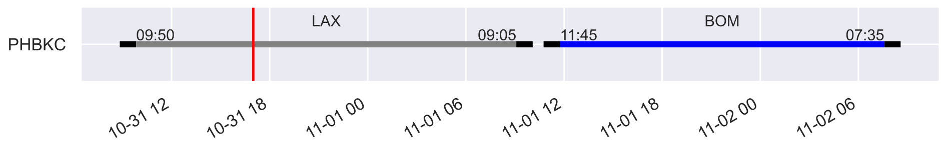

Figure 2.8: Framework plan showing that flying is prioritized over task deferral.

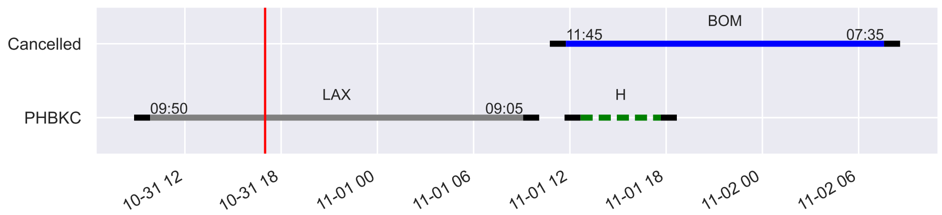

Figure 2.9: Framework plan showing that to prevent a large task backlog a rotation is canceled to schedule maintenance tasks.

Figure 2.10: Framework plans that show that correct choices are made when trading flying with maintenance.

utilization of the NSRE task. Therefore, the model correctly prioritizes task types and task interval utilization.

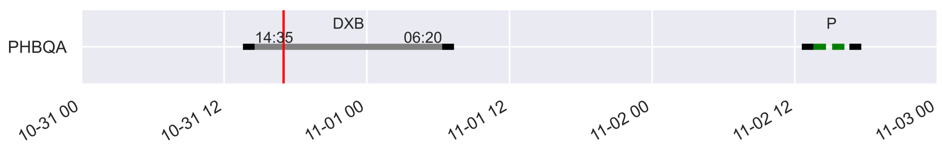

Figure 2.11: Plan showing that the model prioritizes the interval utilization of the MRI task due on November 1, over that of the NSRE task due on November 10.

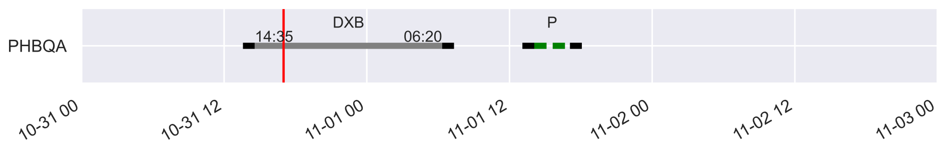

Figure 2.12: Plan showing that when the MRI task is removed, the model does schedule the NSRE task as soon as possible.

Figure 2.13: Framework plans that task types are prioritized and task intervals are optimized correctly.

3

Model validation

This appendix delves into model validation to understand if the framework's decisions resemble those of airline planners. Validation is performed on the tail and task assignment.

3.1. Tail assignment validation

The framework creates daily plans using the same inputs as the airline's planners. To provide decision support to the airline's planners, the model should make feasible tail assignments, and hence could be operational. Moreover, any significant deviations in the plans must be justified.

The validation of the tail assignment process involved collaboration with the airline's planners, who analyzed thoroughly a subset of plans generated by the framework. Presented with the framework's plans alongside their own, without knowledge of which plan belonged to whom, the planners encountered difficulty in distinguishing between the two. While they acknowledged the differences, they expressed an understanding of the rationale behind decisions made in both sets of plans. The planners asserted that both sets could potentially represent plans created by their teams. Consequently, based on their experience and observations, it was concluded that the framework generates distinct yet feasible and operationally sound plans that assign maintenance when required, schedule all rotations (654 in the 7-day case study time frame), and consider the airline's tail assignment restrictions.

While the model generates different plans in general, in complex scenarios characterized by limited space and consequently few tail assignment options, the model produces plans that closely resemble those created by the airline's planners. An illustrative example is presented in Figure 3.1. In this scenario, the constrained schedule compels both the planners and the framework to assign a quick turnaround to the JFK rotation and assign eight out of ten rotations on November 1 to the same registrations.

Although the framework's plans differ from the real ones, these differences do not compromise feasibility. According to the planners, these differences are feasible and result in scheduling advantages. From a maintenance perspective, the framework, thanks to its innovative slot generation process, can allocate the same maintenance work package in a smaller slot and thereby enhance fleet availability. An example is illustrated in Figure 3.2 for the PHBQA tail. From the network perspective, the framework plans more ground time after rotations to mitigate any potential delay. Examples are visible in Figure 3.3, where more ground time is planned after the MEX rotation on November 3 by changing the tail assignment of the BOS rotation, and more ground time is planned before PHBHL's maintenance slot on November 7 by planning the AUS rotation instead of the SJO rotation before the slot. None of these changes resulted in infeasible plans but rather contributed to the gains obtained when using the framework.

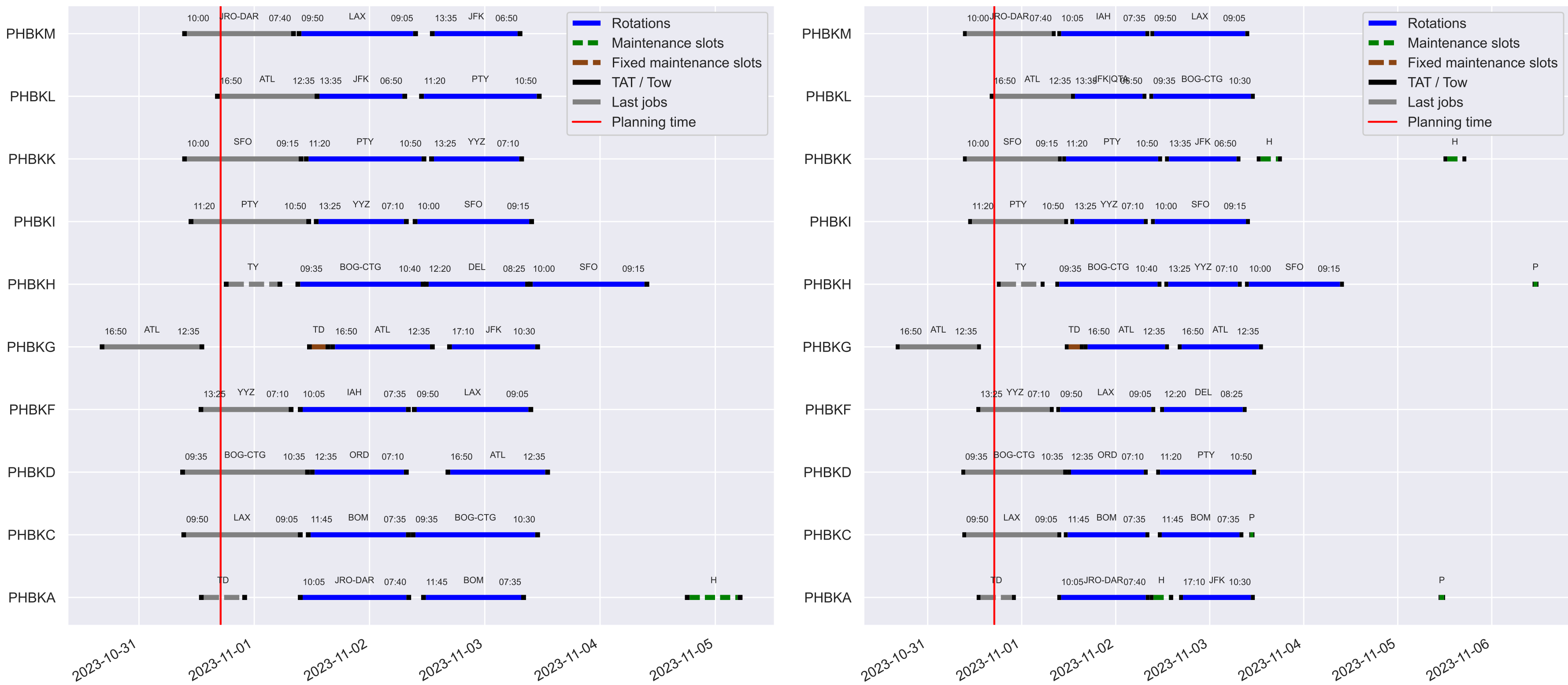

Figure 3.1: Left: Tail assignment performed manually by the airline's planners on October 31, 2023, for the Boeing 781 fleet. Right: Tail assignment performed by the framework on October 31, 2023, for the Boeing 781 fleet, showing a similar tail assignment to prevent cancellations.

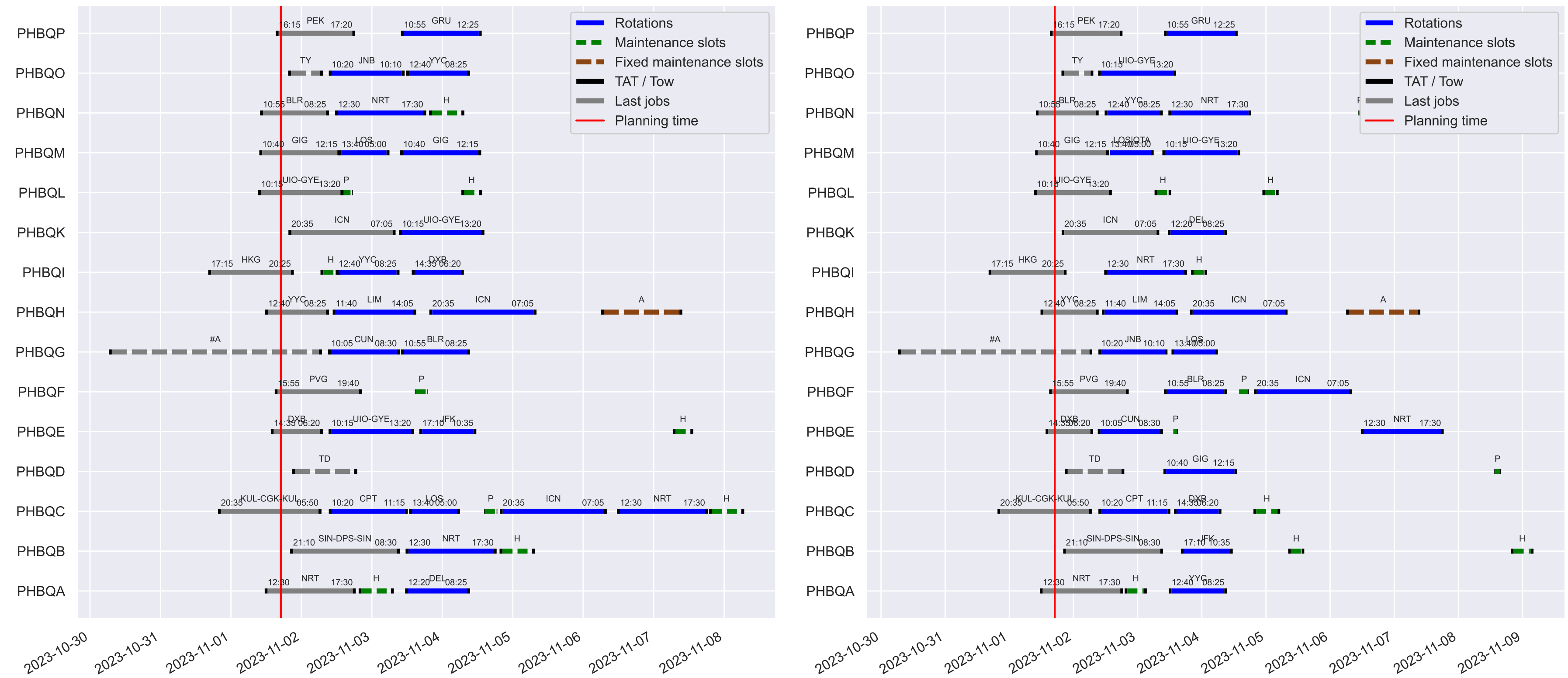

Figure 3.2: Left: Tail assignment performed manually by the airline's planners on November 1, 2023, for the Boeing 772 fleet. Right: Tail assignment performed by the framework on November 1, 2023, for the Boeing 772 fleet, showing a shorter slot planned for PHBQA.

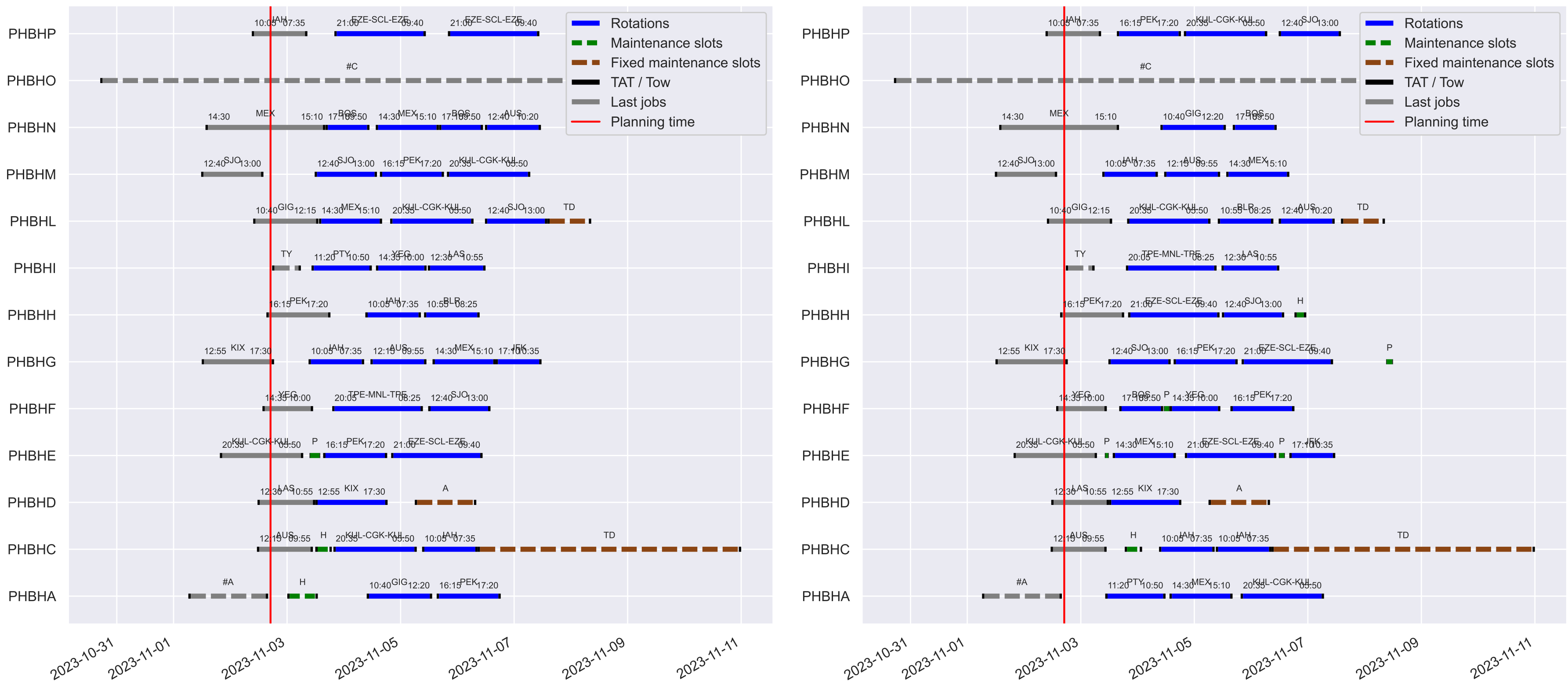

Figure 3.3: Left: Tail assignment performed manually by the airline's planners on November 2, 2023, for the Boeing 789 fleet. Right: Tail assignment performed by the framework on November 2, 2023, for the Boeing 789 fleet, showing extra ground time planned after the MEX rotation on November 3 and before the maintenance slot of PHBHL on November 7.

3.2. Task assignment validation

In this section, the model's task assignment capabilities are validated. Validation is performed in collaboration with the airline's maintenance planners, which expressed as the main priorities the number of scheduled tasks and fleet health. Therefore, the framework aims to resemble the plans predominately in these two aspects.

Figure 3.4 portrays the results obtained by the framework when varying the days-clean target. Results are as expected. Scheduled tasks increase with the days-clean target, as task deferral becomes more expensive, thus resulting in more maintenance time. Preventive task interval utilization logically worsens with increased days-clean target, as tasks are scheduled sooner. On the other hand, although it may appear incorrect that corrective task interval utilization is mostly unchanged, it is an expected result as these tasks are often planned well before the days from the due date become fewer than the days-clean target.

Planning 10 days clean closely aligns with the manually made airlines plans, based on the average scheduled tasks. Nevertheless, the framework tends to allocate more tasks within the same maintenance ground time, as planners use fixed slots and often schedule fewer tasks per slot to mitigate the risk of work extending beyond the allotted ground time. Consequently, the framework's labor utilization is higher, 81% compared to the manual 67%. However, as the model plans roughly the same amount of tasks, it results in a similar fleet health profile for the subsequent 7 days of the plan, as shown in the main paper. Thereby suggesting that the model and the airline's maintenance planners follow similar task assignment priorities.

The task internal usage of the airline's maintenance planners is slightly different from that of the framework, as shown in Figure 3.5. As preventive tasks are planned later, it performs marginally better; however, as corrective tasks are scheduled earlier by the airline, it performs worse. As it is not their top priority, maintenance planners were nevertheless pleased with the model's task interval utilization. Moreover, they plan with anything from 10 to 14 days clean, thus increasing the range of possible interval utilization.

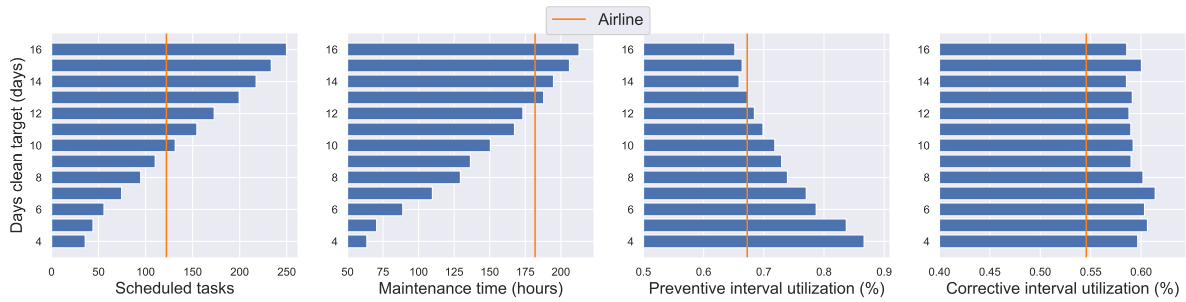

Figure 3.4: Number of average scheduled tasks, planned maintenance ground time, and preventive and corrective interval utilization in a daily plan obtained by the framework with different days-clean targets.

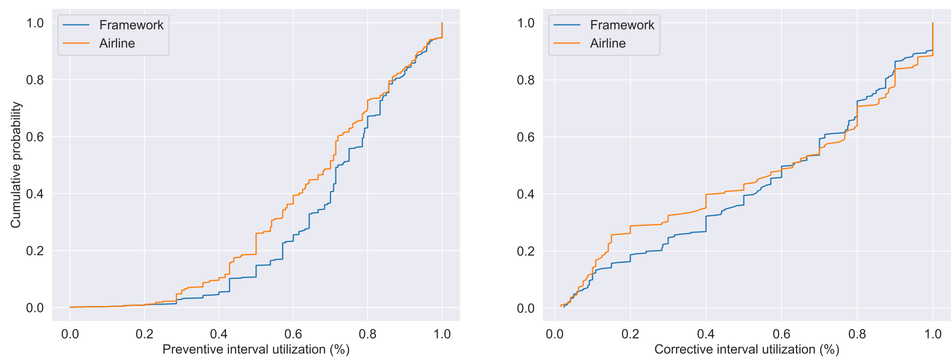

Figure 3.5: Cumulative distribution of the preventive and corrective task interval utilization from the airline's and framework's task assignments.

A great part of the framework's reduction in maintenance time is attributable to the model's slot generation process. Whereas the model can dynamically determine the slot duration based on the aircrafts specific maintenance time requirements, as explained in chapter 1, planners adhere to predetermined slots. For example, when an aircraft requires 6 hours of maintenance the model creates 6-hour slots, while planners are forced to use predefined and longer 9-hour slots resulting in 3 extra hours of maintenance. Indeed, in Figure 3.6 it is clear how the planners adhere to fixed-length slots while the framework has more freedom in the slot duration, contributing to the reduction in maintenance time.

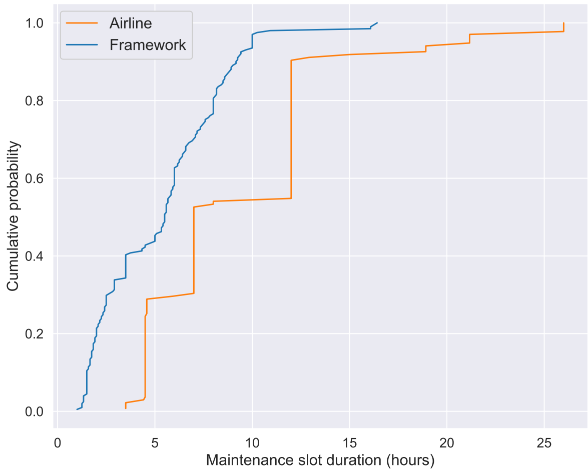

Figure 3.6: Cumulative distribution of the duration of maintenance slots planned by the framework and by the airline's planners.

4

Scheduling policies sensitivity analysis

The main objective of the framework presented in this thesis work is to provide decision support to airline planners and help them create efficient, resilient plans for the day of operations. However, the framework can also be used to assess different planning policies to understand which would be most beneficial to the case study airline. This appendix delves into this sub-objective, by touching upon the days-clean target in section 4.1, the delay mitigation parameter in section 4.2, and planning flexibility in section 4.3.

4.1. Days-clean target

The days-clean target has a direct effect on fleet health and maintenance time, and consequently on fleet availability. Therefore, a trade-off exists between fleet availability and fleet health that can be discovered by varying the days-clean target. This trade-off was introduced in the scientific paper to decide the optimal days-clean target based on operational availability. In this section, that decision is reviewed in greater detail, to better understand the relationship between fleet health and fleet availability by also considering schedule flexibility.

Airlines aim to have a flexible schedule to facilitate disruption recovery on the day of operations. In this section, flexibility is measured with percentage of incoming aircraft with at least one swapping possibility on the day of operation, and hence have a recoverable LOF. Multiple conditions need to apply for two aircraft's LOFs to be swappable. Firstly, swaps are allowed only within the same aircraft subtype to prevent cargo and passenger spillage, and crew problems. Secondly, the swap will not induce any delays in either of the aircraft's subsequent jobs. Lastly, aircraft must be able to swap back to their original LOFs before any critical maintenance activity (i.e., fixed maintenance slots and maintenance slots where the aircraft's health before the slot is less than H_{min}, in this case 3 days). Any non-critical maintenance slot can be canceled and postponed. Hence, flexibility is expected to improve with higher fleet availability, but also with higher fleet health, or better with a lower percentage of the fleet with health below 3 days, which increases potential buffer time by increasing the number of maintenance slots that can be postponed.

The scientific paper conducted a brief analysis that led to the selection of a 10-day days-clean target. This decision aimed at maximizing fleet availability and was justified as it aligns with the case study airline's days-clean target. The analysis, summarized in Figure 4.1, revealed that lower and higher fleet health led to increased maintenance time, respectively due to more frequent maintenance visits and greater tasks lost interval, consequently leading to reduced fleet availability. However, the previous analysis did not account for the benefits of higher fleet health obtained when increasing the days-clean target, which will now be explored through the newly defined measure of schedule flexibility.

Planners ideally seek to maximize both fleet health and fleet availability. However, achieving high health results in more maintenance time, leading to reduced fleet availability. Therefore, the focus can shift to maximizing schedule flexibility. Figure 4.2 illustrates the three-way trade-off between fleet health, fleet availability, and schedule flexibility. In this case, fleet health is replaced by low-health aircraft, which represents the daily average percentage of the fleet with health below 3 days, as it is a better indicator of schedule flexibility. The analysis involved simulating 5-week-long scenarios with varying days-clean targets using a rolling horizon technique. As anticipated, swapping opportunities and thereby flexibility improve with high fleet availability and low low-health aircraft. Notably, there is a significant difference in flexibility between low and high

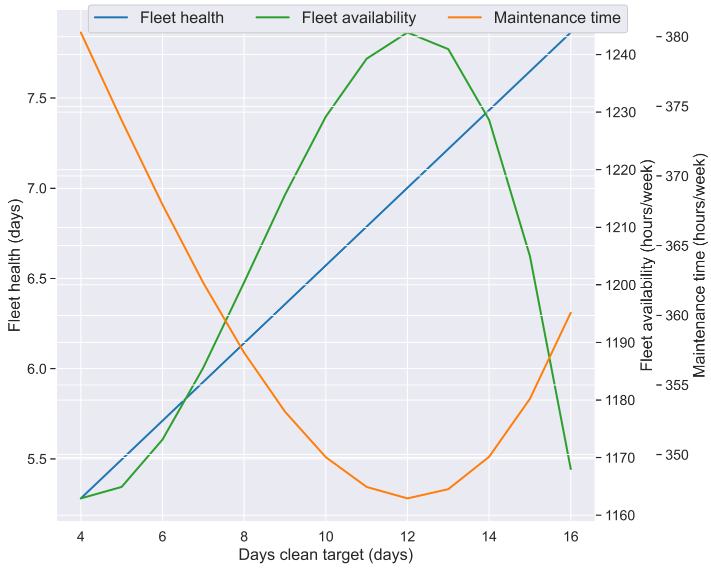

Figure 4.1: Trade-off between fleet health, maintenance time and fleet availability

days-clean targets, with the former performing considerably worse due to high low-health aircraft and low availability. Additionally, while not as pronounced, flexibility decreases at high days-clean targets due to a reduction in fleet availability. Striking the right balance between fleet availability, maintenance time, and fleet health is crucial for achieving maximum flexibility on the day of operations.

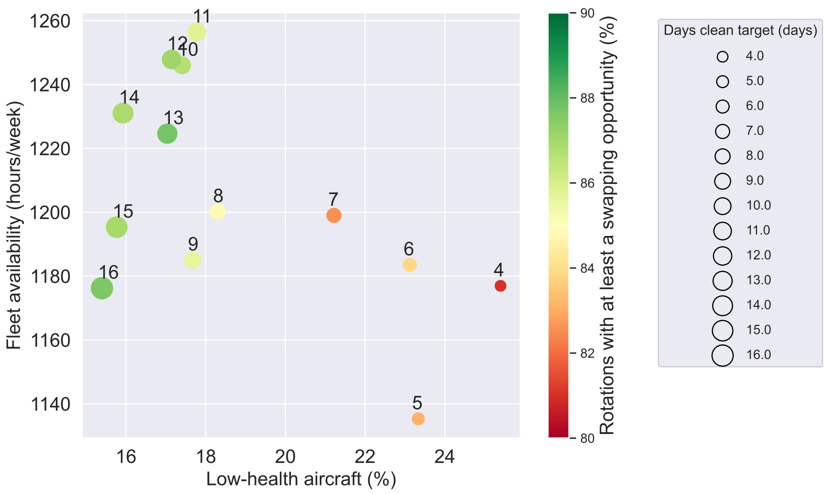

Figure 4.2: Trade-off between low-health aircraft, fleet availability, and schedule flexibility.

4.2. Delay mitigation parameter

The TSN makes use of robust ground arcs to mitigate delay propagation in the schedule. These robust ground arcs are specific to a rotation and have a length that varies with the delay mitigation parameter and is derived from historical airline arrival delay data. While a previous trade-off analysis pointed to a delay mitigation parameter of 95% for maximum arc utilization and presumed optimal schedule stability, this section aims to reassess that decision. By comparing various delay mitigation policies, the objective is to identify the most effective strategy for mitigating delay propagation.

Diverse policies are created by adjusting the delay mitigation parameter. Additionally, to assess the incremental benefits of robust planning, a specific policy analyses the repercussions of a 0% delay mitigation parameter, thus the exclusion of robust ground arcs from the TSN. All policies are assessed in the same time frame between 31/10/2023 and 6/11/2023, and using the same 100 stochastic delay scenarios as in the scientific paper.

Figure 4.3 illustrates how disrupted LOFs and propagated delay change with the delay mitigation parameter. The analysis confirms the validity of selecting 95% as the delay mitigation parameter, as it leads to both low disrupted LOFs and low propagated delay, hence improved schedule stability. Interestingly, a parameter value of 97% performs remarkably similarly but achieves a slightly lower average propagated delay in the subsequent day. Despite this, the assumption in the paper, that maximum robust ground arc utilization would yield the highest schedule resilience, is justified. Moreover, there is an added schedule stability benefit derived from using robust ground arcs, as the 0% delay mitigation policy results in the highest disruptions and propagated delay, respectively 38% and 42% more on the day of operation compared to the 95% policy.

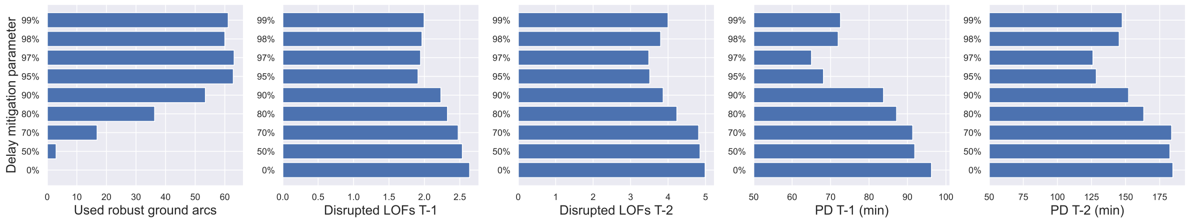

Figure 4.3: Average number of assigned robust ground arcs, and disrupted LOFs and average propagated delay (PD) in the subsequent one (T-1) and two days (T-2) for several delay mitigation parameters.

Table 4.1 presents the distribution of propagated delay for several delay mitigation parameters. Notably, there isn't a single delay mitigation parameter that consistently outperforms others across all delay ranges. Performance can vary significantly, providing flexibility in choosing a delay mitigation parameter based on the specific delay duration to be addressed. For instance, the 90% delay mitigation parameter achieves the lowest delays overall, while the 98% parameter performs best in the 30 to 60-minute delay range. This variation has to do with the length of the robust ground arcs created with a specific delay mitigation parameter being better suited to certain delay ranges. Nonetheless, it can still be said that optimal performance is achieved when robust ground arc utilization is maximized, specifically between 90& and 98% delay mitigation parameter

The variability in the performance of delay mitigation parameters across different delay ranges is also evident in Table 4.2. The table illustrates the diverse distributions of the expected propagated delay (EPD) when disruptions occur associated with different delay mitigation parameters. As expected, high delay mitigation parameters, which on average have longer robust ground arcs, result in a reduction of long delays and lower average EPDs. Overall, the 95% delay mitigation parameter stands out as the best performer, exhibiting the lowest average EPD and no EPD above 30 minutes, reinforcing the rationale behind the selection of this value in the scientific paper.

	> 0	(0,30]	(30,60]	(60,90]	> 90
0%	7.62%	3.61%	2.01%	1.47%	0.53%
50%	6.80%	3.47%	1.60%	1.33%	0.40%
70%	8.29%	3.74%	2.27%	1.74%	0.53%
80%	5.97%	2.60%	1.82%	1.04%	0.52%
90%	4.18%	1.57%	1.44%	0.92%	0.26%
95%	5.43%	2.38%	1.59%	1.19%	0.26%
97%	4.84%	2.22%	1.70%	0.65%	0.26%
98%	4.93%	2.67%	1.33%	0.67%	0.27%
99%	5.86%	3.20%	1.73%	0.53%	0.40%

Table 4.1: Distribution of propagated delay in minutes for varying delay mitigation parameters.

	(0,15]	(15,30]	(30,60]	> 60	avg. EPD
0%	75.3%	15.6%	6.5%	2.6%	12.9 min
50%	80.0%	10.0%	7.1%	2.9%	12.1 min
70%	75.0%	16.7%	6.9%	1.4%	11.9 min
80%	76.4%	14.6%	7.3%	1.8%	12.1 min
90%	73.0%	24.3%	2.7%	0%	11.5 min
95%	86.5%	13.5%	0%	0%	8.4 min
97%	77.3%	15.9%	6.8%	0%	11.5 min
98%	80.4%	13.0%	6.5%	0%	10.6 min
99%	83.3%	10.4%	6.3%	0%	10.5 min

Table 4.2: Distribution of the EPD when disruptions occur in minutes for varying delay mitigation parameters.

4.3. Planning flexibility

Planning with higher flexibility has the potential to result in more efficient and stable plans. Increasing planning flexibility does not mean making unrealistic assumptions, but is at times a realizable possibility for the airline. In this subsection, a planning flexibility policy is evaluated against the standard framework, which is a solid benchmark, as it models the same method currently adopted in practice.

The tested policy assumes that network planners can assign any registration to a rotation, independent of the registration's subtype and the fleet assignment. Performing aircraft subtype swaps (e.g., B787-10 to B787-9) and/or aircraft type swaps (e.g., B787-10 to B777-300ER) may mean the difference between operating or canceling a flight. Moreover, if swaps are done during the tail assignment, the airline has more time to prepare and therefore the impact and costs of the swap are lower. Allowing this flexibility not only has the potential to decrease cancellations and costs but also to improve ground time allocation, thereby improving schedule stability.

In real life swapping aircraft types and subtypes incurs additional costs. Therefore, in the framework, these are also penalized. The airline provided a swapping matrix with swapping costs to be added to $W_{a,f}$ based on landing charges, downgrading costs, and denied boarding costs, however, loss in cargo volume capacity or crew swapping costs are excluded. Therefore, it is assumed that the value of lost cargo capacity is low relative to the value of lost passenger capacity and that all cockpit crews are certified for all aircraft subtypes, which is practically true as the case study uses a whole wide-body Boeing fleet.

The policy resulted in an improvement in ground time allocation, as shown in Table 4.3. Although no additional fleet availability is created, nor additional maintenance time is scheduled, the framework, thanks to this flexibility, manages to allocate ground time more effectively. Critically, rotations followed by at least 2 hours are increased by 2 pp, while also slightly decreasing the number of rotations followed by more than 8 hours of ground time. Thus, swapping flexibility results in a more even distribution of ground time.

Consequently, this policy improves schedule stability. Disrupted LOFs on the day of operations decreased from 1.91 to 1.59, a 17% reduction. Thereby, the propagated delay is also reduced, as visible in Table 4.4. Only 4.23% of subsequent aircraft jobs are delayed due to the late arrival of the previous rotation, as opposed to 5.43% when no swaps are allowed. Moreover, impactful delays longer than 60 minutes are decreased by more than half, from 1.45% of all jobs to just 0.66%. The improvement in the delay distribution is also visible in Table 4.5, where a reduction in average EPD and EPD is greater than 15 minutes are obtained through swapping flexibility. Thus, the possibility to perform type and subtype swaps during the tail assignment enhances schedule stability and efficiency due to greater flexibility in the planning.

	(2, 8]	> 2	> 8
Framework	32.4%	57.9%	24.1%
Swapping flexibility	33.8%	55.9%	24.5%

Table 4.3: Distribution of the ground time planned after rotations in hours.

	> 0	(0,30]	(30,60]	(60,90]	> 90
Framework	5.43%	2.38%	1.59%	1.19%	0.26%
Swapping flexibility	4.23%	1.98%	1.59%	0.53%	0.13%

Table 4.4: Distribution of propagated delay in minutes.

	(0,15]	(15,30]	(30,60]	> 60	avg. EPD
Framework	86.5%	13.5%	0%	0%	8.4 min
Swapping flexibility	90.5%	9.5%	0%	0%	7.5 min

Table 4.5: Distribution of the expected propagated delay (EPD) when disruptions occur in minutes.

Bibliography

[1] F. T. S. Chan S. H. Chung H. L. Ma A. E. E. Eltoukhy, Z. X. Wang and X. P. Wang. Robust aircraft maintenance routing problem using a turn-around time reduction approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(12):4919–4932, 2020. doi: https://doi.org/10.1109/TSMC.2019.2937648.

[2] R. Batta A. Sarac and C. M. Rump. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, 175(3):1850–1869, 2003. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2004.10.033.

[3] J. Abara. Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19(4):20–28, 1989. ISSN 0092–2102. doi: https://doi.org/10.1287/inte.19.4.20.

[4] J. F. Bard B. G. Thengvall and G. Yu. A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. *Transportation Science*, 37(4):392–407, 2003. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.37.4.392.23281.

[5] C. Barnhart. *Chapter 9. Irregular Operations: Schedule Recovery and Robustness.* Wiley, 2009. ISBN 9780470744727.

[6] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9:279–298, 2006. doi: https://doi.org/10.1007/s10951-006-6781-0.

[7] C. A. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi. The fleet assignment problem: 1 solving a large-scale integer program. Technical report, 1995.

[8] L. W. Clarke E. L. Johnson G. L. Nemhauser C. Barnhart, N. L. Boland and R. G. Shenoi. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):208–220, 1998. ISSN 0041–1655. doi: https://doi.org/10.1287/trsc.32.3.208.

[9] T. S. Kniker C. Barnhart and M. Lohatepanont. Itinerary-based airline fleet assignment. *Transportation Science*, 36(2):199–217, 2002. ISSN 0041–1655. doi: https://doi.org/10.1287/trsc.36.2.199.566.

[10] F. Delgado C. Lagos and M. A. Klappa. Dynamic optimization for airline maintenance operations. *Transportation Science*, 54(4):998–1015, 2020. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.2020.0984.

[11] L. Cadarso and Á. Marín. Integrated robust airline schedule development. *Procedia Social and Behavioral Sciences*, 20:1041–1050, 2011. ISSN 1877-0428. doi: https://doi.org/10.1016/j.sbspro.2011.08.113.

[12] Q. Deng and B. F. Santos. Lookahead approximate dynamic programming for stochastic aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 299(3):814–833, 2022. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2021.09.0197.

[13] G. De Maere J. Mulder M. Paelinck E. K. Burke, P. De Causmaecker and G. Van den Berghe. A multi-objective approach for robust airline scheduling. *Computers Operations Research*, 37:822–832, 2010. ISSN 0305–0548. doi: https://doi.org/doi:10.1016/j.cor.2009.03.026.

[14] EUROCONTROL. All-causes delays to air transport in europe quarter 3 2022. Technical report, 2022.

[15] EUROCONTROL. All-causes delays to air transport in europe - quarter 2 2023. Technical report, 2023.

[16] S. J. Maher G. Froyland and C. L. Wu. The recoverable robust tail assignment problem. *Transportation Science*, 48(3):351–372, 2014. ISSN 1526–5447. doi: http://dx.doi.org/10.1287/trsc.2013.0463.

[17] T. Grosche. *Computational Intelligence in Integrated Airline Scheduling*. Springer, 2009. ISBN 978-3-540-89886-3. doi: https://doi.org/10.1007/978-3-540-89887-0.

[18] N. Safaei H. Shahmoradi-Moghadam and S. J. Sadjadi. Robust maintenance scheduling of aircraft fleet: A hybrid simulationoptimization approach. *IEEE Access*, 9:17854–17865, 2021. doi: https://doi.org/10.1109/ACCESS.2021.3053714.

[19] B. F. Santos H. W. M. Vos and T. Omondi. Aircraft schedule recovery problem a dynamic modeling framework for daily operations. *Transportation Research Procedia*, 10:931–940, 2015. ISSN 2352–1465. doi: https://doi.org/10.1016/j.trpro.2015.09.047.

[20] P. J. Van Kessel I. Tseremoglou and B. F. Santos. A comparative study of optimization models for condition-based maintenance scheduling of an aircraft fleet. *Aerospace*, 10(2), 2023. doi: https://doi.org/10.3390/aerospace10020120.

[21] IATA. Airline Maintenance Cost Executive Commentary. Technical report, 2020.

[22] J. Larsen J. Clausen, A. Larsen and N. J. Rezanova. Disruption management in the airline industryconcepts, models and methods. *Computers Operations Research*, 37(5):809–821, 2010. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2009.03.027.

[23] H. Fricke J. Evler, M. Lindner and M. Schultz. Integration of turnaround and aircraft recovery to mitigate delay propagation in airline networks. *Computers and Operations Research*, 138, 2022. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2021.105602.

[24] G. Yu J. F. Bard and M. F. Argüello. Optimizing aircraft routings in response to groundings and delays. *IIE Transactions*, 33(10):931–947, 2001. doi: https://doi.org/10.1023/A:1010987008497.

[25] L. Marla J. Lee and A. Jacquillat. Dynamic disruption management in airline networks under airportoperating uncertainty. *Transportation Science*, 54(4):973–997, 2020. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.2020.0983.

[26] E. L. Johnson J. M. Rosenberger and G. L. Nemhauser. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3):357–368, 2004. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.1030.0038.

[27] J. Van den Bergh, P. De Bruecker, J. Beliën, and J. Peeters. Aircraft maintenance operations: state of the art. Technical report, 2013.

[28] W.J.C. Verhagen I. Medeiros J. Vink, B.F. Santos and R. Filho. Dynamic aircraft recovery problem - an operational decision support framework. *Computers Operations Research*, 117, 2020. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2020.104892.

[29] A. F. Abdelghany K. F. Abdelghany and G. Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185(2):825–848, 2008. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2006.12.045.

[30] H. A. Kinnison and T. Siddiqui. *Aviation Maintenance Management, 2nd Edition*. McGraw-Hill Education, 2 edition, 2013. ISBN 9780071805025.

[31] L. Hassan. Aircraft Disruption Management: Increasing Performance with Machine Learning Predictions. Technical report, 2019.

[32] L. S. Kang. Degradable Airline Scheduling: an Approach to Improve Operational Robustness and Differentiate Service Quality. Technical report, 2004.

[33] E. L. Johnson L. W. Clarke, C. A. Hane and G. L. Nemhauser. Maintenance and crew considerations in fleet assignment. *Transportation Science*, 30(3):249–260, 1996. ISSN 0041–1655. doi: https://doi.org/10.1287/trsc.30.3.249.

[34] M Lapp and A. Cohn. Modifying lines-of-flight in the planning process for improved maintenance robustness. *Computers Operations Research*, 39(9):2051–2062, 2012. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2011.08.024.

[35] B.F. Santos L.K. Hassan and J. Vink. Airline disruption management: A literature review and practical challenges. *Computers and Operations Research*, 127:105–137, 2021. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2020.105137.

[36] David J. Worthington Luke Rhodes-Leader, Bhakti Stephan Onggo and Barry L. Nelson. Airline disruption recovery using symbiotic simulation and multi-fidelity modelling. *Proceedings of the Operational Research Society Simulation Workshop 2018 SW*, pages 146–155, 2018.

[37] M. Haouari M. A. Aloulou and F. Zeghal Mansour. Robust aircraft routing and flight retiming. *Electronic Notes in Discrete Mathematics*, 36:367–374, 2010. ISSN 1571–0653. doi: https://doi.org/10.1016/j.endm.2010.05.047.

[38] M. Haouari M. B. Ahmed, W. Ghroubi and H. D. Sherali. A hybrid optimization-simulation approach for robust weekly aircraft routing and retiming. *Transportation Research Part C: Emerging Technologies*, 84: 1–20, 2017. ISSN 0968–090X. doi: https://doi.org/10.1016/j.trc.2017.07.010.

[39] G. Nemhauser M. Ball, C. Barnhart and A. Odoni. *Chapter 1 Air Transportation: Irregular Operations and Control.* Elsevier, 14 edition, 2007. ISBN 9780444513465. doi: https://doi.org/10.1016/S0927-0507(06)14001-3.

[40] M. Dresner M. Hansen K. Neels A. Odoni E. Peterson L. Sherry A. Trani B. Zou R. Britto D. Fearing P. Swaroop N. Uman V. Vaze M. Ball, C. Barnhart and A. Voltes. Total delay impact study a comprehensive assessment of the costs and impacts of flight delay in the united states. Technical report, 2010.

[41] G. Froyland M. Dunbar and C. L. Wu. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers Operations Research*, 45:68–86, 2014. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2013.12.003.

[42] A. Atamtürk M. S. Aktürk and S. Gürel. Aircraft rescheduling with cruise speed control. *Operations Research*, 62(4):829–845, 2015. ISSN 1526–5463. doi: https://doi.org/10.1287/opre.2014.1279.

[43] Q. Deng M. Witteman and B. F. Santos. A bin packing approach to solve the aircraft maintenance task allocation problem. *European Journal of Operational Research*, 294:365–376, 2021. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2021.01.027.

[44] M. Marseguerra and E. Zio. Optimizing maintenance and repair policies via a combination of genetic algorithms and monte carlo simulation. *Reliability Engineering and System Safety*, 68(1):69–83, 2000. ISSN 0951–8320. doi: https://doi.org/10.1016/S0951-8320(00)00007-7.

[45] M. Salani N. Eggenberg and M. Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Computers Operations Research*, 37(6):1014–1026, 2010. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2009.08.006.

[46] J. Larsen A. Ross N. Kohl, A. Larsen and S. Tiourine. Airline disruption managementperspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162, 2007. ISSN 0969–6997. doi: https://doi.org/10.1016/j.jairtraman.2007.01.001.

[47] V. Xanthakis D. Mourtzis N. Papakostas, P. Papachatzakis and G. Chryssolouris. An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48(4):604–612, 2010. ISSN 0167–9236. doi: https://doi.org/10.1016/j.dss.2009.11.010.

[48] V. Mousseau S. Michel O. Khaled, M. Minoux and X. Ceugniet. A multi-criteria repair/recovery framework for the tail assignment problem in airlines. *Journal of Air Transport Management*, 68:137–151, 2023. ISSN 0969–6997. doi: https://doi.org/10.1016/j.jairtraman.2017.10.002.

[49] Bureau of Transport Statistics. On-time performance - reporting operating carrier flight delays at a glance. https://www.transtats.bts.gov/homedrillchart.asp, 2022.

[50] D. Guimarans P. Arias, M. M. Mota and G. Boosten. A methodology combining optimization and simulation for real applications of the stochastic aircraft recovery problem. *Proceedings 8th EUROSIM Congress on Modelling and Simulation, EUROSIM 2013*, pages 265–270, 2013. doi: https://doi.org/10.1109/EUROSIM.2013.55.

[51] W. J. Verhagen P. Callewaert and R. Curran. Integrating maintenance work progress monitoring into aircraft maintenance planning decision support. *Transportation Research Procedia*, 29:58–69, 2018. doi: https://doi.org/10.1016/j.trpro.2018.02.006.

[52] F. C. Freeman P. J. Van Kessel and B. F. Santos. Airline maintenance task rescheduling in a disruptive environment. *European Journal of Operational Research*, 308(2):605–621, 2023. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2022.11.017.

[53] X. Su J. Liu P. Yuan, W. Han and J. Song. A dynamic scheduling method for carrier aircraft support operation under uncertain conditions based on rolling horizon strategy. *Applied Sciences*, 8, 2018. ISSN 2076–3417. doi: https://doi.org/10.3390/app8091546.

[54] B F. Santos Q Deng and R Curran. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281(2):256–273, 2020. ISSN 0377–2217. doi: https://doi.org/10.1016/j.ejor.2019.08.025.

[55] G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51:155–170, 2007. doi: https://doi.org/10.1002/net.20212.

[56] A. Cohn S. AhmadBeygi and M. Lapp. Decreasing airline delay propagation by reallocating scheduled slack. *IIE Transactions*, 42(7):478–489, 2010. doi: https://doi.org/10.1080/07408170903468605.

[57] G. Laporte S. Bisaillon, J. F. Cordeau and F. Pasin. A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR*, 9:139–157, 2011. ISSN 1614–2411. doi: https://doi.org/10.1007/s10288-010-0145-5.

[58] J. P. Clarke S. Lan and C. Barnhart. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1):15–28, 2006. doi: https://doi.org/10.1287/trsc.1050.0134.

[59] S. P. Ackert . BasicsofAircraftMaintenance ProgramsforFinanciers Evaluation Insights of Commercial Aircraft Maintenance Programs. Technical report, 2010.

[60] C. L. Wu F. Delgado S. Shaukat, M. Katscher and H. Larrain. Aircraft line maintenance scheduling and optimisation. *Journal of Air Transport Management*, 89, 2020. ISSN 0969–6997. doi: https://doi.org/10.1016/j.jairtraman.2020.101914.

[61] B. C. Smith and E. L. Johnson. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516, 2006. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.1060.0153.

[62] C. Sriram and A. Haghani. An optimization model for aircraft maintenance scheduling and reassignment. *Transportation Research Part A: Policy and Practice*, 37(1):29–48, 2003. ISSN 0965–8564. doi: https://doi.org/10.1016/S0965-8564(02)00004-6.

[63] Q. Deng T. Van der Weide and B. F. Santos. Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers Operations Research*, 141, 2022. ISSN 0305–0548. doi: https://doi.org/10.1016/j.cor.2021.105667.

[64] S. Gürel U. Arkan and M. S. Aktürk. Flight network-based approach for integrated airline recovery with cruise speed control. *Transportation Science*, 51(4):1259–1287, 2017. ISSN 1526–5447. doi: https://doi.org/10.1287/trsc.2016.0716.

[65] C. L. Wu. Improving airline network robustness and operational reliability by sequential optimisation algorithms. *Networks and Spatial Economics*, 6:235–251, 2006. ISSN 1572–9427. doi: https://doi.org/10.1007/s11067-006-9282-y.

[66] H. Wang Z. Liang W. A. Chaovalitwongse Y. Su, K. Xie and P. M. Pardalos. Airline disruption management: A review of models and solution methods. *Engineering*, 7(4):435–447, 2021. ISSN 2095–8099. doi: https://doi.org/10.1016/j.eng.2020.08.021.

[67] Y. V. Ageeva. Apprroaches to Incorporating Robustness Into Airline Schedules. Technical report, 2020.

[68] C. Yan and J. Kung. Robust aircraft routing. *Transport Science*, 52(1):118–133, 2016. ISSN 0041–1655. doi: https://doi.org/10.1287/trsc.2015.0657.

[69] S. Yan and D. Yang. A decision support framework for handling schedule perturbations. *Transportation Research Part B: Methodological*, 30:405–419, 1996. ISSN 0191–2615. doi: https://doi.org/10.1016/0191-2615(96)00013-6.

[70] X. Qianb L. Zhou X. Jin X. Lu Z. Liang, F. Xiao and S. Karichery. A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. *Transportation Research Part B*, 113:70–90, 2018. ISSN 0191–2615. doi: https://doi.org/10.1016/j.trb.2018.05.007.

[71] X. Zhang T. Wu Z. Liang, Y. Feng and W. A. Chaovalitwongse. Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B: Methodological*, 78:238–259, 2015. ISSN 0191–2615. doi: https://doi.org/10.1016/j.trb.2015.03.013.

[72] B. Li C. Dang Z. Wu, Q. Gau and F. Hu. A rapid solving method to large airline disruption problems caused by airports closure. *IEEE*, 5:2645–2655, 2017. doi: https://doi.org/10.1109/ACCESS.2017.2773534.

[73] C. Zhang. Two-stage heuristic algorithm for aircraft recovery problem. *Discrete Dynamics in Nature and Society*, 2017, 2017. doi: https://doi.org/10.1155/2017/9575719.

[74] Chuangyin Dang Zhengtian Wu, Benchi Li. Solving multiple fleet airline disruption problems using a distributed-computation approach to integer programming. *IEEE*, 5:19116–19131, 2017. doi: 10.1109/ACCESS.2017.2747155.