Anticipatory Airline Disruption Management

A model-based reinforcement learning approach to anticipatory aircraft recovery under disruption uncertainty

Willem Vos

Anticipatory Airline Disruption Management

A model-based reinforcement learning approach to anticipatory aircraft recovery under disruption uncertainty

Thesis report

by

Willem Vos

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on February 26, 2025 - 13:00

Thesis committee:	
Chair:	Dr. A. Jamshidnejad
Supervisors:	Dr. M.J. Ribeiro
	I. Tseremoglou
	G. Campuzano
	G. Stojkovic
	Dr. F. Schulte
External examiner:	Dr. R. Merino Martinez
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	May, 2024 - February, 2025
Student number:	4692772

Faculty of Aerospace Engineering · Delft University of Technology

Contents

Preface

This document contains the Scientific Paper and Literature Review & Research Definition as both parts of the MSc Thesis at the department Control & Operations, faculty of Aerospace Engineering at Delft University of Technology. I want to thank everyone who was involved in this research and contributed to the conceptualization, design, analysis and interpretation and making this research possible. I want to give a special thanks to Marta Ribeiro for her supervision and continuous guidance, helping me shape the research and steering and challenging me to reach the highest level of research possible. Without such good supervision the research could not have been what it is. I would like to thank Goran Stojkovic and Frederik Schulte for making this research possible and providing useful and interesting insights that helped shaped the research. Lastly I would like to thank Iordanis Tseremoglou, Giovanni Campuzano and Pieter Becking for all the insightful discussions we had that inspired the creative thinking needed to tackle the obstacles along the way.

Willem Vos
Delft, February 2025

1 Introduction

Airlines run tight optimized schedules in order to make profit in a competitive market where margins are thin. Due to the highly optimized nature of the airlines' operations, disruptive events can pose a significant problem every day. Many factors can lead to disruptions. Bad weather conditions, aircraft technical problems, airport congestion or airline delays are some of these causes. For a large portion, these disruption are uncertain in when they will occur and how severe their impact will be in the short and long term. The potential impact of these disruptions is vast however. For example, according to **?**, airline disruptions cost an estimated 60 billion annually worldwide, which is around 8% of worldwide airline revenue. Not only do airlines incur significant extra costs by having to compensate crew, passengers or even airports, they can also experience customer dissatisfaction which can have an influence on future revenue streams.

In disruption management, the recovery of the airlines' resources (i.e. aircraft, crew, passengers) during and after disruptive events is referred to as the Airline Recovery Problem (ARP). The goal of this research is to address the ARP by developing a novel solution method that anticipates potential future disruptions using a model-based Reinforcement Learning (RL) algorithm. Disruption management is currently done by airlines in a reactive manner, where airline operators only act after disruptive events have occurred. This sparks the belief that proactive methods (i.e. methods where there is acted in advance of future disruptions with a high probability of occurring, without knowing if these disruption will actually happen) yield better performance for airline recovery. That is, a proactive approach would result in less flight cancellations and delays then current reactive methods. Reinforcement Learning exploits patterns in data in such a way that future scenarios can be anticipated, paving the way for a shift from reactive disruption management to proactive disruption management for airlines. This MSc Thesis is conducted in collaboration with Boeing Digital Aviation Solutions (DAS).

The structure of this report is as follows: Part I contains the scientific paper of the complete research, Part II contains the Literature Review & Research Definition.

Part I

Scientific Paper

A model-based reinforcement learning approach to anticipatory aircraft recovery under disruption uncertainty

B.F.W. Vos

[a] *Delft University of Technology, Department of Control & Operations, Kluyverweg 1, Delft, 2629 HS, Netherlands*

Abstract

Disruptive events pose a significant challenge to airlines' everyday operations due to the highly optimized nature of their schedules. Unforeseen events force airlines to rapidly reschedule and adjust their operations. Current disruption management methods rely mostly on reactive and static models that fail to capture the dynamic and probabilistic nature of airline recovery. This study presents a model-based reinforcement Reinforcement Learning (RL) method for aircraft recovery under disruption uncertainty that anticipates future potential disruptions. The Aircraft Recovery Problem (ARP) is formulated as a Markov Decision Process (MDP) and a framework is proposed in which an Approximate Dynamic Programming (ADP) algorithm that relies on Value Function Approximation (VFA) determines optimal recovery actions considering the immediate and future impact of each action. The uncertain disruptions are modelled as aircraft unavailabilities with a fixed probability of realizing. The aim of the model is to keep flight delays and cancellations at a minimum while exploiting stochastic information on potential aircraft unavailabilities. The model is tested on multiple scenarios with different objectives and levels of disruptions and is benchmarked against an exact optimization algorithm. Results indicate that a proactive approach outperforms reactive models, particularly in high-disruption scenarios with high aircraft utilization. The comparison with the exact benchmark shows that the RL method can achieve sub-optimal solutions with considerably less corrective actions. This framework offers a decision support tool that allows airline operators to find more resilient solutions in uncertain environments by incorporating probabilistic predictions on disruptions in the decision-making process.

Keywords: Airline Disruption Management, Aircraft Recovery, Reinforcement Learning, Anticipatory Disruption Management

1. Introduction

The aviation industry is highly competitive and airlines run tight optimized schedules sustain profitability from thin margins. Disruptive events pose a constant threat to airlines' operations, and – if not managed properly – have detrimental effects on the airlines' operational costs and revenue. It is estimated that airline disruptions cost an estimated 60 billion annually worldwide, representing around 8% of worldwide airline revenue (Gershkoff, 2016). Not only do airlines incur significant extra costs with irregular operations and extra compensation for crew, passengers or even airports. Customer dissatisfaction potentially may have a negative effect on future revenue streams as well (Zhao et al., 2023).

Bad weather conditions, aircraft technical problems, airport congestion or airline delays are some common causes for disruptions in airline operations. These disruption are mostly uncertain in terms of when and whether they occur and the severity of short- and long term impact on operations. Although many events that trigger disruptions are highly unpredictable in nature, some of them come with a certain quasi-predictability, which can help airlines anticipate potential disruptions. Information on Weather forecasts, Prognostics and Health Monitoring (PHM) for airline equipment, and predictions on airport congestion can help airlines identify potential disturbances and mitigate risks to their operations, and more effectively recover their resources and flight schedules in the face of disruptions.

Current practice for airlines is that they manage disruptions in their schedule reactively (Kohl et al., 2007). Once a disruptions has occurred, operators will assess the impact of the disruption and decide what actions needs to be taken to mitigate negative effects of these disruptions. These actions are referred to as *recovery actions* and usually involve swapping airline resources (aircraft & crew), retiming departures/arrivals or canceling flights (Su et al., 2021). Since a reactive approach limits AOCC operators in making recovery decisions at an earlier stage, it also limits the number of options they have for achieving effective recovery solutions. By leveraging the ability to navigate disruptive events before they occur, airlines can adopt more effective recovery strategies that can help reduce last minute cancellations and potentially result in better short- and long-term recovery solutions in terms of costs and resiliency of their operations (Lee et al., 2020; Zang et al., 2024).

Existing works on disruption management predominantly rely on exact mathematical formulations or heuristics-based optimization frameworks defined in a reactive setting. Although in recent works an increase in number of functionalities that resemble real-world operations incorporated in the proposed frameworks was seen (Hassan et al., 2021), these methods still lack the dynamic and uncertainty related aspects of disruption management. They often lack flexibility to adapt to changing circumstances (Wang et al., 2019) and their reactive and static approaches do not mimic the disruption management process close enough (Hassan, 2019), leading to underestimations of recovery costs (Vos et al., 2015). Additionally, the shift to more extensive recovery framework brings new computational challenges with it, where the requirement of finding solutions within 120 seconds still remains for AOCC operators. These challenges stimulate us to search for different methods that are better able to handle the issues on flexibility, inherent problem characteristics, uncertainties and computation times. Reinforcement Learning (RL) is a field in machine learning that enables adaptive optimization in dynamic environments. Unlike traditional optimization methods, RL models are able to learn recovery strategies through interaction with the system, continuously improving their decisions based on feedback from past actions. This approach allows for greater flexibility in responding to disruptions as they occur, rather than relying on pre-defined rules or static models. Moreover, RL can incorporate stochastic elements of real-world operations, making it better suited for handling uncertainties inherent in disruption management. Given these advantages, RL has the potential to bridge the gap between the challenges currently seen in disruption management research.

This research aims to address Airline Disruption Management (ADM) under uncertain conditions by developing a solution framework for the Aircraft Recovery Problem (ARP) that anticipates potential future disruptions. To achieve this, a model-based Reinforcement Learning framework is proposed in which the ARP is modelled as a sequential decision making process. Considering the dynamic and uncertain properties of ADM – and considering the inherent ability of RL models to capture these dynamic and uncertain problem characteristics – we are motivated to explore proactive aircraft recovery strategies using RL. A flexible proactive recovery framework that relies on (uncertain) information about potential future disruptions is constructed, and its performance in terms of recovery costs as well as solution robustness is evaluated and compared to other recovery strategies. Furthermore, the explainability of the proactive recovery model is assessed to gain additional insights in the models behavior, particularly in how the model decides which actions lead to good future outcomes.

The structure of this paper is as follows: In Section 2, a review of the (published) literature regarding the airline recovery is done and the state-of-the-art is discussed. In Section 3, the problem and its underlying assumptions are formulated, and the modelling framework is presented in detail. Section 4 contains an outline of the methodology and the algorithmic strategy used to solve to problem. Section 5 presents the experimental setup, discussing the choices made regarding the data and the scenarios in which the model is used. In Section 6, the hypothesis are stated and discussed. The results from the model training and the testing experiments are presented in Section 7 and are evaluated further in Section 8. Section 9 discuss the research's limitations and recommendations for future work, and a concluding statement in given in Section 10

2. Literature Review

This section provides a review of publications stemming from literature on ADM. Section 2.1 dissects the literature on ADM, with a focus on aircraft recovery. In Section 2.2, the research gaps identified from the review are stated and the main research question is defined.

2.1. Airline Disruption Management

Airline Disruption Management is a widely studied topic in research with an increase in attention in recent years. In practice, Airline Disruption Management is a sequential process where different resources are recovered (i.e. their planning is adjusted to work around the disruption) in order of importance. Usually, aircraft are recovered first, followed by crew and passengers. Research in ADM can generally be grouped into these distinct categories, although during recent years integrated models – optimizing for all resources simultaneously – have gotten increasing attention. Integrated recovery models are computationally harder, but they allow more optimal solutions (Petersen et al., 2012). Recent advancements in computational power and efficient solution methods has made the use of integrated methods more attractive. However, Airline Operation Control Center (AOCC) operators require solutions within minutes (Vink et al., 2020), which remains the cause for a gap between theoretical models and practical applicability, making ADM problems exceptionally hard. For a more encompassing review on ADM including works on crew and passenger recovery, the reader is referred to Hassan et al. (2021).

Generally speaking, the scope of publications on aircraft recovery covers combinations of having one or more disruptions types (e.g. airport closures (Eggenberg et al., 2010; Wu et al., 2017c; Lee et al., 2020, 2022), aircraft unavailabilities (Hu et al., 2017; Wu et al., 2017a; Zhao et al., 2023), flight delays/cancellations (Vos et al., 2015; Huang et al., 2022)) and one or more recovery options (e.g. tail swaps, delays, flight cancellation, maintenance swaps (Eggenberg et al., 2010; Liang et al., 2018), reserve aircraft (Le and Wu, 2013), ferry flights (Wang et al., 2019)). Many different objectives metrics are identified ranging from straightforward measures such as flight delays and cancellations to more specific measures such as deviation from schedule (Thengvall et al., 2000; Hu et al., 2017). See Table 1 for an overview of some ARP works in terms of their scope, objectives and methods.

The first works to tackle the ARP came from Teodorovic and Guberinic (1984), who developed a method that minimizes total passenger delays by delaying flights or swapping tail-numbers. Teodorovic and Stojkovic (1990) expanded this method by incorporating airport curfews and flight cancellations. In the following decades, more extensive models were proposed that included flexible multi-objective (Thengvall et al., 2000), multi-fleet scenarios (Clarke, 1998; Thengvall et al., 2001, 2003) , and new network representations (Arguello, 1998; Bard et al., 2001). However, the complexity of the problem restricts the practical applicability of theoretical models, especially models focused on integrated recovery of more than one resource. As research on aircraft recovery advanced, more advanced solution methods were developed based on meta- and hybrid-heuristics aimed at solving larger more complex problems in a sufficiently short time. This allowed for more extensive models to be investigated that reflected real-life operations more closely, more with aspects included in the objectives and scope such as maintenance requirements (Hu et al., 2017; Huang et al., 2022), cruise speed control (Marla et al., 2017; Yetimoğlu and Aktürk, 2021), passenger preferences and rebooking options (Maher, 2015; Marla et al., 2017; Cadarso and Vaze, 2022), and even multi-priority flights (Zhong et al., 2024). Models that integrated these operational aspects have better practical applicability, since these models are more in line with real airline operations, and often improved solution qualities of earlier works by including these new aspects in the frameworks.

Given the need to reduce computational processing times, alternative solution methods to exact or heuristics mathematical programs were developed, picking up on the dynamic nature of the problem by solving the problem iteratively as it progresses through time (Vos et al., 2015; Santos et al., 2017; Vink et al., 2020). These methods showed that not incorporating dynamic nature of the problem in frameworks led to significant underestimation of recovery costs as much as 24% to 80% (Vos et al., 2015). Wang et al. (2019) showed that mathematical programs lacked the ability to adapt to real world constraints without increasing the computational complexity disproportionately, by proposing a simulation based approach to the ARP. Methods that relied on the selection of a subset of the problem to incorporate in the solution were also proven useful to limit the computational complexity to a fraction of the whole problem (Vink et al., 2020) while reducing run times by a factor of 10 but retaining the quality of the solution within 3.1% compared to an exact approach, and improving the solution quality when compared to heuristic methods (Rashedi et al., 2024). Machine Learning proved a useful tool in these subset selection frameworks (Hassan, 2019; Eikelenboom, 2022; Rashedi et al., 2024). The interest in the employment of ML in ADM frameworks became more evident as new solution methods were developed based on Reinforcement Learning. Hondet et al. (2018) and Lee et al. (2022) solved the ARP directly via *Q-learning* agents, while, in an extensive recovery framework, Ding et al. (2023) proposed a hybrid approach where a *Deep Reinforcement Learning* (DRL) agent guides a *Variable Neighborhood Search* (VNS) procedure that solved large instances near optimally in a matter of seconds.

Anticipatory frameworks —Although some attention has gone to addressing the dynamic nature of airline recovery, only a handful of works in literature touch the subject of using probabilistic inputs for their recovery frameworks. Lee et al. (2020) addressed the dynamic and uncertain nature of disruptions by proposing an reactive-proactive model that incorporates partial and probabilistic forecasts based on a queuing model to represent airport congestion. They classified delays into three categories: systemic, contingent and propagated. Where systemic delays are due to congestion, contingent delays are due to unforeseen events, and propagated delays are due to downstream effects in the schedule. They found that using these partial predictions yields better results compared to a a myopic baseline, ultimately reducing expected disruption costs without creating additional risk in airline recovery, a promising insight for adaptation of proactive approached to aircraft recovery.

Zang et al. (2024) also included predictions on disruption probabilities, reduced airport capacity in particular, in a recovery framework that dynamically reschedules the disruptions. Based on delay probabilities, they formulate expected costs associated with recovery decisions, which they use in their objective function. Following a case-study with a Chinese airline, results indicate that their method effectively reduces airline delays and operating costs in actual operations.

Other papers do not address anticipatory disruption management directly, but propose methods that can be used in combination with anticipatory models (Zhao et al., 2023; Ogunsina

Table 1: Overview of publications covering Aircraft Recovery.

Paper	Disruptions					Recovery Actions					Characteristics			Objectives	Method
	AU	AC	RAC	DL	CL	SW	DL	CL	MSW	MDL	Maint.	Curfews	Proactive		
Eggenberg et al. (2010)		✓				✓	✓	✓	✓		✓			min. RC	CG
Vos et al. (2015)					✓	✓	✓	✓						min. RC	Aircraft Selection Algorithm
Hondet et al. (2018)				✓		✓								min. DC	RL
Hu et al. (2017)	✓					✓	✓	✓				✓		min. SD, min. max. delay, min # swaps	ϵ-constraint NSH
Wu et al. (2017a)	✓					✓	✓	✓						min. RC	DFPI-IP
Wu et al. (2017b)	✓					✓	✓	✓						min. RC, min. SD	DFPI-IP
Wu et al. (2017c)		✓				✓	✓	✓						min. delay	DFPI-IP
Liang et al. (2018)			✓			✓	✓	✓	✓		✓			min. RC	CG
Hassan (2019)		✓		✓	✓	✓	✓	✓						min. RC	ML-selection
Lee et al. (2020)		✓				✓	✓	✓					✓	min. RC	DP
Lee et al. (2022)		✓				✓	✓							min. delay, min. # delays, min. # delays ¿ 30 minutes	RL
Huang et al. (2022)		✓		✓	✓	✓	✓	✓		✓	✓	✓		min. RC	ICD-CG
Rhodes-Leader et al. (2022)	✓		✓				✓	✓			✓	✓		min. SD, min. DC	Simulation
Zhao et al. (2023)	✓					✓	✓	✓			✓	✓		min. SD, min. RC	RH
Zang et al. (2024)			✓	✓		✓	✓	✓			✓		✓	min. RC	DDB Heuristic
Rashedi et al. (2024)			✓	✓		✓		✓			✓			min. RC	ML subset selection

**AU: Aircraft Unavailability, AC: Airport Closure, RAC: Reduced Airport Capacity, DL: Flight Delays, CL: Flight Cancellations, SW: Aircraft Swap, MSW: Maintenance Swap, MDL: Maintenance Delay, DC: Delay Costs, RC: Recovery Costs, SD: Schedule Deviation, CG: Column Generation, RL: Reinforcement Learning, NSH: Neghbourhood Search Heuristic, DFPI-IP: Distributed Fixed Point Iterative-Integer Program, ML: Machine Learning, DP: Dynamic Programming, ICD-CG: Iterative Cost Driven-Copy Generation algorithm, RH: Rolling Horizon, DDB: Decision-Decomposition-Based Heuristic* et al., 2019, 2021, 2022). Zhao et al. (2023) addressed two uncertainties regarding disruptions: disruption duration, and time that the disruption duration becomes known. Their aim was to develop different recovery strategies as a function of the (unknown) disruption length, such that the recovery scheme could be easily modified with the arrival of new information. Although this is not strictly a proactive approach, a comprehensive scenario analysis indicated which aircraft or flights are likely to be affected when certain disruptions occur. Ogunsina et al. (2019) focused on discovering patterns in historical flight- and disruption data such that it can be leveraged for use in anticipatory mechanisms. In particular, the adoption of Hidden Markov Models (HMM's) for learning temporal patterns from historical data. Ogunsina et al. (2022), extends Ogunsina et al. (2019) by implementing an Uncertainty Transfer Function Model (UTFM) for managing disruptions in airline operations. The UTFM is based on the models described in Ogunsina et al. (2019) and are implemented and assessed with real-world data in this paper. In Ogunsina et al. (2021), they focused on exploratory data analysis for scheduling and operations. They highlight ML methods to analyze the data and reveal important features to predict disruptions. These works highlight the interests in probabilistic models that could be used in conjunction with anticipatory recovery models.

2.2. Research gaps and research question

Following the literature review, several research gaps can be identified. First, although some studies exists that contain *anticipatory frameworks*, no model exists yet that is based on aircraft unavailabilities as disruptions source. Also, no works in ADM (with the exception of (Lee et al., 2020) and Lee et al. (2022)) solve the ARP directly via *learning-based/MDP-defined models*. Also, *dynamic disruption management* has gotten little attention recent ADM works. Vos et al. (2015) addressed the issue by proposing a model that recovers the schedule by building forth on previously found solutions as (new) disruptions happen, but no papers yet have used this dynamic solving based on real-time information on future potential disruptions. Lastly, a significantly overlooked in ADM literature is *robustness* and *explainability* (in case of ML based frameworks). This study aims to tackle this gap by assessing these metrics, looking at how and why solutions are formed and what they mean for airline operators. With these research gaps defined, this research aims to contribute to existing literature by:

(1) Proposing a proactive aircraft recovery model that considers potential future disruptions.
(2) Assessing the robustness of solutions following a proactive strategy.
(3) Assessing the explainability of Machine Learning methods in an ADM context.

This is done by answering the following research question:

> **Can a proactive Reinforcement Learning ARP model lead to more effective and quicker aircraft recovery than existing reactive methods?**

To answer this question, the efficiency and effectiveness of the recovery solutions is evaluated in terms of total delays and cancellation as primary metrics, but also the type and number of recovery actions involved in the solutions plays part in the efficiency of the model. The time to recovery and computation times play an important role in disruption management, and will therefore also be part of the assessment.

3. Problem Statement & Modelling Framework

This section outlines the ARP and its modelling framework as an *Markov Decision Process* (MDP). First, the basic assumptions underlying the model are discussed in Section 3.1, after which a mathematical formulation for the sets and parameters is given in Section 3.2. Section 3.3 provides a definition of the the MDP that lays the foundation for the model used in this paper.

3.1. Assumptions

This paper considers the following assumptions:

(1) Hub & Spoke model with aircraft rotations, each flight represents a rotation in the form of Hub-spoke-hub;
(2) Turnaround times are included in the flight times, thus no extra time is needed between flights;
(3) A homogeneous fleet is used throughout the operations;
(4) No differentiation is made between flights in terms of costs/importance;
(5) Disruption characteristics and probabilities are known.

These assumptions are made to limit the complexity of the model. Although the resulting model does not reflect realistic airline operations, the assumptions allow for a proof of concept of the proactive RL-based method, which will be discussed further in this section.

3.2. Mathematical formulation

This subsection outlines the mathematical formulation for the the MDP. In Tables 2 and 3, the sets and parameters are defined, respectively.

Table 2: Set notations and their definitions.

Symbol	Definition
$\mathcal{F}$	set of flights scheduled for fleet $\mathcal{K}$.
$\mathcal{K}$	set of aircraft in the fleet.
$\mathcal{K}_p$	set of aircraft in the fleet that are prone to unavailabilities.
$\mathcal{F}_k$	set of flights scheduled to be operated with aircraft k.
$\mathcal{F}_{k_p}$	set of flights scheduled to be operated with aircraft k that are prone to disruptions.
$\mathcal{U}_k$	set of unavailabilities that are known for aircraft k.
$\mathcal{F}_k^u$	set of flights that are conflicted by unavailability u for aircraft k.

Note in this ARP, "disruptions" are triggered by aircraft unavailabilities. Therefore, the word unavailability is used interchangeably with disruption throughout this paper.

Table 3: Parameters and their definitions.

Symbol	Definition
$\delta_{fk}^{f'}$	1, if swapping flight f to aircraft k leads to a cancellation of flight f'; 0, otherwise
$\delta_{f}^{f'}$	1, if canceling flight f leads to a cancellation of flight f'; 0, otherwise
$\delta_{0}^{f'}$	1, if doing nothing leads to a cancellation of flight f'; 0, otherwise
δ_{f}^{curf}	1, if flight f violates the curfew; 0, otherwise
c_s	cost of swapping flight f to aircraft k
d_{fk}	minimum time of delays to allow swapping flight f to aircraft k
c_c	cost of canceling a flight
c_d	cost per minute of delays
c_{curf}	cost of violating the curfew
ADT_f	actual Departure Time of flight f
AAT_f	actual Arrival Time of flight f
d_f	duration of flight f
$START_u$	start time of unavailability u
END_u	end time of unavailability u
T_0, T	start time, end time of the recovery window

In this table, recovery costs are split up into different components. Note that the costs $\delta_{fk}^{f'}$, $\delta_{f}^{f'}$ and $\delta_{0}^{f'}$ are all costs of flight cancellations, only following a different action. This can occur as the model can only perform one recovery action per time step, resulting in situations where the model might recover flight f with an certain recovery action, which inadvertently leads to cancellation of flight f', since this flight also needed a recovery action at that time step, something that is not longer possible. Also, the costs c_s, c_c, c_d, c_{curf} are costs that follow directly from individual actions and these costs can vary, depending on the preferences of the airline. The actions will be explained further in Section 3.3.3.

3.3. Markov Decision Process

Since airline disruptions happen dynamically and in uncertain environments, modelling them as a Markov Decision Process can be particularly fitting. In an MDP, a sequential decision process is modelled mathematically, where the outcomes are partially controllable and partially uncertain. Decisions are made sequentially to move from one state to another while obtaining rewards that reflect the quality of the action given the current state and future states. The goal is to derive a policy that maximizes the accumulated rewards. By modelling the ARP as MDP, the inherent dynamics and uncertainties are aimed to be captured. The remainder of this section outlines an MDP framework for the ARP at hand.

3.3.1. States

The state of the system comprises of two parts; The aircraft state vector and the aircraft unavailability state vector:

$$S_t = (K_t, U_t) \tag{1}$$

Where K_t symbolizes the aircraft state vector, and U_t symbolizes the unavailability state vector. For each aircraft k and unavailability u, their state attributes are:

$$k = \begin{pmatrix} \text{Conflicts} \\ \text{Flights} \end{pmatrix} = \begin{pmatrix} \mathcal{F}_k^u \\ \mathcal{F}_k \end{pmatrix}$$

$$u = \begin{pmatrix} \text{Tail \#} \\ \text{Start time} \\ \text{End time} \end{pmatrix} = \begin{pmatrix} k_u \\ START_u \\ END_u \end{pmatrix}$$

Here, conflicts are disrupted flights: the conflict attribute $\mathcal{F}_k^u$ is defined as the set of flights scheduled to be operated with the aircraft k, which cannot depart in the current state of the system due to the unavailability u:

$$END_u \leq ADT_f < START_u \tag{2}$$

Where END_u, ADT_f, and $START_u$ are the end time of unavailability u, departure time of flight f, and start time of unavailability u, respectively. The flights attribute $\mathcal{F}_k$ is the set of all currently scheduled flights to be operated with aircraft k. Note that in this framework, hub - spoke - hub rotations are used for the sake of simplicity, and when a reference to "flights" is done, this is in reality such a rotation. For the remainder of this paper, the "rotations" will therefore be called "flights".

3.3.2. Uncertainties

The uncertain factors in this framework are the unavailability of aircraft. Triggers for such an unavailability can range from various sources, such as unexpected aircraft maintenance or unavailability due to aircraft delays. The unavailabilities are modelled with fixed probabilities that indicate whether an unavailability realizes or not. The start times of the unavailabilities are assumed to be known. Furthermore, all unavailabilities have a fixed duration, meaning it is also assumed that the time when each unavailability ends is known. The potential unavailabilities then realize based on a Bernoulli random variable $X \sim \text{Bernoulli}(p)$. Here p is the individual probability of the unavailability being realized.

3.3.3. Actions

To recover the disrupted schedule, the model must be able to perform recovery actions. In this work, the model is able to perform one action at every time step of the MDP. As mentioned earlier, in this work, three recovery options are considered: tail swaps, flight cancellations, and flight delays. Notably, delays are *not explicitly defined* in the MDP as recovery actions however, but rather follow as a logic consequence of tail swaps. That is, when a flight is assigned to a new tail that has another flight with time overlap with the newly assigned flight, the model delays one of the overlapping flights such that no more flights overlap while keeping the total delay at a minimum (and also considering propagated delays by delaying as many flights as needed in a sequential/recursive manner).

At each time step, the model can either 1) *do nothing*, 2) *swap* a flight to a new tail or 3) *cancel* a flight. If necessary, the swap thus results in one or more flights being delayed to ensure a valid reassignment. See Figure 1 for a visualization of such an action. Here, Flight 1 is disrupted due to unavailability of AC1, and is swapped to AC2. To accommodate this swap, Flight 2 needs to be delayed, which results in the propagation of delay to Flight 3. In this fashion, the model can "swap a flight to the same aircraft" in case a flight delay without reassignment poses a good action e.g. when a flight is scheduled to depart close before the end of an aircraft unavailability. This is a swap action by definition of the MDP, but it appears as a delay in the environment.

For a fleet of $|K|$ aircraft and $|F|$ flights the actions space spans all combinations of tail number-flight assignments for swaps, plus all flights for cancellations, plus 1 for doing nothing. For example, with a fleet of $\mathcal{K}$ aircraft with $\mathcal{F}$ flights, the action space $|\mathcal{X}|$ has a size of $|\mathcal{F}| \times (|\mathcal{K}| + 1) + 1$.

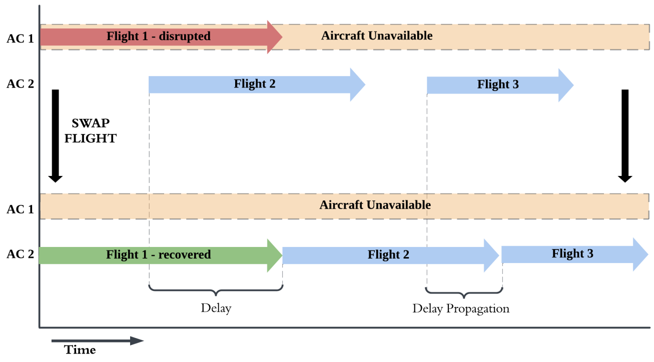


Figure 1: Visualization of a flight re-assignment

3.3.4. *State Transitions*

At each time step t, the transition function for applying decision x to aircraft state vector K is deterministic:

$$K_{t+1} = K^M(K_t, x_t) \tag{3}$$

Where K_t is the aircraft state at t and x_t is the action performed at t. The transition function of the complete state is probabilistic however, where U_{t+1} includes the realization of the potential unavailabilities at $t+1$, drawn from a stochastic process:

$$S_{t+1} = S^M(S_t, x_t, U_{t+1}) \tag{4}$$

The function $S^M(\cdot)$ describe the system dynamics following decisions and new information applied to the state, where the M denotes the "model" for the system dynamics.

3.3.5. *Rewards*

The main goal of airline recovery is to modify the disrupted schedule in such a way that reaches a certain objective (which can be specific to the airline, but usually involves minimizing operational costs, delays, cancellations etc. with the goal of minimizing long term cost and revenue loss). It is not straightforward to quantify all negative effects of disruption into monetary terms, or any other single unit for that matter. With RL however, the need to quantify the disruption impacts into one single metric is requisite, since RL models receive rewards without distinguishing units or types of reward. For this purpose, delays, cancellations, curfew violations and swaps are quantified into one "cost" which is used fed back to the model. In the context of this study, the rewards are negative and are accumulated as the costs of recovery. In a best case scenario, no flights are disrupted and the accumulated reward or costs is 0. For every solution the sum of the rewards is always lower or equal to 0. Throughout this paper, the notation C is used to indicate the negative rewards. The rewards have two components, one where the reward is a direct effect of a decision ($c_{decision}$), and another where a penalty is given when there are still remaining conflicting flights at the end of the recovery window ($c_{not\_recovered}$):

$$C = c_{decision} + c_{not\_recovered} \tag{5}$$

Recovery decisions are defined by the variable $x_{f,k}$ where f and k denote the flight and aircraft it involves, respectively. For a swap action, $x_{f,k}$ swaps flight f to aircraft k. For a cancellation, $x_{f,k}$ cancels flight f and $k = 0$. For a do nothing action, $f = 0$ and $k = 0$. The reward for a decision $x_{f,k}$ is defined mathematically as in (6). The costs parameters c_s, c_c, c_d, c_{curf}, auxiliary variables δ and parameter d_{fk} are defined as in Table 3.

$$c_{x_{f,k}} = \begin{cases} c_c \cdot \left(1 + \sum\limits_{f' \in \mathcal{F}\setminus\{f\}} \delta_f^{f'}\right), & \text{(cancel)} \\ c_s + c_d d_{fk} + c_c \sum\limits_{f' \in \mathcal{F}} \delta_{fk}^{f'} + c_{curf} \sum\limits_{f \in \mathcal{F}} \delta_f^{curf}, & \text{(swap)} \\ c_c \cdot \sum\limits_{f' \in \mathcal{F}} \delta_0^{f'}, & \text{(nothing)} \end{cases} \tag{6}$$

This reward structure can be easily modified to reflect the individual preferences of the airlines. Two types of reward structures are proposed in this work. In the first, *no distinction* is made into the quantification of costs *for the time between the decision and when the decision takes its effect*. That is, a flight that gets canceled 10 minutes prior to departure will result in the same cost as one canceled 8 hours prior to departure. For this reward structure:

$$c_{decision} = c_{x_{f,k}} \tag{7}$$

In the second type of reward structure, the costs are incurred in a similar fashion. The key difference is thus that *costs are higher for decisions made closer in time to when they take effect*, and lower for decisions made farther in time to when they take effect. This structure reflects the undesired effects of last minute changes to the schedule, both for passengers and the airline. To model this, the costs are multiplied with a cost factor between 1 and β based on the relative time left to the departure time of the flight that is changed with action x_f (i.e. swapped/delayed, or canceled) via:

$$c_{decision} = c_{x_{f,k}} \cdot (\beta - \frac{t}{ADT_f - T_0}) \tag{8}$$

Where $c_{x_{f,k}}$ follows from Equation (6), t is the current time, t_0 the start of the recovery window, and ADT_f the departure time of the flight that is changed. Regarding $c_{not\_recovered}$: this remains unchanged, as this a penalty that is only awarded at the absolute end of the recovery horizon.

4. Methodology

An *Approximate Dynamic Programming* (ADP) method is proposed to solve the aircraft recovery problem. ADP lies under the umbrella of RL as it is based on the Bellman optimality equation (explained later in this section). However, the way they are adopted and their vocabularies differ across communities. In Operations Research (OR) problems the focus is on approximation techniques for large scale problems and the ADP adoption is more widely used, while in computer science and artificial intelligence, RL frameworks are the standard that often do not require a model of the environment (Powell, 2011).

Where exact optimization methods lack the ability of handling uncertain and dynamic problem characteristics, ADP is proven well able to handle these characteristics whilst producing quality solutions for industrial scale problems (Powell, 2011). In this section, the ADP framework is outlined in detail. Section 4.1 discusses the main working principles of the algorithm and the way it is used in the context of this ADM problem. Section 4.2 details the way the ADP method generalizes to different problem instances. In Section 4.3, the choices regarding the specific algorithm parameters and architecture are outlined and discussed.

4.1. ADP algorithm with VFA

At the core of all RL frameworks lie the Bellman optimality equations, that states that an optimal policy π for a stochastic optimization problem can be found by solving (where γ^t is a discount factor for the value of $C_t(S_t, X_t^\pi(S_t))$:

$$\max_\pi \mathbb{E}^\pi \left\{ \sum_{t=0}^{T} \gamma^t C_t(S_t, X_t^\pi(S_t)) \right\}.$$

This can be done by recursively solving:

$$V_t(S_t) = \max_{x_t} \left(C_t(S_t, x_t) + \gamma \mathbb{E}[V_{t+1}(S_{t+1}) \mid S_t] \right) \tag{9}$$

Where $C_t(S_t, x_t)$ is the immediate contribution of applying decision x_t to S_t, and $\mathbb{E}[V_{t+1}(S_{t+1}) \mid S_t]$ is the expected downstream value of the state S_{t+1} given that the system is in state S_t (note that the term contribution, costs or (negative) reward are used interchangeably). Unfortunately, solving these optimality equations becomes intractable for virtually every real-life problem as one would face *the curses of dimensionality* (Bellman, 1966). These curses represent the intractability of the problem for growing state spaces, outcome spaces, and action spaces. In this aircraft recovery problem, while the outcome space is fairly constraint (does a disruption happen or not?), the action space grows exponentially with a growing fleet. The state space is near-continuous as it contains the flight times of the schedule. Although this can be discretized to some extent, the state space still remains far too large to visit every state.

Approximate Dynamic Programming (ADP) is a class of solution methods that instead of using backward recursion to solve Equation (9), steps forward in time. This eliminates the need to enumerate all states seen in classical dynamic programming. However, it does not immediately solve the curse of dimensionality. Two problems arise that need a resolution: A mechanism is needed to simulate what *might* happen in the future, and a mechanism is needed to be able to choose the best action at any given t, i.e. having a way to determine the future impact of current decisions (the expected value of the next state $\mathbb{E}[V_{t+1}(S_{t+1}) \mid S_t]$). The first requirement can be achieved via Monte Carlo sampling of stochastic information, reducing the need to calculate all the outcomes that stem from this information (more on this later). The second requirement can be achieved via *Value Function Approximation* (VFA), in which the idea is to iteratively solve the problem over again, and evaluate decisions in each iteration n to improve their quality for the next iteration (Heinold, 2024), and thereby approximating the value function of the states that are visited. Using these approximations for estimating the value of states that are not seen during the iterations, reduces the need to visit every state to learn about the value of that state. This iterative solving of the problem works as follows: In each iteration n, the value estimate of a state $\overline{V}_t^n(S_t^n)$ in updated via exponential smoothing with *stepsize* or *learning rate* α over it's estimate of the previous iteration $\overline{V}_t^{n-1}(S_t^n)$ via:

$$\overline{V}_t^n(S_t^n) = (1 - \alpha_{n-1})\overline{V}_t^{n-1}(S_t^n) + \alpha_{n-1}\hat{v}_t^n, \tag{10}$$

$$\text{in which:} \quad \hat{v}_t^n = \max_{x_t \in \mathcal{X}_t} \left(C_t(S_t^n, x_t) + \overline{V}_t^{n-1}(S^M(S_t^n, x_t)) \right) \tag{11}$$

Since the state contains uncertain information, the step to approximate the expected value of a state directly is difficult. Fortunately, an elegant construct to overcome this exists that relies on the post-decision state. This construct splits the state transition function $S_{t+1} = S^M(S_t, x_t, U_{t+1})$ up into two steps:

$$S_t^x = S^{M,x}(S_t, x_t), \tag{12}$$

$$S_{t+1} = S^M(S_t^x, U_{t+1}). \tag{13}$$

In Equation (12), the post-decision state transition S_t^x is the state right after applying x and is deterministic. In Equation (13), the stochastic information is added to the post-decision state to reach the next pre-decision state S_{t+1}. The value of S_{t+1} is also the value of the decision that leads to S_{t+1} (at least from a decision making perspective). And since S_{t+1} is the result of a random variable. The value of the post-decision state S_t^x represents the value of the decision that leads to S_{t+1}, which could be determined deterministically. The values can now be updated via:

$$\overline{V}_{t-1}^n(S_{t-1}^{x,n}) = (1 - \alpha_{n-1})\overline{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n, \tag{14}$$

$$\text{In which:} \quad \hat{v}_t^n = \max_{x_t \in \mathcal{X}_t} \left(C_t(S_t^n, x_t) + \overline{V}_t^{n-1}(S_t^{x,n}) \right). \tag{15}$$

Using these approximations to estimate the value of states that are not seen during the iterations, reduces the need to visit every state to learn about the value of that state and thus the curses of the state and outcome space are tackled. However, the need to evaluate all possible decisions still exists in this framework. In Algorithm 1, the main algorithm for this methods is outlined.

Algorithm 1 An ADP algorithm based on the post-decision state values and exogenous information

1: Choose an initial approximation $\overline{V}_t^0, \forall t \in T = \{0, 1, \ldots, T\}$.
2: Set the initial state to S_0^1.
3: **for** $n = 1, \ldots, I$ **do**
4: Obtain a sample path u^n.
5: **for** $t = 0, 1, \ldots, T$ **do**
6: $\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t} \left(C_t(S_t^n, x_t) + \overline{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$
7: Let x_t^n be the value of x_t that solves the maximization problem.
8: Update the value function using:

$$\overline{V}_t^{n-1}(S_{t-1}^{x,n}) = (1 - \alpha_{n-1})\overline{V}_t^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n.$$

9: Compute the subsequent post-decision state:

$$S_t^{x,n} = S_x^M(S_t^n, x_t^n)$$

10: Compute the next pre-decision state by adding stochastic information $U_{t+1}(u^n)$:

$$S_{t+1}^n = S^M(S_t^{x,n}, U_{t+1}(u^n)).$$

11: Increment t. If $t < T$: go to step 6
12: **end for**
13: Increment n. If $n < N$: go to step 4
14: **end for**
15: **return** the value function: $\{\overline{V}_t^N(S_t^N) \mid \forall t \in T\}$.

4.2. Generalization across the state space

Since the state space is theoretically continuous and therefore practically infinitely large, enumerating all states is not possible. The value function is only learned at the hand of a relatively low number of states, compared to the full state space. A method is required to estimate the value of states that are never visited before, i.e. approximate the full value function based on some samples of the value function learned during the iterations. Operators in AOCC's require fast solutions when faced with flight disruptions. To ensure practical usability, the proposed model should be able to produce solution to unseen problems fast, without having time to repeatedly solve the problem and learn a good solution through value iteration. In Section 4.2.1, the method to approximate the value function from previous experience is detailed. In Section 4.2.2, the exact way this is done is discussed and the choices regarding the function approximation method used in this work are explained.

4.2.1. Function Approximation

In general, value functions can be approximated via aggregation structures, parametric models and non-parametric models. Where aggregations structures rely on combinations of value estimates of states in different levels of state representations (each level has a lower/higher level of detail in the state), to lower the state space size and define the value of this smaller state space. This averts the requirement of having to exploit a special structure in the problem. On the other hand, a disadvantage at the same time is that one can not take advantage of a special structure in the state if it were present. This advantage can be exploited with the use of parametric and non-parametric models, where the state is mapped to a set of *features* $f \in \mathcal{F}$ using a *basis function* $\phi_f(S)$. In a regression form, the value of state S is then estimated by:

$$\overline{V}(S|\theta) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(S) \tag{16}$$

Where the model is linear, but the basis functions can have non-linear characteristics. Finding a basis function might be easy in case the value function is assumed to be linear in, for example, the aircraft state, and contributions to the value of a state come from individual aircraft. In the problem presented in this paper, it is assumed that the value of a state is captured by more complex relations. The difficulty is then to formulate the basis functions $\phi_f(S)$ in such a way that captures the relationships between features to allow the linear model to produce good enough approximations. This is not straightforward, however. A different approach would be to use a non-parametric model that is able to capture complex relationships between features and their contribution to the values. In this model, the features are designed to explain the value of a state as good as possible. However, the model should still be able to approximate values of states without the relationships between features being specified by the basis functions. In this work, the value functions will therefore be approximated using a *non-parametric* model.

4.2.2. Feature Generation

In order to obtain good function approximations using a non-parametric model, it is desirable to define the state in terms of features that explain the value of the state sufficiently well. Intrinsic understanding of the characteristics of the problem is needed (Van Heeswijk et al., 2015). However, finding the features that explain the value of a state is not straightforward. In fact, one of the motivations for using RL/ADP frameworks in this aircraft recovery problem is that it could allow operators to find solutions that may not seem satisfactory but represent a good or near-optimal "value". Either way, a set of features is still needed that captures the state of the system in such a way that the non-parametric model can approximate values of unseen states. To succeed in this, at least a rough idea of some features that may present a good value. In some works in literature (Beirigo et al., 2022), (Simão et al., 2009), this is done by representing the value of a state as the sum of the marginal value of the single resources that make up the state space. Here, for example, an estimate of the marginal value of a single resource can be the dual variable in a Mixed Integer Programming model (MIP) associated with that resource. In this work, no linear relationship between the actions of single resources (aircraft) and the value of a state is assumed, but other features that can provide an estimate of the value of the state can potentially be identified by looking into the problem closely.

It becomes evident that the ease with which flights can be recovered depends on the properties of the best candidate aircraft for tail swaps. If the best candidate aircraft has little to no overlapping flight assignments with the disrupted flights, little to no delays are necessary to recover these flights. If there is substantial overlap, the number of subsequent flights of the candidate aircraft can be an indication of how much this delay might be propagated. Using this logic, a measure to estimate the cost of recovery (read: *value*) of a state can be derived.

To derive such a measure, one would have to account for how easily *all* disrupted flights can be recovered. To achieve this, the total overlap of all (potentially) disrupted flights of an aircraft with the best candidate replacement aircraft is calculated. In this scenario, all disrupted flights from one aircraft are thus swapped to other aircraft. To add to this, the number of subsequent flights is multiplied by the overlap to define an upper bound for delay propagation. In case of multiple unavailable aircraft, the best combinations of overlap multiplied with subsequent flights of candidates are calculated to arrive at an estimate of the ease of recovery for that state.

The following equations describe the calculations of the overlap of (potentially) disrupted flights with the flights of other candidate aircraft in the fleet. Equations (17) and (18) calculate the start and end of an overlap o between flights f and f'. Equation (19) calculates the length of the flight overlap $o_f^{f'}$. By summing $o_f^{f'}$ over all flights of prone aircraft k_p and k, respectively, the total flight overlap between the two aircraft is determined. In case the candidate aircraft k_p is the prone aircraft itself, the overlap is determined as the longest time between a (potentially) disrupted flight departure time and the end of the unavailability as in equation (21).

$$o_{f,start}^{f'} = \max\left(ADT_f, ADT_{f'}\right) \tag{17}$$

$$o_{f,end}^{f'} = \max\left(AAT_f, AAT_{f'}\right) \tag{18}$$

$$o_f^{f'} = \begin{cases} \max\left(o_{f,end}^{f'} - o_{f,start}^{f'}\right), & \text{if } o_{f,start}^{f'} < o_{f,end}^{f'} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$o_{k_p,k} = \sum_{f \in \mathcal{F}_{k_p}} \sum_{f' \in \mathcal{F}_k} o_f^{f'} \tag{20}$$

$$o_{k_p,k_p} = \begin{cases} \min\limits_{f \in \mathcal{F}_{k_p}} \left(END_u - ADT_f\right) \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Now we have a small optimization problem where we want to minimize the total overlap in the state o_S for the combinations of candidate aircraft with binary decision variable $x_{k_p,k} \in \{0,1\}$ indicating 1 if an candidate aircraft k has been found to replace the flights of prone aircraft k_p, 0 otherwise:

$$\min(o_S) = \min \sum_{k_p \in \mathcal{K}_p} \sum_{k \in \mathcal{K}} o_{k_p,k} x_{k_p,k}, \tag{22}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} x_{k_p,k} = 1, \quad \forall k_p \in \mathcal{K}_p, \tag{23}$$

$$\sum_{k_p \in \mathcal{K}_p} x_{k_p,k} \leq 1, \quad \forall k \in \mathcal{K}, \tag{24}$$

$$x_{k_p,k} \in \{0,1\}, \quad \forall k_p \in \mathcal{K}_p, \forall k \in \mathcal{K}. \tag{25}$$

This metric $\min(o_S)$ is an important one that will be present in many of the state features, and it will play a role in the design of the VFA algorithm as well, as will be explained in Section 4.3. In a similar fashion, the number of subsequent flights for candidate aircraft k since the first disrupted flight of k_p plays a role, and is defined by:

$$n_{k_p,k} = |\mathcal{F}_k^{k_p}| \tag{26}$$

Where $\mathcal{F}_k^{k_p}$ is the set of flights $f \in \mathcal{F}_k$ for which $AAT_f \geq ADT_{f',disruped}$ when $ADT_{f,disruped}$ is the first disrupted flight of k_p. The combination of the number of subsequent flights for each potentially unavailable aircraft is then given by:

$$\min(n_S) = \min \sum_{k_p \in \mathcal{K}_p} \sum_{k \in \mathcal{K}} n_{k_p,k} x_{k_p,k} \tag{27}$$

Now, a new interaction metric of finding the minimum of $n_{k_p,k}$ and $o_{k_p,k}$ assignments can be found by minimizing equation (28) under the same constraints and will be denoted by *int_1*:

$$int\_1 = \min \sum_{k_p \in \mathcal{K}_p} \sum_{k \in \mathcal{K}} (n_{k_p,k} \cdot o_{k_p,k}) x_{k_p,k} \tag{28}$$

To include the stochastic nature of the disruptions into these metrics, these metrics are also calculated with weights representing the probabilities of disruptions. Note that the goal of these metrics is not to calculate the exact value of being a state, but rather provide a lower bound for the expected recovery costs.

Looking at the state representation, more features can be identified that might have something to do with the value of the state. The aim is to map the systems' state to as many useful states as possible, as we can later remove the ones that are useless by means of feature selection for our model. In Table 4, the features are listed with a brief description of how they are derived. To summarize and clarify their definitions: t is the current time step in that state, *p_unavail_1, p_unavail_2, p_all, p_none* are the probabilities that the the first, second, all, and none aircraft become unavailable. *min_overlap* is the minimum overlap of disrupted flights with candidate aircraft flights, *min_n_flights* is the minimum number of subsequent flights for the candidate aircraft. *min_util* is the minimum utilization from all candidate aircraft. *int_1, int_2, int_3, int_4* are combinatorial metrics that combine the minimum overlap and subsequent flights from candidate aircraft, some in combination with the expected number of conflicts or probabilities of single aircraft unavailabilities. *n_potential_conflicts* is the number of flight at risk of being disrupted by an unavailability, *exp_n_conflicts* is the expected number of flights that will be disrupted given the probabilities of the disruptions. *total_remaining_conflicts* is the number of future disrupted flights that are already known with certainty, and *recovered* is a binary for whether the schedule is recovered or not. Note that the set of features are mostly features that are unique to the whole systems' state, and not features for individual aircraft or flights. This allows the model to generalize the value approximations to problems with different characteristics in terms of number of flights and fleet size.

Table 4: State features identified as predictors of the state value.

Feature	Definition
t	current time step
p_unavail_1	$p_{u,1}$
p_unavail_2	$p_{u,2}$
⋮	⋮
p_unavail_K	$p_{u,K}$
p_all	$\Pi_k^{\mathcal{K}_p} p_{u,k}$
p_none	$\Pi_k^{\mathcal{K}_p} (1 - p_{u,k})$
min_overlap	$\min(o_S)$, (refer to Equation (22))
min_n_flights	$\min(n_S)$, (refer to Equation(27))
min_util	minimum utilization from all candidate aircraft
int_1	interaction, $\min(o_{k_p,k} * n_{k_p,k})$ (refer to Equation (28))
int_2	interaction, $\min(o_{k_p,k} \cdot E[n\_conflicts]_{k_p})$
int_3	interaction, $\min(o_{k_p,k} * n_{k_p,k} * p_{k_u})$
int_4	interaction, $\min(o_{k_p,k} * n_{k_p,k} \cdot E[n\_conflicts]_{k_p})$
n_potential_conflicts	# of potential conflicts
exp_n_conflicts	expected number of conflicts.
n_disruptions_occurred	# of unavailabilities that have realized.
total_remaining_conflicts	# remaining future conflicts.
recovered	binary if schedule recovered or not.

4.3. Algorithmic choices

This subsection discusses the algorithmic choices for training the algorithm and testing the model to new instances. The choices in parameter values of the ADP algorithm and rewards are outlined in Section 4.3.1 and the choices for the testing architecture and function approximation method are discussed in Section 4.3.2. Additionally, an overview of the training and testing architecture is given in Figure 2.

4.3.1. Training

Parameters —Determining the optimal parameters for an ADP algorithm is not trivial and depends largely on the problem characteristics. One particularly important parameter in value iteration frameworks is the *stepsize*. The stepsize (or learning rate) can have a great impact on the convergence behavior. Since the estimates of values in Algorithm 1 are observed samples from an expectation, the values are updated via a smoothing formula (Equation (14)) to account for the noise in the estimates. However, updating with a small stepsize smooths the learning curve too much and leads to slowed down convergence. In this trade-off, there is no silver bullet for determining the best stepsize strategy, as its effectiveness is highly dependent on the specific problem. There are some rules of thumb to derive a good stepsize rule for a given problem, however. One good strategy to strike balance between the elimination of noise in the value estimates and the rate to convergence, is a harmonic stepsize rule:

$$\alpha = \frac{a}{a + n - 1} \tag{29}$$

Here, a is predefined constant that is derived from observing the convergence behavior with trial and error, and n is the current iteration.

Similarly, the discount factor γ was determined at 1, since discounting the values of future states was observed to slow down the convergence without improving the value to which the states converged. Regarding the rewards, there is no consensus on how to quantify disruption effects into one costs metric. Therefore, sensitivity to changes in the reward structure should be investigated, and airlines should implement individual preferences to optimize for different rewards with care. In this work, costs of delays, swaps, and cancellation are quantified into one unit of costs, where a minute of delay is equal to one unit of costs (c_d). A cancellation is equal to 300 units of costs (c_c) – meaning that airlines prefer to cancel a flight when they expect delays of over 300 minutes – and small costs for performance tail swaps of 5 units of costs (c_s) is implemented to reflect unwanted effects of too many (unnecessary) swaps.

Exploration vs. Exploitation —A common challenge in all RL frameworks is the Exploration-Exploitation trade-off. Particularly for ADP, the initial values significantly influence the convergence behavior. Poor initial values steer the model into the direction of the states it has visited before (of which it now has more optimistic estimates than the poor initial value). Disproportionately optimistic values on the other hand, would encourage full exploration since the visited states are likely believed to be worse than unvisited states with optimistic estimates. Considering the fact that the aim of this work is to tackle the dynamics and uncertainties in disruption management —rather than finding the optimal solution under all conditions —the intent is to find a balance between exploration and exploitation that yields sufficiently good solutions and allows for reasonable convergence rates simultaneously. For this purpose, the *initial values are designed to slightly overestimate the true value*. This slight overestimation leads to a certain degree of exploration without having detrimental effects on the rate of convergence. The formula for the initial values is given by Equation (30) and follows the same principles used during feature generation:

$$\overline{V}_0(S_t) = \min \sum_{k_p \in \mathcal{K}_p} \sum_{k \in \mathcal{K}} (n_{k_p,k} \cdot o_{k_p,k} \cdot p_{k_u}) x_{k_p,k}, \tag{30}$$

S.t. to constraints (23), (24), (25). In Equation 30, the variables $o_{k_p,k}$ and $n_{k_p,k}$ are defined as in Equation 28. p_{k_u} is the probability of unavailability u of aircraft k happening.
For training, M generated flight schedules are used as input to the model. For each schedule m, the samples of value function is learned throughout the iterations. At the last iteration $n = N$, the state features are stored with their corresponding values. In

Table 5: Parameters of the ADP algorithm.

a	γ	c_s	c_c	c_d	c_{curf}	$c_{\text{not\_recovered}}$	β_2	β_3
200	1	5	300	1	150	10.000	1.5	2

total training will be performed on 200 schedules per scenario.

4.3.2. Testing

The proactive recovery model is tested to new problem instances generated from the same process as the training data to verify the performance. By means of a Random Forest Regression model, the value of unseen states can be estimated based on the features that the state maps to, and the future values of states can be evaluated in the decision making process.

Model selection —Under the assumption that the value function is non-linear in the aircraft state space and feature space, a Random Forest Regression (RFR) is chosen to approximate value functions during testing. RFR is an ensemble method that makes use of decision trees to combine predictions from different (bootstrapped) subsets of the data into one. A single tree consists of splits based on the features of the subset used of the data for that tree, where each split minimizes the prediction error. Each tree then predicts the value of new data point based on the splits it constructed during training. For a regression task, the prediction of al trees are then averaged to derive the final prediction. Due to the random nature of the construction of the decision trees and the averaging of multiple predictions, this method is well able to capture interaction effects and non-linearity, while having the ability to avoid overfitting (Fife and D'Onofrio, 2022). Also, RFR has a low computational cost, high predictive accuracy, flexibility, and interpretability (He et al., 2018), without requiring extensive data pre-processing.

Model optimization —To enhance the performance of Random Forest, the hyperparameters can be tuned. Random Forests' performance is dependent on a set of parameters that dictate to what extend certain parts of the algorithm are executed. A Random Forest has 6 different parameters, and finding the best combination for the given training data can be computationally hard. Seeing that an exhaustive search is too time consuming for the benefit it brings, we make use of *Bayesian optimization*, which aims to optimize the parameter by building a surrogate model for the evaluation metric using prior knowledge of the Gaussian Process (GP), and refines this until an optimum is reached. This method is widely used and shown to be effective in terms of accuracy and computational efficiency (Stuke et al., 2021). The optimized values for the hyperparameters based on the training data are shown in Table E.14 in Appendix E.

Model evaluation —performance of the RFR is evaluated by means of the *relative Mean Absolute Error* ($rMAE$) and the *coefficient of determination* (R^2), which gives insights in the average error in the predictions relative to the mean of the target variable (the value of states) and how well the model fits the data, respectively. To avoid overfitting, the robustness of these metrics is compared via *k-fold cross validation*.

Feature importance —Also, to gain insight in the explainability of the model and the way the Random Forest constructs predictions, an *SHapley Additive exPlanations* (SHAP) analysis is done that explains the Random Forest outputs. This method composes a model $f(x)$ into a additive model (Janssen et al., 2022):

$$f(\mathbf{x}) = \phi_0 + \sum_{i=1}^{M} \phi_{x_i}$$

Where ϕ_{x_i} represents the direct effect of of feature i on the prediction. To approximate the value of S_t^x, a function $\Phi(\cdot)$ maps the state to a set of features. These features are passed to the RF model optimized with parameters Θ to produce value prediction $\hat{y}$:

$$\hat{y} = \text{RF}(\Phi(S_t^x) \mid \Theta) \tag{31}$$

These predictions are then used to evaluate the downstream value of decisions at every time step. The uncertain arrival of disruption information is again drawn from the process described in Section 5.1. The algorithm is displayed in Algorithm 2. In Figure 2, the complete testing and training architecture is displayed schematically. The figure shows that the process is divided into a training and testing phase, where the value function is learned during training on many different flight schedules. From the value function, new unseen disruption scenario's can be solved by via approximations based on the learned value function, which comprises the testing phase.

Algorithm 2 Testing algorithm

1: Set the initial state to S_0.
2: Obtain a sample ω.
3: **for** $t = 0, 1, \ldots, T$ **do**
4: $\quad\hat{v}_t = \max_{x_t \in X_t} \left(C_t(S_t, x_t) + \overline{V}_t(S^{M,x}(S_t, x_t)) \right)$
5: $\quad$Let x_t be the value of x that solves 4, where: $\overline{V}_t(S^{M,x}(S_t, x_t)) = \text{RF}(\Phi(S_t^x) \mid \Theta)$
6: $\quad$Compute the subsequent post-decision state:
$$S_t^x = S_x^M(S_t, x_t)$$
7: $\quad$Compute the next pre-decision state by adding stochastic information $W_{t+1}(\omega^n)$:
$$S_{t+1} = S^M(S_t^x, W_{t+1}(\omega)).$$
8: $\quad$Increment t. If $t < T$: go to step 4
9: **end for**
10: **return** Sequence of decisions, Objective value

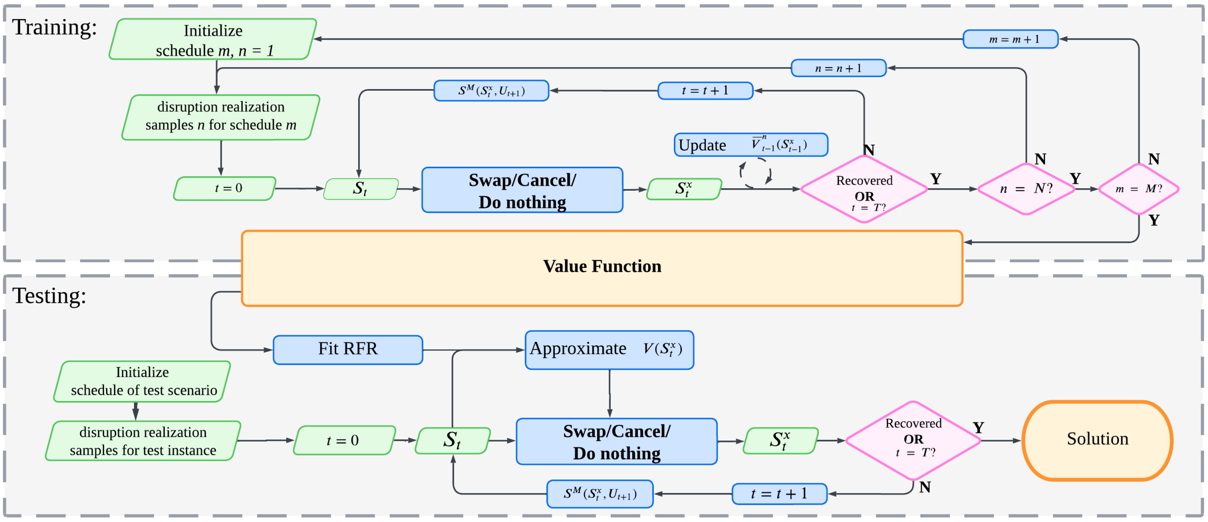


Figure 2: Schematic depiction of training and testing architecture

5. Experimental setup

With the modelling framework in place and the solution method clear, an experimental setup is designed with the goal of providing insights in the performance, behavior and robustness of proactive recovery strategies in comparison to conventional/baseline strategies. This section contains the outline of this experimental setup, explaining the data used for training and testing (Section 5.1) and the configurations in which the ADP framework is tested (i.e. the different policies and scenarios that will be compared (Section 5.2)).

5.1. Data

Data is synthesized based on the formats of the ROADEF Challenge 2009 [1]. Test and training instances are generated via a identical synthesis process using the same parameters such as recovery window times, fleet size, total number of flights in the fleet. For each instance, flights are first randomly assigned to an aircraft. Then, the flight duration and departure time is sampled from uniform distributions $d_f \sim U(60, 180)$ and $ADT_f \sim U(t_{available}, T_{end})$ respectively. Here, $t_{available}$ denotes the earliest available time this flight can depart based on previously assigned flights. Flight durations, departure times and arrival times are then rounded to the nearest quarter hour to discretize the state space to some extend. The maximum number of flights per aircraft is limited to 6. The recovery window starts at 08:00 AM and ends at 18:00 PM. As discussed in Section 3, the potential disruptions realize based on $X \sim \text{Bernoulli}(p)$. For every train and test instance, the potential disruptions are generated with a start time that is sampled from $START_u \sim U(T_0 + 1, T - 2)$ and have a fixed duration of 6 hours. The rate at which unavailabilities realize is based on their probability of occurring, which is again uniformly distributed $p_u \sim U(0.5, 0.9)$. The characteristics of the flight schedule, flights and unavailabilities are displayed in Table 6.

Table 6: Characteristics of synthesized flight schedules

	min.	max.	mean
Flight length (min.)	60	180	-
Unavailability length (min.)	360	360	360
# Flights	0	6	4

5.2. Scenarios and policies

To gain an understanding of the performance of a proactive recovery policy in comparison to conventional policies, and its behavior when subject to changes in the environment, different scenarios are sketched on which the proactive policy is tested. The performance and behavior of the proactive model in these scenarios is compared to different policies. Six configurations of the problem are designed in which two factors are varied:

(1) The number of potential unavailabilities.
(2) The structure of the accumulated rewards. (Based on a penalty factor for last-minute decisions)

The number of potential unavailabilities will vary from one potential disrupted aircraft to two potential disrupted aircraft. The reward structures vary as discussed in Section 3.3.5, with each a different β value that determines the penalty factor that last minute actions get. (Table 8). For each configuration, the results are compared to two benchmark policies. The first is a *reactive policy*. This policy reflects the current practice in AOCC's where only reactive measures are taken after disruptions have occurred. This policy tries to recover the schedule in the same manner as a proactive policy, but has no information on potential disruptions and only "sees" disruptions that have happened. The second policy is a *myopic policy*, this policy reflects a lower bound of the quality of the solution and acts just like a reactive policy, but has no information about the value of future states and is therefore optimizing only for immediate rewards. By combining the different configurations and policies, a set of scenarios will be compared, which are denoted

[1] https://roadef.org/challenge/2009/en/instances.php)

as in Table 7. In the table, the names for these different scenarios are displayed in matrix form where each combination of reward structure, policy, and disruption scenario is named. Furthermore, the scaling characteristics of the models are assessed by testing on larger problem instances. To add to this, the results of the reactive policy is compared to an exact solution, to provide insights in the quality of the VFA method by assessing the optimality gap between the exact solution and RL solution. This exact MILP if formulated in Appendix B.

Table 7: Different policies across rewards and unavailabilities.

Unavailabilities	Single			Multiple		
Reward Structure	1	2	3	1	2	3
Proactive	P_S^1	P_S^2	P_S^3	P_M^1	P_M^2	P_M^3
Reactive	R_S^1	R_S^2	R_S^3	R_M^1	R_M^2	R_M^3
Myopic	M_S^1	M_S^2	M_S^3	M_M^1	M_M^2	M_M^3

Table 8: Parameters for different reward structures – a higher β means a higher penalty factor for last minute decisions, as defined in the reward function in Equation 8.

Reward structure	**1**	**2**	**3**
β (eq. (8))	–	1.5	2

For every configuration of policy and reward structure, 1000 test instances are solved, this could be more but due to the wide range of scenarios and the expectation that 1000 test instances is enough to expose the differences between the scenarios and models. Each test instance is solved once for every possible outcome of the uncertainties (so for two potential disruptions, the instance is solved for a scenario where no disruptions realize, both disruptions realize, and one of the two disruptions realize). The obtained results for each outcome are then weighted with the probability of that outcome.

6. Hypotheses

With the methodology described in this Section, the aim is to gain insights in properties of the proactive approach with reinforcement learning. Several hypotheses have been outlined to support the analysis:

(1) *A proactive approach leads to lower recovery costs than a reactive recovery approach.*
(2) *The advantage of a proactive approach over a reactive approach in terms of objective metrics increases as more aircraft are subject to potential disruptions.*
(3) *The advantage of a proactive approach decreases as the slack of the schedule increases*
(4) *A proactive approach allows for quicker recovery than a reactive one.*

Hypothesis *(1)* is the main hypothesis of this work and is meant to check if a proactive recovery solution can outperform a reactive approach when faced with potential future disruptions. Hypothesis *(2)* focuses more specifically on the behavior of proactive models in when used in schedules with different levels of disruptions; when the schedule is more disrupted, performing proactive actions could provide more options for a better recovery solution and should thus yield a better advantage when compared to a reactive approach. Similarly, hypothesis *(3)* checks if this advantage is correlated with the schedule slack beforehand. A high slack and low utilization of the fleet is expected to have a negative effect on the advantage of proactive models, as a reactive approach might have equally good recovery options in a low slack schedule as a proactive approach. Hypothesis *(4)* is there to validate whether proactive recovery leads to quicker recovery than a reactive one, meaning that a solution is reached in fewer discrete time steps with the proactive model than with the reactive one. These Hypothesis are validated in Section 8.
Furthermore, additional hypotheses can be made regarding the computational scalability of the model, different reward structures, and the performance of an RL model compared to the exact solution: since in this work the curse of the action space must still be handled, it is expected that the proposed ADP algorithm does not scale well with a growing fleet. This also holds for the exact model, but it is expected that the RL model scales worse than the exact model since this relies on a exhaustive search of the action space at every time step. Regarding the different reward structures, it can be expected that a proactive model gets a better advantage over the reactive model when rewards are subject to last-minute penalties, which would follow logically from the hypothesis that a proactive model reaches solutions quicker (which implies it makes decisions at an earlier stage). The convergence of the reward structures that include last-minute penalties is expected to be slower than for rewards without time penalties, since the state value updates for "bad" last minute actions have to propagate backwards through the time steps in order for the model to know, at an early stage, that making a decision at a later stage results in a bad reward.

Another interesting effect of the different reward structures would be how the delays and cancellations are effected. The total reward gets a penalty factor for last-minute decisions, which may lead to the model choosing actions that result in more delays, but less negative reward since these decisions are made well ahead in time. This raises another question for airlines, in whether they prefer less last minute changes with more delays, or less delays but more last minute changes.

Additionally, the reactive RL model is expected to perform suboptimal compared to an exact solution. Although expected, the optimality gap should not be too large, and the RL model must be able to find sufficiently good (for example a gap of less then 25%) solutions in most of the test scenarios.

7. Results

This section contains the computational results following the experiments, showcasing the results for the training algorithm, function approximation, and testing results. The models performance during testing is assessed by comparing their obtained rewards and the resulting delays and number of cancellations. The results of the reactive RL model are benchmarked against an exact (optimal) solution. Moreover, the actions of the proactive model are discussed based on an example solution. Additionally, the robustness of the models is assessed and discussed, and the actions of the RL model are explained.

7.1. Convergence

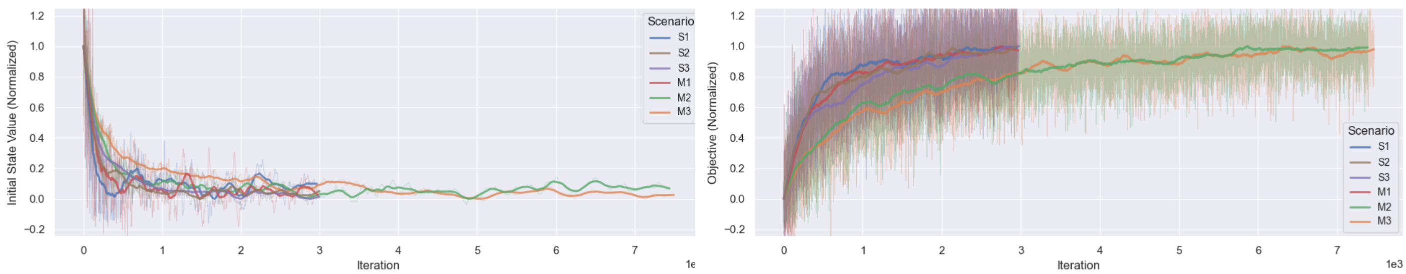
Although there exist stopping criteria rules, they are not always reliable and implementable when dealing with stochastic information (Powell, 2011). With computational power being a limiting factor on the complete training set and having a reliable stopping rule not being a guarantee, trial and error with a limited amount of training schedules is done to determine the sufficient amount of iterations needed for convergence of the full set. Figure 3 shows the average objective values and expected value of the initial state for each scenario. Each scenario is indicated by ”S” for single disruption and ”M” for two disruptions, and a number indicating the reward structure. These are the average values at each iteration for all 200 training instances. Figure 3c shows the absolute values for the average objectives per iteration. It can be seen that with a reward structure that includes higher penalties for last-minute decisions, the convergence is noticeably slower (*M2, M3*), but only for multiple disruption scenarios. This could be attributed to the fact that the benefit of making earlier decisions is not reflected in the immediate rewards, as it seems to the model that making no decision is still better than making a decision (with costs associated) that affects a flight far in the future. Only at a later stage in the time horizon, where the decision is ”more costly”, the model recognizes that making the decision one step earlier would result in better rewards. The point where the decision recognizes the benefit of making earlier decisions thus has to propagate back through the time steps during the iterations, which results in significantly slower convergence. It is remarkable that this effect is less evident in single disruption scenarios (S1, S2, S3). Although this could have to do with the simplicity of the single disruption scenarios and the magnitude of additional benefit of making earlier decisions (which make the effect less apparent in the plot:

7.2. Function approximation

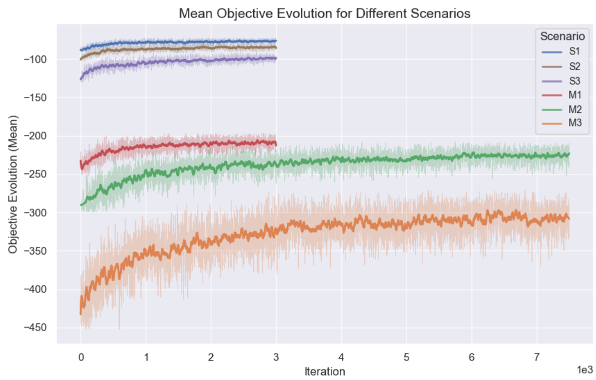
To determine the sizes of the training set that will be used for approximating the value function, the size of the smallest training set for all scenarios is chosen. Due to computational limits, only 200 training instances are used in case of the most computational heavy scenario. For most scenarios, training was done on more than 200 instances. However, the size of the training set influences the computational efficiency of the value approximation during testing and thus the computation times of the solutions during testing. For that reason, and to keep a fair comparison between the scenarios, the training set size is determined at 200 for all scenarios.

7.3. Objectives

This subsection displays the results in terms of model performance w.r.t. the given objectives. The total rewards are displayed, as well as the number of cancellations and the total flight delay in each scenario, for each policy. In Figure 4, the results for these combinations of policies and scenarios are compared. It (logically) becomes clear that multiple disruptions results on higher costs, more delays, and more cancellations.



(a) Average normalized va … ive value obtained per iteration.



(c) Average objective value obtained per iteration for different scenarios

Figure 3: Convergence behavior for the different scenarios on a normalized scale.

The difference between reactive and proactive policies are not immediately evident, although the myopic is noticeably worse in almost all cases. Different reward structures 1, 2 and 3 have – by definition – a large impact on the reward. The impact the different reward structures have on the delay and number of cancellation in the single (S) disruption scenario, is not evident. However, it is remarkable that for multi-disruption (M) scenarios, reward structure 2 (M2) seems to lead to more delays, but less cancelations, than M1 and M3. Another thing that stands out is the fact that for cancellations, the different reward structures in multi-disruption scenarios (M1, M2, M3) seems to influence the advantage of the proactive model versus the reactive one, meaning that with more time penalties, the proactive model manages to find solutions with less cancellations compared to the reactive model and myopic model.

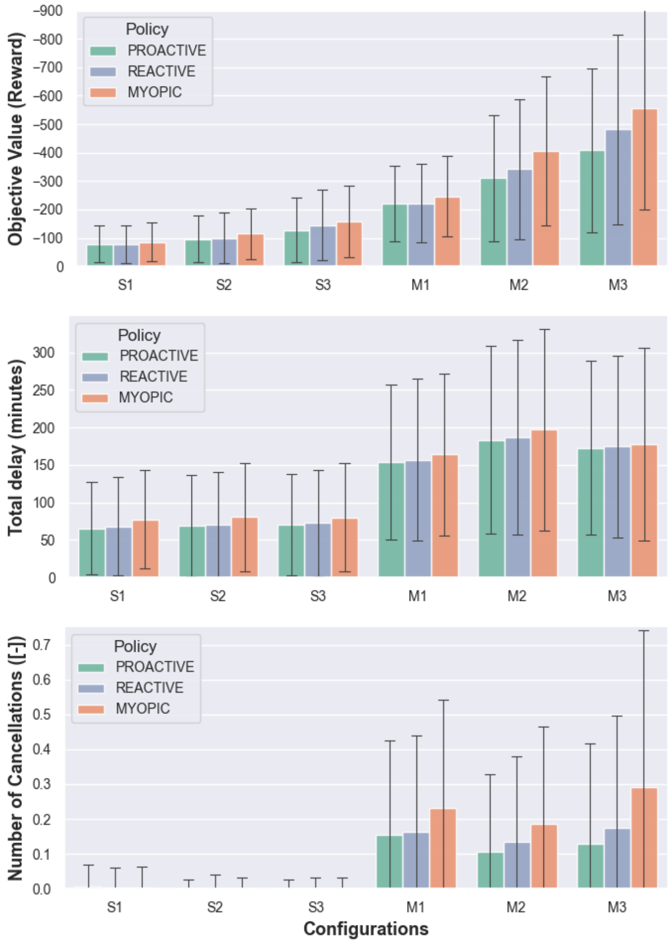


Figure 4: Results for objective metrics across all scenario/policy combinations.

In Figure 5, the performance for R^1_S and R^1_M is compared to that of the exact model. While the optimality gap in single disruption scenarios is large, VFA method achieves good performance in multiple disruption scenarios with a optimality gap of 19.6%. From 100 test instances, the exact method found a solution within the time limit of 100 seconds for 99 instances on the single disruption scenario, and 86 instances for the multiple disruption scenario.

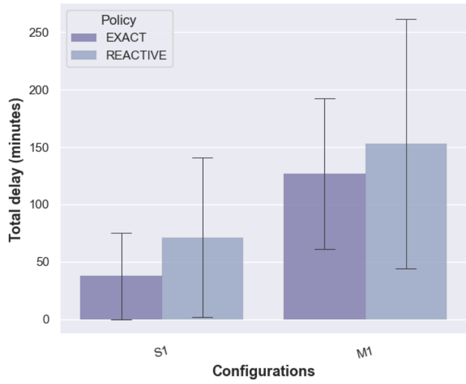
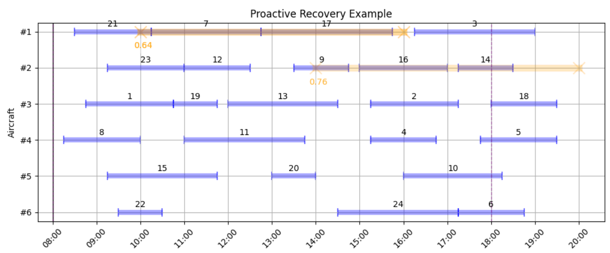


Figure 5: Performance of the reactive policy compared to exact solutions. *(Optimality gaps for $S1$ and $M1$: 61.0% and 19.6% resp.)* – The results are only compared for R^1_S and R^1_M, as the goal of the comparison is to briefly provide insights on the differences between an RL and exact methods, not to perform a comprehensive analysis such as the comparison of proactive and reactive.

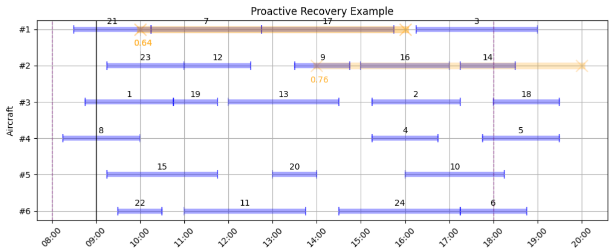
A key difference in the exact solution and the RL solutions, is the number of actions they take. The RL model performs a limited number of actions, and chooses the actions that yield the most benefit for the recovery solution. The exact model however, takes as many actions as necessary to arrive at the optimal solution. In many cases, the RL model can achieve a good solution with only a few actions. This happens largely in scenarios where the recovery solution is more straightforward. Once a ”good” solution becomes less straightforward, the performance gap between exact and RL becomes bigger. In Figures B.13, B.14 and B.15 in the Appendix Appendix B, an example is given of a disrupted schedule where the solution for both a reactive RL solution, and exact solution is shown. As can be seen, the schedule is heavily disrupted with four flights being in conflict. In this example, the RL model reached a solution by performing 7 actions with an objective of -805, while the exact model did 17 actions and found a solution with an objective value of -591. This is a good example of a scenario where the RL model does not manage to find a ”good” solution with only a few actions. It also showcases the ability of the exact method to find good solutions, but at the costs of severe changes to the schedule.

7.4. Proactive actions

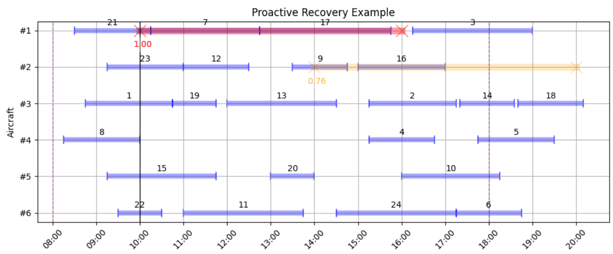
The behavior of the proactive model is important for airlines, since they must decide whether a proactive solution will be implemented or not. In particular, the actions the proactive model does *before* any disruptions have realized. The proactive manages to effectively solves for disruptions by swapping flight around to keep delays to a minimum. By taking some proactive measures (actions), the model creates straightforward recovery options in case a disruption would realize. In Figure 6, a stepwise simulation of a proactive recovery solution is displayed where this happens. Two potential disruptions for aircraft 1 and aircraft 2 are known with probabilities $p = 0.64$ and $p = 0.76$, respectively. Flight 11 gets swapped to aircraft 6 even if there are no potential disruptions for this flight and no other disruptions have yet occurred. However, the model recognizes that aircraft 4 is a good candidate aircraft for the potentially disrupted flights 7 and 17 of aircraft 1. With no delays, flight 11 can be swapped from aircraft 4 to aircraft 6 and as the disruption of aircraft 1 realizes, the model now has created an easy recovery option *in advance* without having additional delays. In a similar way, flight 20 is swapped from aircraft 5 to aircraft 4, which allows flight 17 to be recovered without delays. The actions per step are described below the schedules.



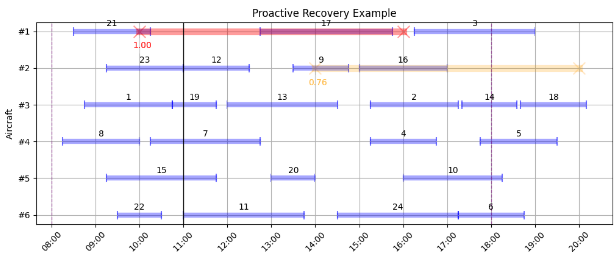
Initial schedule with potential unavailabilities.



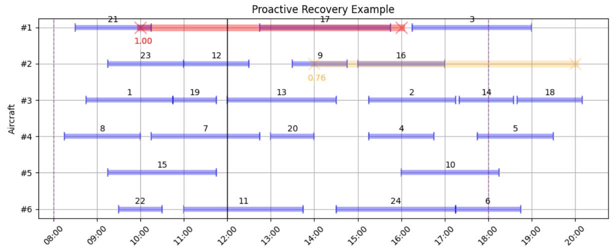
Flight 11 from aircraft 4 to aircraft 6.



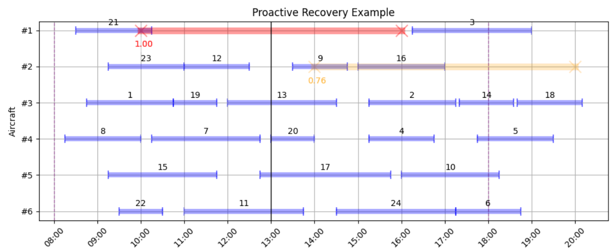
Flight 14 from aircraft 2 to aircraft 3.



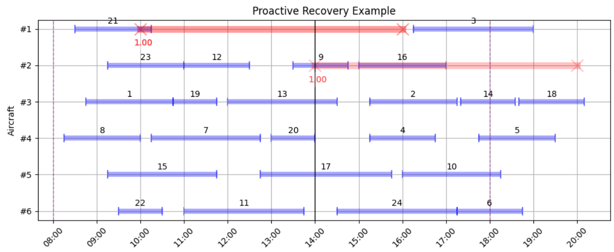
Flight 7 from aircraft 1 to aircraft 4.



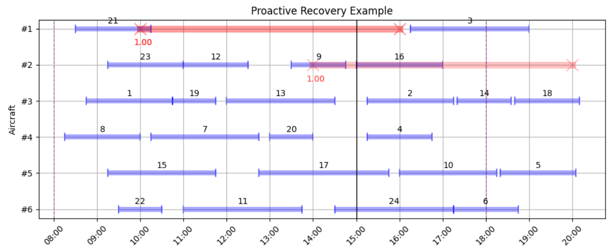
Flight 20 from aircraft 5 to aircraft 4.



Flight 17 from aircraft 1 to aircraft 5.



No action.



Flight 5 from aircraft 4 to aircraft 5

.

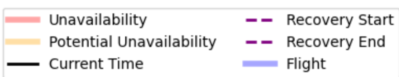


Figure 6: Stepwise simulation of proactive recovery solution to a multi-disruption scenario

7.5. Robustness

Besides assessing the performance of different models in terms of airline objectives, it is of interest to capture the behavior of different policies by quantifying that behavior in measures for robustness. In this context, robustness of a model means: "the extend to which the schedule can be recovered while keeping the changes in the original plan to a minimum" (Chiraphadhanakul and Barnhart, 2013). To assess this, the number of recovery actions, number of involved aircraft, and number of affected flights are compared. Figure 7 depicts the behavior of a proactive model in comparison to a reactive model and (reactive) exact solution. Looking at these results, it is clear that an exact method, while finding optimal solutions, requires more extensive recovery in terms of all three metrics. An RL approach (reactive and proactive) arrives at solutions with fewer actions and impact on the flight schedule. Proactive and reactive approaches seem to share similar characteristics. Although on average, slightly more actions and aircraft are involved in a proactive approach. This is due to the fact that proactive models might make decisions, even when a potential disruption does not realize. Cases in which a reactive model would never act.

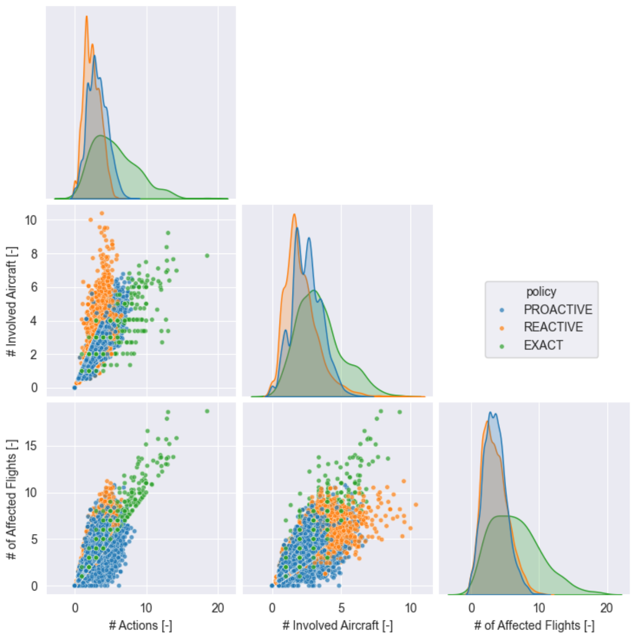


Figure 7: Comparison of nr. of affected flights, involved aircraft and recovery actions for different policies.

Another interesting measure for robustness of the models is how quick a solution is reached. In Figure 8, a comparison is made between the time to solution for the reactive and proactive model (note that by time to solution, the number of time steps until recovery is meant and not the CPU time). A proactive model reaches solutions noticeably quicker, indicating that proactive policies can indeed set up the schedule in such a way that when a disruption happens, a solution can be reached with very little effort. The findings presented in Figure 9 support this. It becomes clear that a proactive model accumulates more negative rewards at an earlier stage than a reactive model, but saves negative rewards later on the day by doing so, and thus limit the corrective measures that have to be undertaken at a later stage.

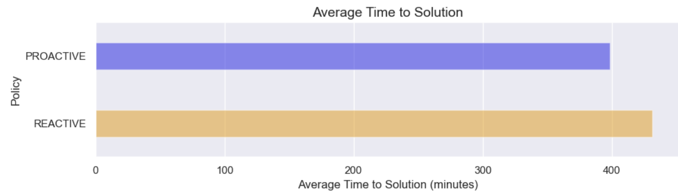


Figure 8: Mean time to recovery of proactive and reactive policies in minutes.

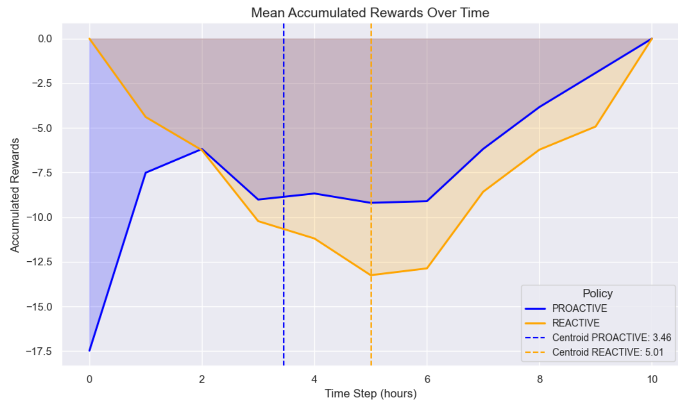
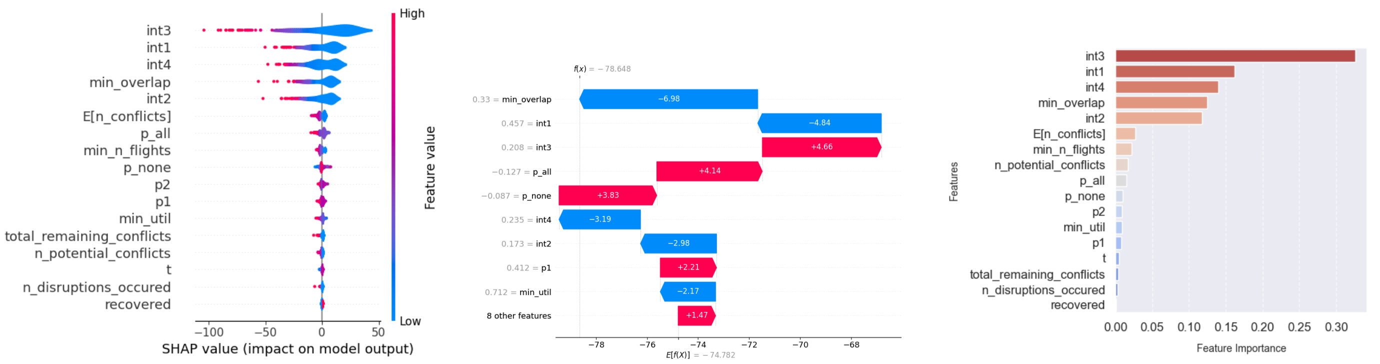


Figure 9: Mean accumulated rewards of proactive and reactive policies over all test schedules.

7.6. *Model reasoning*

Regarding the reasoning of the model (i.e. why the model chooses certain actions), two factors play a role: 1) How and why does the model learn that a certain state represents a good value? 2) How and why does the model estimate a certain state at a certain value, based on its experience/training? The first question can be answered by considering the configuration of the ADP algorithm defined in 4.1. The value of states is learned iteratively and the way these iterations progress is influenced by factors like the stepsize, discount factor, initial state values, exploration vs. exploitation trade-off, as well as the reward structures. For the second question, the SHAP values used are discussed in Section 4.3.2. The results are displayed in Figure 10. In Figure 10a, the impact of the individual features on the model output (prediction of a value) is plotted. The different colors represent high and low values of the features, while the position on the x-axis represents the value that that specific feature value has on the prediction. In Figure 10b, a single example prediction is analyzed, which shows, for that prediction, how much each feature contributed to the value of that prediction w.r.t. the expected value of a prediction (the mean value of all predictions). Additionally, a feature importance graph is displayed in Figure 10c. The feature importance scores measures for each individual feature how much it reduces the variance in the output. Looking at the values in Figure 10a and Figure 10c, it becomes evident that the metrics established in Section 4.2.2 to derive an estimate of the value of a state have the most impact on the model output (*int_1, int_2, int_3, int_4*, Table 4). This tells us that these designed features derived at least serve the purpose being predictors of the state value to some extend. On the other hand, however, it could also imply that the model



(a) Shapley Values for the Random Forest Regression (b) Waterfall plot of SHAP values for a single prediction (c) Feature Importance scores for the random forest regression

Figure 10: Visualization of SHAP explanations, waterfall plot, and feature importance. See Table 4 for the exact definitions of the features.

does not explore enough (since the values of states are also initialized based on these metrics, and thus the model does not explore enough to beyond what it believes is the value of a state initially). In any case, the actions of the model can partly be explained by these features, where the most important ones are derived from the overlap of flights within the fleet and the number of subsequent flights swap candidate aircraft have, which could thus be an intuitive explaining factor to why the model might perform certain actions.

7.7. Computational performance and scalability

The scalability of the model must be assessed through the computing times, which is an important requirement for airlines using such models. Since the curses of dimensionality are only partially tackled in this ADP framework. The action space grows exponentially and each action requires to be evaluated, causing computation times to explode. In this work, the evaluation is done exhaustively, which results in the scaling characteristics found in Figure 11. Doubling the fleet already requires 5 times longer computations. In Figure 12, the CPU times of the RL (reactive) model and the exact model are compared, showing that both do not scale well, but a stronger increase in CPU time is visible for the RL model.

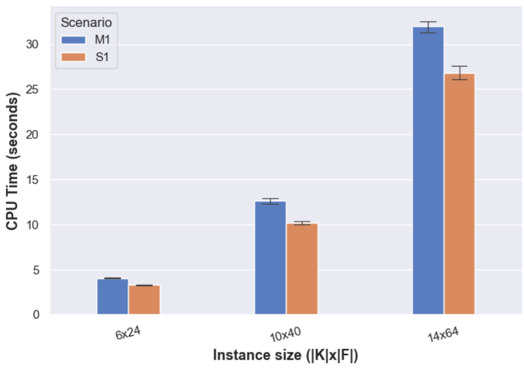


Figure 11: Computational Performance for different instance sizes.

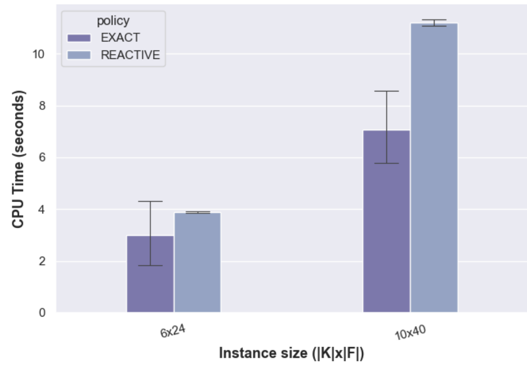


Figure 12: Computational Performance exact optimization vs. reinforcement learning.

8. Hypothesis validation

Four main hypotheses were stated. Several tests have been constructed to be able to draw conclusions and validate the hypotheses:

Hypothesis 1
A Proactive recovery strategy leads to lower recovery costs than reactive strategies. Note that "costs" in this hypothesis is an ambiguous term, and it will therefore be assessed threefold by 1) rewards, 2) delays, 3) cancellations. For each metric, the null and alternative hypothesis are:
$\boldsymbol{H_0}$: $Z_{proactive} = Z_{reactive}$
$\boldsymbol{H_1}$: $Z_{proactive} > Z_{reactive}$

Where Z represents the mean of the samples, observed from the 1000 test instances. It can be seen from Table D.10 that with more stringent reward structures in terms of penalties for time between decision and effect (the last minute time penalty discussed in Section 3.3.5) do have some significant results. For most configurations however, although the means of the proactive objective metrics are lower, there is no statistical significance, meaning that a proactive approach does not necessarily yield lower cost solutions.

Hypothesis 2
The advantage of a proactive approach increases as the schedules get more increasingly disrupted. This time, the comparison is between the mean of the advantage of one policy of another, for a scenario with a single disruption and a scenario with two disruptions. Although the same flight schedules are used as test instances for the different disruption scenarios, the disruptions themselves differ, making the results inherently different and thus independent. Again, the significance of the different mean is tested for each reward structure.
$\boldsymbol{H_0}$: $A_M = A_S$
$\boldsymbol{H_1}$: $A_M > A_S$
Although there is no significant difference for the objectives with reward structure 1, there is a significant increase in relative performance with more disruptions in terms of total rewards and number of cancellations for reward structures 2 and 3 (with these reward structures, last minute decisions are penalized more), see Table D.11.

Hypothesis 3:
The advantage of a proactive approach decreases as the slack of the schedule increases. Here, the slack between two flights is determined as defined by (Chiraphadhanakul and Barnhart, 2013), where it is the *additional* time between two flights in excess of the Minimum Turn Time (MTT). Since in this study, the MTT is included in the flight times, the slack is therefore the time between two consecutive flights. The schedule slack is then determined as the average slack per aircraft relative to the recovery horizon. Again, the test is performed for each reward scenario, and each disruption scenario. Results are displayed in Table D.12.
$\boldsymbol{H_0}$: $r = 0$
$\boldsymbol{H_1}$: $r > 0$

Looking at the table, there seems to be a significant relation between the advantage of a proactive approach and the slack in the schedule in most cases. However, the number of cancellations seems to not be effected statistically by the slack in the schedule.

Hypothesis 4
Proactive recovery allows for quicker recovery. To test the difference in the means of the time to recovery T_r for proactive and reactive policies, the same test is performed as for hypothesis 1:
$\boldsymbol{H_0}$: $T_r^{proactive} = T_r^{reactive}$
$\boldsymbol{H_1}$: $T_r^{proactive} < T_r^{reactive}$

The results are found in Table D.13 and indicate a clear statistical difference between the time to recovery for proactive and reactive models.

9. Discussion

This study showcased the potential of a proactive approach to tackle Airline Disruption Management (ADM) problems through an RL framework that relies on Approximate Dynamic Programming (ADP) with Value Function Approximation (VFA) when faced with uncertain disruption information. Results show that a proactive policy – in circumstances where last minute schedule changes are penalized – lead to better performance in terms of number of flights cancellations. This seems to be the case for the total reward obtained from the RL model and the total delay as well. However, not all results show significant differences of a proactive policy over a reactive one in these areas. Furthermore, additional insights in the actions of the ADP framework are provided by means of an explainability assessment that exposes important problem characteristics that influence decision making by the ADP model. The robustness of the different policies is assessed and compared to an exact method, where results indicate that an RL approach yields more stable solutions over the exact benchmark in terms of schedule intervention, meaning that the RL model achieves, although suboptimal, efficient recovery solutions with significantly less actions than an exact model. Additionally, the proactive model obtains solutions quicker by anticipating the disruptions, resulting in fewer corrective measures when disruptions actually realize. The advantage of a proactive policy is more apparent – especially for flight cancellations – in more heavily disrupted scenarios, as well as scenarios in which flight schedules have little slack, which could motivate AOCC operators to consider proactive recovery strategies when faced with such scenarios.

9.1. Limitations

While this study provided insights into proactive recovery frameworks for ADM, several limiting factors must be discussed. First of all, the proposed models are subject to a variety of assumptions that restrict their resemblance to real life operations; operational constraints are excluded under a simplified hub and spoke model that models rotations as single flights, thereby ignoring important requirements that individual aircraft may have regarding their location (for maintenance or schedule continuity purposes). Uncertainties regarding disruptions are assumed to be known (e.g. duration of unavailability, probability of disruption realizing). Also, the data is synthesized, and therefore required rough assumptions on the data characteristics without investigating the effects of how these assumptions are substantiated. Furthermore, the ”costs” of flight cancellations, swaps, and delays are assumed at some value, which might differ strongly in real life depending on the airlines. Although easily adaptable for airlines that might use this framework, the rewards influence the models’ behavior and thus the insights proactive approaches derived from this study might no longer hold when subject to significant changes in rewards. Lastly, the general ARP is only a piece of the full disruption management problem pie, as operators have to consider crew and passengers into the recovery as well for optimal decisions, which is not included in this study.

In terms of algorithmic limitations, there are several points of attention. Firstly, there is the initialization of state values in the ADP algorithm. This determines for a large part how the algorithm behaves in terms of convergence and exploration. bounded by limited computing power, the choice was made to steer the algorithm towards exploiting states assumed to be good and thus reaching faster convergence, while allowing some, but limited, exploration. The set of basis features on which the states' values are predicted are designed to represent the state globally (the whole system) instead of having features that represent individual state resources (aircraft). This allows the model to generalize to bigger problem sizes, even with training on small problems, but comes at the cost of a loss of detail of the states, which deteriorates the predictions on state values. To add to this, one limitation of the algorithmic architecture is that the values are predicted based on two approximations (learning the real state values by approximation and predicting a state value based on those learned values), which might amplify any inaccuracies the model may have.

Then there is limited computational ability, restricting the models training in terms of duration, training set size, granularity of time steps in the time horizon, and problem sizes. These are all factors that, if not limited, could improve the model performance and allow for unbiased results and thus better insights into the proactive framework when compared to the reactive approach.

9.2. Scalability of RL model

Limited computational ability is a result of inefficient algorithmic performance. The results have shown that the currently proposed ADP model – while tackling some of the curses of dimensionality – misses the elimination of the curse of the action space, and therefore does not scale well when subject to increased problem sizes. This does not mean that the proposed method is unsuited for ADM problems, however. The model is still able to solve instances with a fleet of 20 aircraft in under 2 minutes. However, in many RL applications, one of the advantages is near instantaneous computation times after training. In this work, an exact model, although scaling equally bad, is shown to have superior CPU times over the RL approach. The exploration of a method to bypass the curse of the action space in this proposed framework is therefore needed to open up the full potential of this method to solve large scale problems in a matter of seconds. It must be added that the comparison with the computation times of the exact model is biased, since this exact model does not always find a solution, meaning that some instances with very large computation times are not included in the comparison. This points to the rather advantageous property of the RL method to find solutions with roughly the same CPU times for every instances, where an exact method is not guaranteed to find a solution within the desired runtime, since the user is unsure beforehand whether the solution time will be a matter of seconds or in the order of 10 minutes. To add to this, the scope of both the RL and exact models is overly simplified, and it is well known that MILP's do not handle additional complexity in terms of constraints and variables very well when scaling, while the RL model CPU times still scales linearly (with the action space size that is, not the problem size) when subject to additional constraints. This could lead us to question the computational advantage that the exact model has over the RL approach in this work. More on this is Section 9.4

9.3. Applicability of proactive framework

One major question surrounding this research is the use of such a framework in real-life operations by AOCC operators. Several factors must be considered. First and foremost is the computation time. Operators require solutions within two minutes (Vink et al., 2020), and depending on the size of the problem at hand, it is not a guarantee that this model meets this condition.

Airlines also require decision support models to be transparent in the decision making, meaning that they should understand why a model picks certain actions. For the proposed model, an analysis on the state features that influence the decision the most can give substantial insights into this, allowing airlines to run checks for why a certain action is proposed by the model. However, the model might still make decisions that seem random, even when considering the features and the value predictions for the system's state. In case that this is detrimental for AOCC operators, additional research and insights in the explainability of the model is required to meet the standards for use in AOCC's.

The quality of the solutions should not be forgotten when deciding whether a proactive RL model should be used by airline operators, as this is ultimately the goal of airline recovery. In case exact or heuristic solution – although static and reactive – provide significantly better solutions, the operators might make the trade-off that the proactive RL simply does not meet perform enough in terms of cost savings and decide not to use it.

The proactive model relies on uncertain information in aircraft unavailabilities. It is important for airline operators to consider if the reliability of the uncertain information is sufficient, as the uncertainty of the predictions of disruptions add extra noise to the problem: a proactive model can lead to better recovery outcomes, given that the uncertain information is reliable.

9.4. Recommendations for future work

Since this work is proof of concept for anticipatory disruption management, and little proactive disruption management works exist in literature, there are countless directions for future work on this topic. To begin, the scope of the model can be expanded to include more realistic operational factors and constraints such as multi-fleet scenarios, multi-airport rotations, maintenance routing constraints, soft-reserve aircraft. Incorporating these operational aspects in the framework would not only make the model more resemblant of real life and thus more applicable, it could also reveal an increased performance gain of a proactive strategy over a reactive one. We have already seen with that highly disrupted and highly utilized schedules, the proactive approach has a greater advantage. This could lead us to hypothesize that increasing the complexity and scope of the framework would also increase the advantage of a proactive approach, thereby showcasing the true benefit that anticipatory disruption management can bring. To amplify this, real operational data should be used to allow RL models to exploit underlying patterns of that data, rather than working on synthesized data. This includes the flight schedules that the models are trained on, but also the characteristics of the disruptions, which can stem from real life historical data.

Regarding the model itself, more extensive training should be performed, preferably with tailored exploration strategies, since general exploration is not worth the extra computation time when using ADP with large state spaces. This could abolish the restriction of the model in finding good solutions imposed by too little exploration, making the ADP method much more powerful. Then, the computational challenges of the model should be tackled, and the incorporation of a method that removes the need to evaluate all possible decisions exhaustively must be investigated to improve and validate the scalability of the model. In the ADP paradigm, many methods exist to bypass the curses of dimensionality (Value Function Approximation, Policy Function Approximation, Lookahead Policies, and Cost Function Approximations (Powell and Meisel, 2015)). Although they often must be tailored to the problem at hand and combined with more advanced methods to work properly (function approximation with state features (parametric or non-parametric), or hierarchical aggregation for example). The proposed method still requires a method that can overcome the need for an exhaustive search in the action space. This could be achieved by implementing an internal optimization problem (or any other search algorithm) in the ADP algorithm instead of searching exhaustively, or redefining the MDP in such a way that the action space remains small regardless the problem size. The latter being something that is not straightforward and not guaranteed to work, however. Regardless, this model can still be used for small subsets of fleets that are selected by an aircraft selection algorithm that is used in conjunction with the recovery models, something that is proven useful for the ARP (Vink et al., 2020; Rashedi et al., 2024).

Additionally, different RL methods should be used in the context of aircraft recovery to gain insights in the individual RL methods and single out their limitations and advantages to be able to determine the best suited methods for different disruption management problems.

10. Conclusion

This paper proposes a novel proactive aircraft recovery framework using model-based reinforcement learning (RL) to address airline disruption management problems by formulating the Aircraft Recovery Problem (ARP) as a Markov Decision Process (MDP). In a simplified model scope where hub and spoke rotations are represented as single flights, the proposed anticipatory model integrates information on future potential disruptions, offering a significant difference over conventional reactive strategies. This work's findings highlight the fact that a proactive RL-based framework is particularly effective in high-disruption scenarios, where it outperforms reactive methods in minimizing delays and cancellations. Similarly, the results indicated the advantage of a proactive model in terms of delays increased when the slack of the daily flight schedule decreased, demonstrating the potential of proactive models in highly utilized schedules.

When incorporating time penalties for last minute changes to the schedule, the proactive model obtains better rewards and mitigates flight cancellations better than the reactive model, while no differences become visible in terms of delays. This indicates that proactive models are particularly beneficial to airlines that prioritize passenger satisfaction and desire fewer last minute changes with less cancellations. However, a balance must be struck when quantifying these time penalties, since the results in this study showed that it is not the case that the higher the time penalty, the more the model mitigates cancellations over delays.

Additionally, proactive methods reach faster solutions by proactively setting up the schedule for easy recovery in case of disruptions. This indicates that anticipatory recovery strategies are suited to provide timely and effective solutions to disruptive scenarios. The RL model also provides robust solutions with fewer recovery actions, emphasizing its stability compared to an exact model. The scalability of the model remains a major limitation that should be tackled in future research. Doing so would unlock the full potential of RL models and allow for better practical applicability of the model to real-world operations. The integration of explainability into the RL framework exposed important properties of the problems that explained the reasoning of the model when picking individual actions and could aid operators in making decisions when using such a model as a decision support tool. This adds additional value for viability of real-world application, although it remains a subjective matter for individual airline preferences, and should be investigated as such before real-world application of the model. Nonetheless, this work provided the first steps and insights towards a method for anticipatory airline disruption management. By addressing the remaining practical implications and combining the proactive frameworks with data analytics for probabilistic predictions on disruptions, it could become a viable tool for improving airline resilience when operating in uncertain environments.

References

Arguello, M., 1998. Framework for exact solutions and heuristics for approximate solutions to airlines irregular operations control aircraft routing problem. URL: https://api.semanticscholar.org/CorpusID:86607112.

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

Beirigo, B., Schulte, F., Negenborn, R., 2022. A learning-based optimization approach for autonomous ridesharing platforms with service level contracts and on-demand hiring of idle vehicles. Transportation Science https://doi.org/10.1287/trsc.2021.1069.

Bellman, R., 1966. Dynamic programming. Science 153, 34–37. URL: https://www.science.org/doi/abs/10.1126/science.153.3731.34, doi:10.1126/science.153.3731.34, arXiv:https://www.science.org/doi/pdf/10.1126/science.153.3731.34.

Cadarso, L., Vaze, V., 2022. Passenger-centric integrated airline schedule and aircraft recovery. Transportation Science 57. https://doi.org/10.1287/trsc.2022.1174.

Chiraphadhanakul, V., Barnhart, C., 2013. Robust flight schedules through slack re-allocation. EURO Journal on Transportation and Logistics 2, 277–306. URL: https://www.sciencedirect.com/science/article/pii/S2192437620301126, doi:https://doi.org/10.1007/s13676-013-0028-y.

Clarke, M.D.D., 1998. Irregular airline operations: a review of the state-of-the-practice in airline operations control centers. Journal of Air Transport Management 4, 67–76. https://doi.org/10.1016/S0969-6997(98)00012-X.

Ding, Y., Wandelt, S., Wu, G., Xu, Y., Sun, X., 2023. Towards efficient airline disruption recovery with reinforcement learning. Transportation Research Part E: Logistics and Transportation Review 179, 103295. https://doi.org/10.1016/j.tre.2023.103295.

Eggenberg, N., Salani, M., Bierlaire, M., 2010. Constraint-specific recovery network for solving airline recovery problems. Computers Operations Research 37, 1014–1026. URL: https://www.sciencedirect.com/science/article/pii/S030505480900210X. https://doi.org/10.1016/j.cor.2009.08.006.

Eikelenboom, B., 2022. A real-time decision-support tool for the integrated airline recovery using a machine learning approach. MSc. Thesis, Faculty of Aerospace Engineering, Delft University of Technology http://resolver.tudelft.nl/uuid:fbe1a4af-54f6-44fc-93ed-6e1069b4ed2a.

Fife, D., D'Onofrio, J., 2022. Common, uncommon, and novel applications of random forest in psychological research. Behavior Research Methods 55, 1–20. doi:10.3758/s13428-022-01901-9.

Gershkoff, I., 2016. "airline disruption management". accessed on 20.05.2024. https://amadeus.com/documents/en/blog/pdf/2016/09/airline-disruption-management-whitepaper-2016.pdf.

Hassan, L., 2019. Aircraft disruption management increasing performance with machine learning predictions. MSc. Thesis, Faculty of Aerospace Engineering, Delft University of Technology http://resolver.tudelft.nl/uuid:07efd193-1d6a-43ad-9461-ccc885dfcb9a.

Hassan, L., Santos, B., Vink, J., 2021. Airline disruption management: A literature review and practical challenges. Computers Operations Research 127, 105137. https://doi.org/10.1016/j.cor.2020.105137.

He, L., Levine, R., Fan, J., Beemer, J., Stronach, J., 2018. Random forest as a predictive analytics alternative to regression in institutional research. Practical Assessment, Research and Evaluation 23, 1–16.

Heinold, A., 2024. A tutorial on value function approximation for stochastic and dynamic transportation. 4OR 22, 145–173. https://doi.org/10.1007/s10288-023-00539-.

Hondet, G., Delgado, L., Gurtner, G., 2018. Airline disruption management with aircraft swapping and reinforcement learning. HAL Open Science https://enac.hal.science/hal-01944612.

Hu, Y., Liao, H., Zhang, S., Song, Y., 2017. Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft. Expert Systems with Applications 83, 283 – 299. https://doi.org/10.1016/j.eswa.2017.04.031.

Huang, Z., Luo, X., Jin, X., Karichery, S., 2022. An iterative cost-driven copy generation approach for aircraft recovery problem. European Journal of Operational Research 301, 334–348. https://doi.org/10.1016/j.ejor.2021.https://doi.org/10.055.

Janssen, A., Hoogendoorn, M., Cnossen, M., Mathôt, R., 2022. Application of shap values for inferring the optimal functional form of covariates in pharmacokinetic modeling. CPT: Pharmacometrics Systems Pharmacology 11. doi:10.1002/psp4.12828.

Kohl, N., Larsen, A., Larsen, J., Ross, A., Tiourine, S., 2007. Airline disruption management—perspectives, experiences and outlook. Journal of Air Transport Management 13, 149–162. https://doi.org/10.1016/j.jairtraman.2007.01.001.

Le, M., Wu, C., 2013. Solving airlines disruption by considering aircraft and crew recovery simultaneously. J. Shanghai Jiaotong Univ. (Sci.) 18, 243–252. https://doi.org/10.1007/s12204-013-1389-y.

Lee, J., Lee, K., Moon, I., 2022. A reinforcement learning approach for multi-fleet aircraft recovery under airline disruption. Applied Soft Computing 129, 109556. https://doi.org/10.1016/j.asoc.2022.109556.

Lee, J., Marla, L., Jacquillat, A., 2020. Dynamic disruption management in airline networks under airport operating uncertainty. Transportation Science 54(4), 973–997. https://doi.org/10.1287/trsc.2020.0983.

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

Maher, S., 2015. A novel passenger recovery approach for the integrated airline recovery problem. Computers Operations Research 57, 123–137. https://doi.org/10.1016/j.cor.2014.11.005.

Marla, L., Vaaben, B., Barnhart, C., 2017. Integrated disruption management and flight planning to trade off delays and fuel burn. Tranportation Science 51. https://doi.org/10.1287/trsc.2015.0609.

Ogunsina, K., Bilionis, I., DeLaurentis, D., 2021. Exploratory data analysis for airline disruption management. Machine Learning with Applications 6, 100102. https://doi.org/10.1016/j.mlwa.2021.100102.

Ogunsina, K., Papamichalis, M., Bilionis, I., DeLaurentis, D., 2019. Hidden markov models for pattern learning and recognition in a data-driven model for airline disruption management. https://doi.org/10.2514/6.2019-3508.

Ogunsina, K., Papamichalis, M., DeLaurentis, D., 2022. Relational dynamic bayesian network modeling for uncertainty quantification and propagation in airline disruption management. Engineering Applications of Artificial Intelligence 112, 104846. https://doi.org/10.1016/j.engappai.2022.104846.

Petersen, J., Sölveling, G., Clarke, J.P., Johnson, E., Shebalov, S., 2012. An optimization approach to airline integrated recovery. Transportation Science 46, 482–500. https://doi.org/10.2307/23362876.

Powell, W., 2011. Approximate Dynamic Programming - Solving the Curses of Dimensionality. second ed., Wiley.

Powell, W., Meisel, S., 2015. Tutorial on stochastic optimization in energy—part i: Modeling and policies. IEEE Transactions on Power Systems 31, 1–9. doi:10.1109/TPWRS.2015.2424974.

Rashedi, N., Sankey, N., Vaze, V., Wei, K., 2024. A machine learning approach for solution space reduction in aircraft disruption recovery. European Journal of Operational Research URL: https://www.sciencedirect.com/science/article/pii/S0377221724008944, doi:https://doi.org/10.1016/j.ejor.2024.11.025.

Rhodes-Leader, L., Nelson, B., Onggo, S., Worthington, D., 2022. A multi-fidelity modelling approach for airline disruption management using simulation. Journal of the Operational Research Society 73, 2228–2241. https://doi.org/10.1080/01605682.2021.1971574.

Santos, B.F., Wormer, M.M., Achola, T.A., Curran, R., 2017. Airline delay management problem with airport capacity constraints and priority decisions. Journal of Air Transport Management 63, 34–44. https://doi.org/10.1016/j.jairtraman.2017.05.003.

Simão, H., Day, J., George, A., Gifford, T., Nienow, J., Powell, W., 2009. An approximate dynamic programming algorithm for large-scale fleet management: A case application. Transportation Science 43, 178–197. doi:10.1287/trsc.1080.0238.

Stuke, A., Rinke, P., Todorović, M., 2021. An approximate dynamic programming algorithm for large-scale fleet management: A case application. Machine Learning Science and Technology 2. doi:10.1088/2632-2153/abee59.

Su, Y., Xie, K., Wang, H., Liang, Z., Art Chaovalitwongse, W., Pardalos, P.M., 2021. Airline disruption management: A review of models and solution methods. Engineering 7, 435–447. URL: https://www.sciencedirect.com/science/article/pii/S2095809921000175. https://doi.org/10.1016/j.eng.2020.08.021.

Teodorovic, D., Guberinic, S., 1984. Optimal dispatching strategy on an airline network after a schedule perturbation. European Journal of Operational Research 15, 178–182. URL: https://ideas.repec.org/a/eee/ejores/v15y1984i2p178-182.html.

Teodorovic, D., Stojkovic, G., 1990. Model for operation delay airline scheduling. Transportation Planning and Technology 14, 273–285. URL: https://trid.trb.org/view/307867.

Thengvall, B., Bard, J., Yu, G., 2000. Balancing user preferences for aircraft schedule recovery during irregular operations. IIE Transactions (Institute of Industrial Engineers) 32, 181 – 193. https://doi.org/10.1080/07408170008963891.

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

Thengvall, B., Bard, J., Yu, G., 2003. A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. Transportation Science 37, 392 – 407. https://doi.org/10.1287/trsc.37.4.392.23281.

Van Heeswijk, W., Mes, M., Schutten, M., 2015. An approximate dynamic programming approach to urban freight distribution with batch arrivals, pp. 61–75. https://doi.org/10.1007/978-3-319-24264-4_5.

Vink, J., Santos, B., Verhagen, W., Medeiros, I., Filho, R., 2020. Dynamic aircraft recovery problem - an operational decision support framework. Computers Operations Research 117, 104892. https://doi.org/10.1016/j.cor.2020.104892.

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

Wang, D., Wu, Y., Hu, J.Q., Liu, M., Yu, P., Zhang, C., Wu, Y., 2019. Flight schedule recovery: A simulation-based approach. Asia-Pacific Journal of Operational Research 36, 1940010. https://doi.org/10.1142/S0217595919400104.

Wu, Z., Gao, Q., Li, B., Dang, C., Hu, F., 2017c. A rapid solving method to large airline disruption problems caused by airports closure. IEEE Access 5, 26545 – 26555. https://doi.org/10.1109/ACCESS.2017.2773534.

Wu, Z., Li, B., Dang, C., 2017b. Solving multiple fleet airline disruption problems using a distributed-computation approach to integer programming. IEEE Access 5, 19116–19131. https://doi.org/10.1109/ACCESS.2017.2747155.

Wu, Z., Li, B., Dang, C., Hu, F., Zhu, Q., Fu, B., 2017a. Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming. Neurocomputing 269, 232 – 255. https://doi.org/10.1016/j.neucom.2017.02.091.

Yetimoğlu, Y., Aktürk, S., 2021. Aircraft and passenger recovery during an aircraft's unexpected unavailability. Journal of Air Transport Management 91, 101991. https://doi.org/10.1016/j.jairtraman.2020.101991.

Zang, H., Zhu, J., Zhu, Q., Gao, Q., 2024. A proactive aircraft recovery approach based on airport spatiotemporal network supply and demand coordination. Computers Operations Research 165, 106599. https://doi.org/10.1016/j.cor.2024.106599.

Zhao, A., Bard, J.F., Bickel, J.E., 2023. A two-stage approach to aircraft recovery under uncertainty. Journal of Air Transport Management 111, 102421. https://doi.org/10.1016/j.jairtraman.2023.102421.

Zhong, H., Lian, Z., Zhou, T., Niu, B., 2024. A time-varying competitive swarm optimizer for integrated flight recovery with multi-objective and priority considerations. Computers Industrial Engineering 190, 110019. https://doi.org/10.1016/j.cie.2024.110019.

Appendix A. Acronyms

Table A.9: List of abbreviations

Abbreviation	**Definition**
AC	*Airport Closure*
ADM	*Airline Disruption Management*
ADP	*Approximate Dynamic Programming*
AI	*Artificial Intelligence*
AOCC	*Airline Operations Control Center*
ARP	*Aircraft Recovery Problem*
CG	*Column Generation*
DRL	*Deep Reinforcement Learning*
DP	*Dynamic Programming*
DQN	*Deep Q-Networks*
GP	*Gaussian Process*
GRASP	*Greedy Randomized Adaptive Search Procedure*
HMM	*Hidden Markov Model*
IRROPS	*Irregular Operations*
LP	*Linear Program*
LNS	*Large Neighborhood Search*
MAS-ADP	*Multi-Agent Simulation-Adaptive Dynamic Programming*
MC	*Monte Carlo*
MCTS	*Monte Carlo Tree Search*
MDP	*Markov Decision Process*
MIP	*Mixed Integer Program*
MILP	*Mixed Integer Linear Program*
ML	*Machine Learning*
MPC	*Model Predictive Control*
OR	*Operations Research*
PHM	*Prognostics and Health Monitoring*
RFR	Random Forest Regression
RL	*Reinforcement Learning*
SHAP	SHapley Additive exPlanations
UTFM	*Uncertainty Transfer Function Model*
VFA	*Value Function Approximation*
VNS	*Variable Neighborhood Search*
VRP	*Vehicle Routing Problem*

Appendix B. Exact vs. RL Comparison

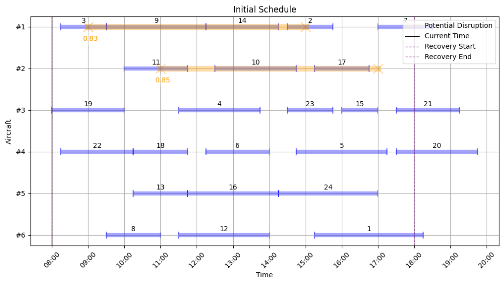


Figure B.13: Initial flight schedule for example scenario.

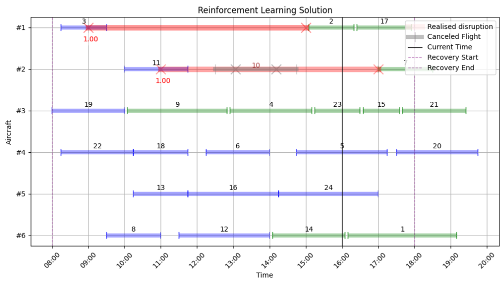


Figure B.14: Solution from an RL model. Objective value of -805. Altered flights are marked in green.

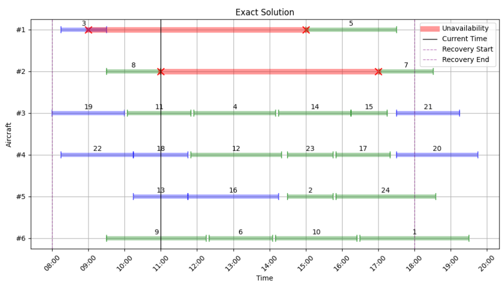


Figure B.15: Solution from an exact model. Objective value of -591. Altered flights are marked in green.

Appendix C. ARP - MILP Formulation

Sets

Symbol	Definition
F	set of flights, indexed by f.
K	set of aircraft, indexed by k.
f, f'	pair of flights such that $f \neq f'$.

Parameters

**The values of parameters c_{swap}, c_{delay} and c_{cancel} are the same as used in the ADP algorithm, defined by Table 5.*

Parameter	Definition
d_f	scheduled departure time of flight f.
a_f	scheduled arrival time of flight f.
c_{swap}	cost of swapping a flight to another aircraft.
c_{delay}	cost per minute of flight delay.
c_{cancel}	cost of canceling a flight.
M	sufficiently large constant for the big-M constraint
F_k^{max}	maximum number of flights that can be assigned to aircraft k.
u_k^{start}, u_k^{end}	start and end times of aircraft k's unavailability.
k_0^f	original aircraft k of flight f.

Decision Variables

Variable	Definition
$x_{f,k}$	1 if flight f is assigned to aircraft k, 0 otherwise.
$s_{f,k}$	1 if flight f is assigned to a different aircraft k than its original aircraft, 0 otherwise.
z_f	delay imposed on flight f in minutes.
y_f	1 if flight f is canceled, 0 otherwise
$\delta_{f,'f}$	1 if f is strictly before f', 0 otherwise.
$\sigma_{f,f',k}$	1 if f and f' are assigned to the same aircraft, 0 otherwise.
$b_{f,k}$	1 if flight f departs before the unavailability of aircraft k, 0 otherwise.
$a_{f,k}$	1 if flight f departs after the unavailability of aircraft k, 0 otherwise.

Exact Model Formulation

The full model is depicted in Figure C.16. The constraints can be grouped into 5 types of constraints:

1. Flight Assignment Constraints.
2. No Overlap Between Flights on the Same Aircraft.
3. Aircraft Availability Constraints.
4. Maximum Flights per Aircraft.
5. Non-Negativity and Binary Constraints.

$$
\begin{array}{llr}
\text{Minimize} \quad \sum_{f\in F}\left(c_{\text{swap}}\sum_{k\in K} s_{f,k} + c_{\text{delay}} z_f + c_{\text{cancel}} y_f\right) & & \\
\text{S.t.} \quad \sum_{k\in K} x_{f,k} + y_f = 1, & \forall f \in F & (1.1)\\
s_{f,k} \le x_{f,k} - x_{f,k_0^f}, & \forall k \in K,\ k \neq k_0^f & (1.2)\\
d_f + z_f \le a_{f'} + z_{f'} + M(1-\delta_{f,f'} + y_f), & \forall k \in K, \forall f, f' \in F,\ f \neq f' & (2.1)\\
d_{f'} + z_{f'} \le a_f + z_f + M(1-\delta_{f',f} + y'_f), & \forall k \in K, \forall f, f' \in F,\ f \neq f' & (2.2)\\
\sigma_{f,f',k} \le \delta_{f,f'} + \delta_{f',f} + y_f + y'_f, & \forall k \in K, \forall f, f' \in F,\ f \neq f' & (2.3)\\
d_f + z_f \le u_k^{\text{start}} + M(1-a_{f,k}) - My_f, & \forall f \in F,\ k \in K & (3.1)\\
d_f + z_f \ge u_k^{\text{end}} - M(1-a_{f,k}) + My_f, & \forall f \in F,\ k \in K & (3.2)\\
\sum_{f\in F} x_{f,k} \le F_k^{\max}, & \forall k \in K & (4)\\
x_{f,k},\ s_{f,k},\ y_f,\ \delta_{f,f'},\ \sigma_{f,f',k}, a_{f,k}, b_{f,k} \in \{0,1\}, & z_f \in \mathbb{Z}^+ & (5)
\end{array}
$$

Figure C.16: Mathematical Formulation of ARP used in this paper

Appendix D. Statistical results

To determine the right statistical test for the first hypothesis, a few factors must be considered. The test is paired since the same instances are compared for the two policies. The samples are tested for normality with a *Shapiro-Wilk* test, which resulted in p-values lower than 0.0001 for all scenarios. The appropriate statistical model is then a *Wilcoxon Signed-Rank* test. The significance of the different means is tested for each reward structure and disruption scenario. In Table D.10, the resulting p-values are displayed.

Table D.10: p-values for Wilcoxon Signed-Rank test to compare the means of objective metrics for multiple scenarios.

Unavailabilities	Single			Multiple		
Reward Structure	1	2	3	1	2	3
Reward	$p = .096$	$p = .383$	$p < .001$	$p = .963$	$p < .001$	$p < .1154$
Total Delay	$p = .458$	$p = .649$	$p = .338$	$p = .510$	$p = .863$	$p = .838$
Cancellations	$p = .475$	$p = .581$	$p = 1.00$	$p = .447$	$p = .005$	$p < .001$

Similarly, for the second hypothesis, the distribution of the samples is tested via a Shapiro-Wilk test, also indicating that the data is non-normal. The appropriate test for this statistic is therefore a Mann-Whitney U test:

Table D.11: p-values for Mann-Whitney U test to compare the means of the proactive-advantage for different objective metrics with multiple rewards structures.

Reward structure	**1**	**2**	**3**
Reward	$p = .517$	$p = .009$	$p < .001$
Total Delay	$p = .950$	$p = .725$	$p = .282$
Cancellations	$p = .178$	$p = .001$	$p < .001$

To test the significance of the relation between schedule slack and the advantage of a proactive approach over a reactive one, a Pearson's test is performed on the the correlation r between the schedule slack as independent variable and the advantage of a proactive approach:

Table D.12: p-values for Pearson's test on the correlation between slack and proactive advantage.

Unavailabilities	Single			Multiple		
Reward Structure	1	2	3	1	2	3
Reward	$p < .001$	$p < .001$	$p < .001$	$p = .0075$	$p = .0127$	$p = .0003$
Total Delay	$p < .001$	$p < .001$	$p < .001$	$p = .0010$	$p < .001$	$p < .001$
Cancellations	$p = .3570$	$p = .4697$	$p = .4083$	$p = .9584$	$p = .9152$	$p = .0918$

To determine the appropriate test for the fourth hypothesis, the same reasining leads to a Wilcoxon Signed-Rank test:

Table D.13: p-values for Wilcoxon Signed-Rank test to compare the means of the time to recovery of a proactive and reactive policy, for multiple scenarios

Unavailabilities	Single			Multiple		
Reward Structure	1	2	3	1	2	3
$T_r^{proactive} < T_r^{reactive}$	$p < .001$	$.p < .001$	$p < .001$	$p < .001$	$p < .001$	$p < .001$

Appendix E. Random Forest hyperparameters

Table E.14: (Bayesian) optimized hyperparameters for training data of different configurations.

	M1	M2	M3	S1	S2	S3
max_depth	15	28	17	17	13	17
max_features	0.46	0.3	0.3	0.3	1.0	1.0
min_samples_leaf	1	1	1	1	1	1
min_samples_split	2	2	2	2	2	2
n_estimators	250	250	250	250	250	250

Part II

Literature Report

*This part has been assessed for the course AE4020 Literature Study.

1 Introduction

Airlines run tight optimized schedules in order to make profit in a competitive market where margins are thin. Due to the highly optimized nature of the airlines' operations, disruptive events can pose a significant problem every day. Many factors can lead to disruptions. Bad weather conditions, aircraft technical problems, airport congestion or airline delays are some of these causes. For a large portion, these disruption are uncertain in when they will occur and how severe their impact will be in the short and long term. The potential impact of these disruptions is vast however. For example, according to Gershkoff (2016), airline disruptions cost an estimated 60 billion annually worldwide, which is around 8% of worldwide airline revenue. Not only do airlines incur significant extra costs by having to compensate crew, passengers or even airports, they can also experience customer dissatisfaction which can have an influence on future revenue streams.

In disruption management, the recovery of the airlines' resources (i.e. aircraft, crew, passengers) during and after disruptive events is referred to as the Airline Recovery Problem (ARP). The goal of this research is to address the ARP by developing a novel solution method that anticipates potential future disruptions using a Model-Based Reinforcement Learning (RL) algorithm. Disruption management is currently done by airlines in a reactive manner, where airline operators only act after disruptive events have occurred. This sparks the belief that proactive methods (i.e. methods where there is acted in advance of future disruptions with a high probability of occurring, without knowing if these disruption will actually happen) yield better performance for airline recovery. That is, a proactive approach would result in less flight cancellations and delays then current reactive methods. Reinforcement Learning exploits patterns in data in such a way that future scenarios can be anticipated, paving the way for a shift from reactive disruption management to proactive disruption management for airlines. This MSc Thesis is conducted in collaboration with Boeing.

The structure of this report is as follows: In Chapter 2, the problem surrounding the current state-of-the-art is explained as well as the necessary background information on airline recovery. In Chapter 3, a review of the (published) literature regarding the aircraft oriented airline recovery is done and the state-of-the-art is discussed. The state-of-the-art surrounding learning based approaches is also highlighted in this Chapter. In Chapter 4, the relevant research gaps are identified that will be addressed by the research, after which the scope and main objectives of the research are defined further, and the research questions are formulated. Lastly, the literature study and research definition are summarized in a concluding remark in Chapter 5.

2 Problem Statement and Background Information

A disruption is an event in which airlines are prevented from running their operational schedules as planned due to additional unforeseen imposed restrictions to the airlines' operations . Causes of disruptions are plentiful, but bad weather conditions, aircraft technical problems, airport congestion are some of the causes that can lead to schedule disruptions. These causes can result in airport closures, late or unavailable aircraft, restricted airport capacity, which in turn requires airlines to modify their schedule: Aircraft may have to deviate from their planned route, and downstream flights cannot depart on time because of late or unavailable aircraft as a result of these deviations. Crews cannot perform their initially planned schedules and passengers might miss their connection at hub airports.

Small disruptive events can already have an enormous impact on airlines' operations if not managed properly: delays accumulate easily throughout the interconnected flight schedules. According to Walker (2017), 24% of all European flights experienced delays in Q3 of 2017. Reactionary delays (delays due to late aircraft, crew, passengers or baggage causes by prior schedule delays) are the identified as the main cause of all delays in European flights in March 2024 (Eurocontrol, 2024). Besides posing a significant operational challenge, not being able to restore the schedule in a fast and efficient manner also forms a significant financial burden for airlines, having to compensate passengers and crew during Irregular Operations as well as facing extra operational costs. Not to mention the "soft costs" associated with the future effects of potential customer dissatisfaction, which is estimated to cost airlines an additional 75$ per passenger per cancelled flight (Marks, 2014). Needless to say it is extremely important for airlines to be able to mitigate these disruptions fast and effectively and direct the planning back to the original schedule. Disruptions happen due to unforeseen events are therefore highly unpredictable in nature. Nevertheless, there are some disruption types in the aviation industry that come with a certain quasipredictability as they come closer in time (Serrano and Kazda, 2017). For example, weather forecasts become more accurate at a shorter time horizon, and aircraft technical failure can be predicted better with Prognostics and Health Monitoring (PHM) when aircraft equipment gets closer to their end of life (Xu et al., 2021). Congestion also can be anticipated better as updated and more accurate demand-capacity information becomes available (Lee et al., 2020).

The process of Airline Disruption Management (ADM) can be divided into three phases and is summarized as follows by Serrano and Kazda (2017): First, the disruption happens and the process enters into the emergency phase. In this phase the impact of the disruption is not known yet as its duration and severeness are still uncertain. Then, the continuity phase begins where the directly affected resources, personnel and customers are helped, managed or recovered. After this, the recovery phase starts, where the airline actively puts in effort to bring back the schedule to it's normal operations. This phase lasts until the schedule is fully recovered. As the name suggests, the *Airline Recovery Problem* is focused on the recovery phase. Generally, the first two phases are not considered in research frameworks surrounding airline recovery. Before diving into the practice of airline disruption management, the process of general airline planning must be understood.

2.1. Airline Planning

The planning of an initial airline schedule is a complex task that is solved in a sequential planning process (Wen et al., 2021), where the most expensive and least flexible resources and actions are planned first. The main resources used in airline planning are aircraft, crew and passengers (Kohl et al., 2007). Planning starts at a tactical level, with determining the airlines' fleet and corresponding flight network for a specified time period based on expected passenger demand. After fleet and network planning, the airlines schedule their flights and assign aircraft types to flights (Fleet Assignment) and a transition to operational planning is made. After the flight schedule is defined, the airlines focus on crew scheduling, where crew members are assigned to flights by subsequently pairing and rostering them (Kohl et al., 2007). A few days before operations, the individual aircraft tail numbers are assigned to flights (Tail Assignment), parallel to any changes in the crew schedule. A schematic depiction of airline planning is shown in Figure 2.1.

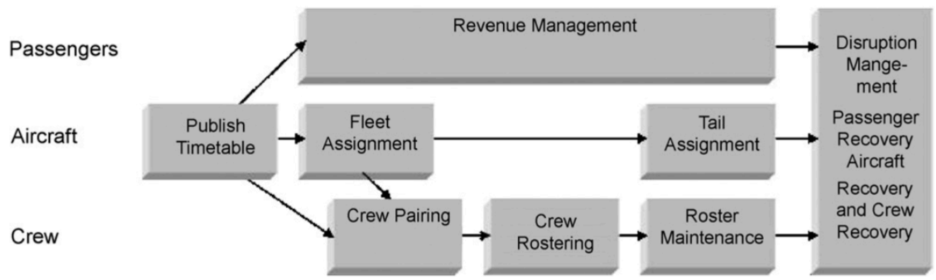


Figure 2.1: Schematic depiction of airline planning, Adopted from Kohl et al. (2007)

2.2. Disruption Management

Similar to airline planning, airline recovery is usually done in a sequential manner. Starting with aircraft, as the most vital resource, then crew, as crew is imperative for operating, and then passengers, who are the main source of revenue for airlines. Disruption Management takes place at the operational planning level. In Figure 2.2, a schematic depiction of the DM process is shown.

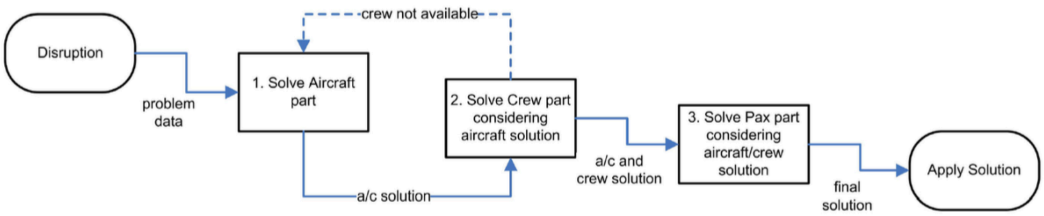


Figure 2.2: Schematic depiction of the airline recovery process. Adopted from Castro et al. (2014)

The ARP is a complex problem, since it tries to restore a highly interconnected and optimized flight schedule that has little slack, and large amounts of data and possible actions have to be processed. In an Airline Operations Control Center (AOCC), human evaluation of recovery actions and decisions is necessary real-time and, to allow this human assessment, a sequential process is the only viable option. In practice, Disruption Management teams consist of different groups focused on flight dispatch, crew, maintenance requirements, and client support, respectively. These groups are overarched by the operations controller, that act as the central function (Fogaça et al., 2022). Airline recovery operators aim to reach a fast but potentially suboptimal solution to the problem, while trying to minimize disruption effects (number of delays, delay times, revenue losses, additional costs) (Su et al., 2021). Controllers in the AOCC actively and continuously monitor the airlines' operations for any unexpected events. When a disruption becomes known to the AOCC, human controllers assess the disruption and any potential recovery solutions, in case any actions should be taken. For this task, however, computer systems can be of great aid to help evaluate decisions real time and process the large amounts of operational data (Kohl et al., 2007).

When determining how to recover the flight schedule as efficiently as possible, airline operators can take several actions regarding recovery for each resource. Recovery actions that airlines can take include, but are not limited to, the following actions: (Su et al., 2021)

- **Tail swapping** - Swapping the assignment of two or more individual aircraft to their respective flights;
- **Delaying** - Purposely delaying a flight, to add slack and allow for connections;
- **Cancellation** - Cancel a flight such that the previously utilized aircraft for that flight becomes available, or such that other flights in the schedule are not affected;
- **Ferrying** - Flying an aircraft from one location to another without passengers;
- **Delay maintenance tasks** - Postponing a scheduled maintenance task.
- **Swap maintenance tasks** - Swapping the assignment of two maintenance tasks for different aircraft
- **Stand-by aircraft utilization** - Having a stand-by aircraft ready at an airport that can act as a substitute aircraft for a flight.
- **Deadheading** - Relocating crew from one airport to another.

Note that recovery actions regarding passengers cannot help recover the schedule directly, but actions like passenger reaccomodation (changing their itineraries) can be taken to minimize the financial losses and passenger dissatisfaction, which is ultimately the goal in disruption management.

The need for rapid evaluation of possible recovery actions prevents controllers from finding optimal solutions. This is due to the interconnectedness of the airlines' resources in their operating schedules; rapid evaluation of these interconnected schedules is too complex for human controllers and can only be done in a sequential process (as depicted in Table 2.2). However, recovering interconnected schedules in a sequential manner leads to suboptimal solutions. Integrated approaches at recovery (taking aircraft, crew and passengers into account simultaneously when making recovery decisions), is not only too complex for rapid human decision making, it also requires too much computational effort for decision support tools and these are therefore of limited use to controllers, who have to evaluate and make decisions within under two minutes (Vink et al., 2020).

Ideally, airlines would find optimal solutions to recover their schedules. However, the complexity of the problem and the need for rapid solutions forms a gap between theoretical solution methods and the practical application thereof. In other words, the practical implications of DM impair theoretical methods to be easily applied. Advancements in technology and research can potentially bridge this gap in the future, allowing airlines to save significant costs incurred by disruptions. Novel solution methods and different approaches to ADM are actively being researched. Recent developments in Data Science and Machine Learning (ML) add a promising outlook to this, providing new computational possibilities and opening the door to new solution methods in ADM. The trends and developments in the field of research of ADM are discussed in detail in a literature review in Chapter 3, as well as a the state-of-the-art of Reinforcement Learning for Operations Management, and Reinforcement Learning for Disruption Management specifically. The motivation behind the exploration of DM in combination with RL will become clear in Chapter 3.

3

Literature Review

In this chapter, two bodies of relevant literature are reviewed: 1) Literature addressing Airline Disruption Management and 2) Literature on Reinforcement Learning methods for Operations Management. First, a systematic review of the published literature on Airline Disruption Management is provided and distinguished in terms of scope and methodology, exploring different types of recovery problems and their respective disruption types, recovery actions, solution methods and objectives. This systematic review is done from an aircraft recovery perspective, and will cover papers that involve variants of the aircraft recovery problem. Subsequently, a review of papers on the use of RL in Operations Management are discussed to highlight the potential of this research field for ADM applications, after which literature on the use of RL in Air Transport Management and ADM is reviewed and discussed.

3.1. Airline Disruption Management

Airline Disruption Management is an active field of research. In recent years there is a significant increase in publications regarding ADM. A SCOPUS search was done on articles and reviews published until 2024 with the query *("aircraft recovery" OR "airline recovery" OR "airline disruption management" OR "crew recovery" OR "passenger recovery" OR "air cargo recovery")*, and 171 papers were found. From Figure 3.1, it becomes clear that this is a growing field of research, with, apart from a dip in 2018, a drastically increasing number of publications per year from 2012 onward.

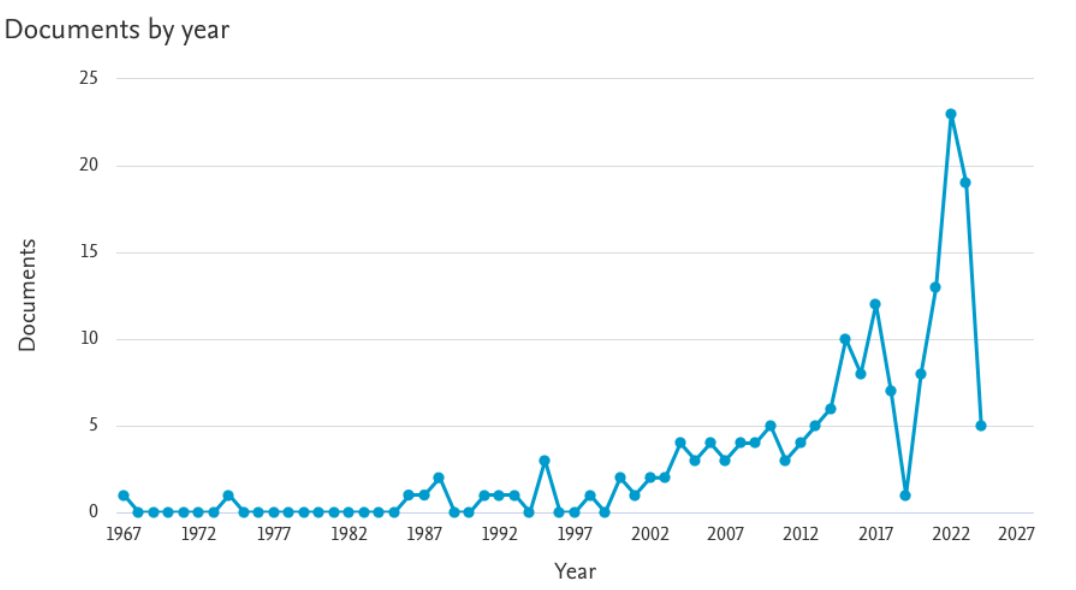


Figure 3.1: Number of published articles or reviews by year

Several authors have studied the literature around disruption management in the past: Early reviews of state-of-the-art include Clarke (1998) and Filar et al. (2001). Clarke (1998) summarized the state-of-the-art in AOCC's during disruptions, describing the structure of an AOCC, its information systems and decision support systems. In addition to a literature review, they proposed a new decision support framework for aircraft recovery. Filar et al. (2001) reviewed disruption management from the airports' perspective. Although not exactly literature reviews, Kohl et al. (2007) and Fogaça et al. (2022) described the airline disruption management process in a more practical sense. Where Fogaça et al. (2022) emphasised the decision making process during disruption and the workflow, while Kohl et al. (2007) described the airline planning and DM process as background information to report on the development of the DESCARTES project, a multiple resource decision support system funded by the European Commission. Clausen et al. (2010) compared multiple studies on airline recovery from a modelling perspective, comparing different network representations and functional aspects for aircraft, crew, and integrated methods. Hassan et al. (2021) expanded on this review by structuring the publications on airline recovery from 2009-2019 in a similar fashion. Su et al. (2021) reviewed basic models and previously proposed extensions for aircraft, crew, and integrated recovery, providing a complete picture of airline recovery for multiple disruption scenarios. The most recent reviews are from Santana et al. (2023) and Wu et al. (2024). Santana et al. (2023) performed a systematic review of publications on ARP's much like Clausen et al. (2010) and Hassan et al. (2021), comparing different problem features, objectives and solution methods. Wu et al. (2024) reviewed airline recovery publications from a different perspective, highlighting the problem from a passenger- and cargo oriented viewpoint as opposed to focusing more on aircraft and crew.

As explained in Chapter 2, the airline recovery process can be divided into a few sequential recovery phases for aircraft, crew, and passengers respectively. Most papers on ADM are primarily focused on aircraft recovery, as this is the most important resource for airlines and form the first step in the decision process (Wu et al., 2024). However, with better computational abilities and improved state-of-the-art, a shift to methods including two or more resources can be enabled. Hassan et al. (2021) found that 50% of publications from 2010-2020 that propose solution models for the ARP focus on recovering multiple resources simultaneously, which indicates the sentiment to explore and develop more sophisticated methods. Santana et al. (2023) also concluded that methods developed in recent decades tend to focus more on the integration of resources as a step forward to attaining practical applicability of these methods.

As mentioned, this review will cover published documents on disruption management from an aircraft recovery perspective. For structure, the publications are categorized and grouped by the resources they address (i.e. Aircraft, Aircraft and Crew, Aircraft and Passenger, and Aircraft, Crew and Passenger). Comparisons will be done mainly on disruption types, recovery actions, objectives, and solution methods. Because maintenance requirements form an important constraint for airlines, publications taking into account these requirements are also distinguished in a separate subsection. Moreover, publications addressing the dynamic and uncertain nature of ADM are grouped and reviewed, to get an understanding of what is done and what can be done in the area of Anticipatory Disruption Management.

3.1.1. Aircraft Recovery

Unsurprisingly, aircraft recovery is the most extensively researched problem in ADM literature. In aircraft recovery, the recovery decisions are solely made on inputs regarding the airlines fleet, without considerations of crew and passengers. The first to tackle the aircraft recovery problem were Teodorovic and Guberinic (1984), who developed a method that minimizes total passenger delays by delaying flights or swapping tail-numbers. Teodorovic and Stojkovic (1990) expanded this method by incorporating airport curfews and flight cancellations.

Thengvall et al. (2000) were the first that approached the problem in a way that allows for minimal deviation from the initial schedule in a flexible model with room for user preferences in the objective. The work was expanded in Thengvall et al. (2001), where hub closures and multi-fleet scenarios were considered. In Thengvall et al. (2003), a similar problem was modelled as a multicommodity network model and the algorithmic design for solving the model was presented, consisting of heuristic techniques for finding feasible solutions from earlier found relaxed solutions.

In most ARP's, the flight networks are modelled either as time-space (or time-line), connection or time-band network and are explained in detail in the literature review of Clausen et al. (2010):

"The idea of a time-line network is to represent the possible schedules in a natural way from the time-and-station point of view, which is not possible when using a connection network. A time-line network has a node for each event, an event being an arrival or a departure of an aircraft at a particular station. Time-line networks are activity-on-edge networks, where directed edges correspond to activities of an aircraft, and schedule information is represented explicitly by the event nodes."

And:

"A connection network is an activity-on-node network,where flight legs correspond to nodes in the network and connections between flight legs correspond to directed edges (arcs) between the nodes. A flight leg is given by its origin, destination, departure time and date and arrival time and date. In addition, there is a set of origin and destination nodes indicating possible positions of aircraft in a fleet at the beginning and at the end of the planning horizon, respectively."

The schematic depictions of a connection and time-space network for an example flight schedule with 3 aircraft and 4 airports is given in Figures 3.2 and 3.3

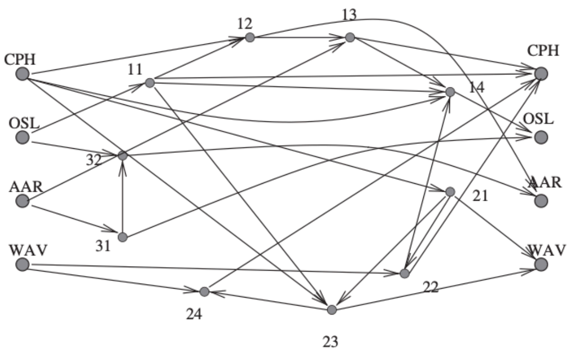


Figure 3.2: Connection network with 4 airports and 3 aircraft, from Clausen et al. (2010)

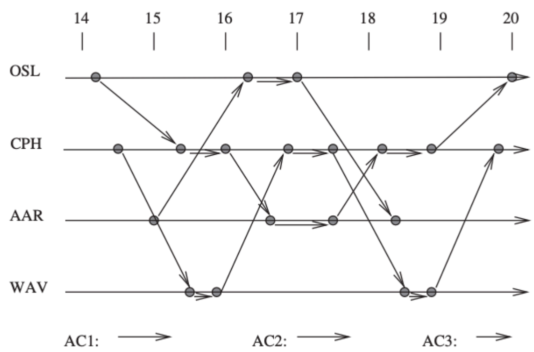


Figure 3.3: Time-space network with 4 airports and 3 aircraft, from Clausen et al. (2010)

Up until 2001, the conventional representations for airline recovery models were connection networks and a time-space Network. Bard et al. (2001) were the first to use a Time-Band Network representation introduced by Arguello (1998). A time-band network is a time-space network where the ground arcs are modelled as activity nodes of a certain time span (band), this reduces the models' size as multiple activities can be represented as a single node, where in a time-space representation at least two nodes and a edge are needed.

Using a connection network, Rosenberger et al. (2003) aimed to minimize cancellation and re-routing costs while considering realistic airport constraints. They developed a heuristic selection algorithm to determine which aircraft to select as potential swaps for disrupted aircraft, which was proven to reach fast solutions to real-sized problems.

Eggenberg et al. (2010) introduced the problem as a constraint-specific model where each resource has its own recovery network in a time-band representation. They divided the models' constraints in unit-specific and structural constraints. Up until then, maintenance constraints were not included in most aircraft recovery frameworks. The unit-specific constraint model allowed easy consideration of these maintenance constraints, however. They illustrated this concept by solving the aircraft recovery problem using a Column Generation (CG) algorithm that combines individual recovery schemes into one scheme that satisfies structural constraints. In addition to aircraft swaps, flight delays and flight cancellations, they also considered maintenance swaps, but considered airport closures as the only disruption source.

Vos et al. (2015) solved the problem dynamically by solving for new disruptions while building on the solution of previous disruptions, they developed the Disruption Set Solver (DSS) that uses a selection algorithm for selecting the subset of aircraft to use in the solution. In a case study, the DSS solved most instances within 10 minutes. Comparing the dynamic approach with a static approach, they showed that actual disruption costs are underestimated in a static framework.

Hu et al. (2017) solved the problem with three conflicting objectives: minimization of deviation from the flight schedule, minimization of the maximum flight delay, and minimization of the number of aircraft swaps. They developed a combined ϵ-*constraint* and neighbourhood search heuristic to enable the method to solve large instances. Inclusion of number of aircraft swaps and deviation from the original schedule in the objective made sure that recovery solutions implicitly followed airline preferences. Theoretically an optimal solution could involve many swaps and large deviations, while this is practically undesired by airlines.

In a series of publications (Wu et al. (017a), Wu et al. (017b), Wu et al. (017c)), addressed the aircraft recovery problem. In Wu et al. (017a) they addressed long-haul airline disruption as a result of aircraft unavailability. They adopted an iterative fixed-point approach from Dang and Ye (2015) and introduced two division methods to divide the solution space into independent segments and compute the solution in a distributed way. This method required less (partial) feasible routes to be generated than with an normal Linear Program (LP) solver, and fast solutions could be generated. They expanded on this method in Wu et al. (017b) where they took multiple fleets into account. In Wu et al. (017c), they used the same method to solve for disruptions due to airport closures.

Liang et al. (2018) proposed a model for recovery in reduced airport capacity scenarios. In addition to swapping, delaying or cancelling flights, they included maintenance task swaps to the recovery actions and solved the problem with a CG generation algorithm. They found that swapping maintenance tasks can significantly reduce recovery costs with 20%-60% dependent on the instance. Khaled et al. (2018) proposed a multi-objective framework to schedule recovery in the tail-assignment problem when considering long term disruptions (in the order of days). An ϵ-constraint method is used to find Pareto-optimal solutions for the different objectives.

Hondet et al. (2018) was one of the first to leverage an RL method as a solution method in a framework where only aircraft swaps are considered. Although the results were far from optimal, the method performed better than an idle scenario in which no actions are taken. In another ML approach, Hassan (2019) developed a decision support system for aircraft recovery that delays, cancels and swaps flights. An Integer Linear Program (ILP) model is used to find solution to the problem, and a Random Forest Classifier is used to select a sub-network of the problem consisting of relevant aircraft to be included in the solution, improving computation times by 45% on average over an exact method.

Lee et al. (2020) addressed the dynamic and uncertain nature of disruptions by proposing an reactive-proactive model that incorporates partial and probabilistic forecasts based on a queuing model to represent airport congestion. They classified delays into three categories: Systemic, contingent and propagated. Where systemic delays are due to congestion, contingent delays are due to unforeseen events, and propagated delays are due to downstream effects in the schedule. They found that using these partial predictions yields better results compared to a a myopic baseline, ultimately reducing expected disruption costs without creating additional risk in airline recovery.

Lee et al. (2022) adopted the time-space network mathematical formulation for the ARP from Thengvall et al. (2001). In an Reinforcement Learning approach, they modeled this ARP as a Markov Decision Process (MDP) and used Q-learning and Double Q-learning algorithms to train the model to make good recovery decisions given the scope of their model. This consisted of aircraft delays and swaps as recovery actions, while the model followed three objectives: Total flight delays, number of flight delays longer than 30 minutes, and number of flight delays longer than zero minutes. A case study on a South Korean airline was done with data consisting of four airports, 70 flights, and seven aircraft of a single fleet. The results were compared to other solution approaches from literature and the RL method showed superior performance in terms of computation time. The author mentions that unnecessary aircraft swapping was not modelled to have any negative effects, although it can have these negative effects in real operations, and should therefore be considered to prevent the model from scheduling too many swaps like Hu et al. (2017) has done in their framework.

Huang et al. (2022) presented the problem in a Time-Space network that relies on copies of flight arcs to generate new solutions. A cost-driven copy evaluation method was designed that evaluates potential copies to be incorporated in the solution on their quality and disregards ”bad” copies to limit computational effort and reduce the problem size. The effectiveness of this method was verified by comparing solution quality and computational time to other methods. In addition to conventional recovery actions, they incorporated retiming options for planned maintenance tasks. Wang et al. (2019) found that an Integer Program was not flexible enough to consider complex constraints and uncertainties. They presented a simulation-based approach to correspond with existing operations of a Chinese airline and demonstrate the potential of simulation-based approaches. Rhodes-Leader et al. (2022) addressed the uncertain nature of disruption management by formulating a deterministic MIP (Mixed Integer Program) of which the solutions are improved by a simulation optimization procedure that accounts for uncertainty, building forth on their research on multi-fidelity networks (i.e. both deterministic and uncertain) in Onggo et al. (2018). They considered multiple disruption sources and found that the combination consistently finds good solutions to the ARP, with the simulation optimization providing improvements over the initial deterministic solutions.

Zhao et al. (2023) accounted for implicit costs of performing recovery actions by incorporating protection arcs in the Time-Space network to consider effects of gate reassignment, crew rescheduling, maintenance requirements, passenger dissatisfaction, and generally keep aircraft on their originally scheduled routes. They captured the uncertainties of the disruptions in a two-stage approach with a rolling horizon window of one hour by developing different solutions as a function of the uncertain disruption parameters. This method could find (near-)optimal solutions in less than a second for a range of scenarios.

Zang et al. (2024) highlights the possibility of predicting disruptions and taking a proactive attitude towards disruption management, examining the aircraft recovery problem from a demand-and-supply balance across parallel time-space networks of aircraft, as proposed by Vink et al. (2020). They classify disruptions into three types as done in Lee et al. (2020): *propagated, systemic, and contingent*. They apply former predictive studies on spatiotemporal distributions of airport demand-supply to the ARP. Following a case-study with a Chinese airline, results indicate that their method effectively reduces airline delays and operating costs in actual operations.

In Table 3.1, an overview of some of the aforementioned papers is provided where they are distinguished between their disruption characteristics, recovery actions, characteristics, objectives, solution methods and recovery horizon. It can be seen that the most common recovery actions for aircraft recovery are aircraft swaps, flight delays and flight cancellations. Some papers also include maintenance-related recovery actions (Eggenberg et al. (2010); Liang et al. (2018); Huang et al. (2022)). Curfew violations considerations are included in the most recent publications (Rhodes-Leader et al. (2022); Huang et al. (2022); Zhao et al. (2023)). Most papers aim to minimize recovery or delay costs, some of which proposed a multi-objective model where costs were minimized in combination with the number of delays, number of cancellations and deviations from the schedule (Thengvall et al. (2001); Wu et al. (017b); Rhodes-Leader et al. (2022); Zhao et al. (2023)). Others focused only on non-economical KPI's such as a combination of schedule deviations, maximum delay, and number of aircraft swaps (Hu et al., 2017). Lee et al. (2022) used on time performance as a KPI by aiming to minimize total delay, number of delays, and number of delays over 30 minutes. Commonly, a horizon of one day of operations is used.

Table 3.1: Overview of publications covering Aircraft Recovery

Paper	Disruptions					Recovery Actions					Characteristics			Objectives	Method	Horizon
	AU	AC	RAC	DL	CL	SW	DL	CL	MSW	MDL	Maint.	Curfews	Proactive			
Thengvall et al. (2000)		✓					✓	✓						max. profit	CPLEX	
Thengvall et al. (2001)		✓					✓	✓						max. profit, min CL, min. DC	CPLEX	
Thengvall et al. (2003)		✓					✓	✓						max. profit	Bundle Algorithm	
Eggenberg et al. (2010)		✓				✓	✓	✓	✓		✓			min. RC	CG	1-7 days
Vos et al. (2015)					✓	✓	✓	✓						min. RC	Aircraft Selection Algorithm	1 day
Hondet et al. (2018)				✓		✓								min. DC	RL	1 day
Hu et al. (2017)	✓					✓	✓	✓				✓		min. SD, min. max. delay, min # swaps	ϵ-constraint NSH	
Wu et al. (017a)	✓					✓	✓	✓						min. RC	DFPI-IP	
Wu et al. (017b)	✓					✓	✓	✓						min. RC, min. SD	DFPI-IP	
Wu et al. (017c)		✓				✓	✓	✓						min. delay	DFPI-IP	
Liang et al. (2018)			✓			✓	✓	✓	✓		✓			min. RC	CG	4 days
Hassan (2019)		✓		✓	✓	✓	✓	✓						min. RC	ML-selection	1 day
Lee et al. (2020)		✓				✓	✓	✓					✓	min. RC	DP	1 day
Lee et al. (2022)		✓				✓	✓							min. delay, min. # delays, min. # delays > 30 minutes	RL	1 day
Huang et al. (2022)		✓		✓	✓	✓	✓	✓		✓	✓	✓		min. RC	ICD-CG	hours- days
Rhodes-Leader et al. (2022)	✓		✓				✓	✓			✓	✓		min. SD, min. DC	Simulation	1 day
Zhao et al. (2023)	✓					✓	✓	✓			✓	✓		min. SD, min. RC	RH	1 day
Zang et al. (2024)			✓	✓		✓	✓	✓			✓		✓	min. RC	DDB Heuristic	1 day

**AU: Aircraft Unavailability, AC: Airport Closure, RAC: Reduced Airport Capacity, DL: Flight Delays, CL: Flight Cancellations, SW: Aircraft Swap, MSW: Maintenance Swap, MDL: Maintenance Delay, DC: Delay Costs, RC: Recovery Costs, SD: Schedule Deviation, CG: Column Generation, RL: Reinforcement Learning, NSH: Neghbourhood Search Heuristic, DFPI-IP: Distributed Fixed Point Iterative-Integer Program, ML: Machine Learning, DP: Dynamic Programming, ICD-CG: Iterative Cost Driven-Copy Generation algorithm, RH: Rolling Horizon, DDB: Decision-Decomposition-Based Heuristic*

3.1.2. Aircraft and Passenger Recovery

In this section, publications that cover aircraft and passenger recovery are reviewed and discussed. Aircraft and passenger recovery involves aircraft recovery where additional considerations for revenue streams originating from passengers are made. Also, passenger experience is often considered in the decision making process.

As part of the 2009 ROADEF Challenge, Bisaillon et al. (2010) developed a Large Neighbourhood Search (LNS) heuristic that iterates between destroying and repairing the solution in three phases: construction, repair and improvement. While considering multiple disruption sources, their objective was to minimize operating costs, passenger dissatisfaction costs (as a function of the total delay), and aircraft inconsistency costs when an aircraft is not at its designated airport after the day of operations. In addition to common recovery decisions, they considered flight creation as a recovery action. They won the 2009 ROADEF challenge with this work. Sinclair et al. (2014) improved the LNS heuristic of Bisaillon et al. (2010) with the aim of enhancing algorithmic performance, on which they extended in Sinclair et al. (2016) by applying a CG post-optimization heuristic after the LNS to improve it further.

Also in the context of the ROADEF 2009 challenge and in the same disruption scenarios as in Bisaillon et al. (2010), Jozefowiez et al. (2013) developed a three-stage heuristic that first finds a new feasible plan by canceling flights and itineraries, then reassigns passengers to existing rotations, and subsequently create flight legs to accommodate the remaining unassigned passengers. Zhang et al. (2016) later extended this model and bench-marked it against data from the ROADEF Challenge. Opposed to previous work, they considered only flight cycles comprising exactly two flight legs and interchanged the second and third stage of the LNS heuristic of Bisaillon et al. (2010).

Hu et al. (2015) proposed a multi-fleet aircraft routing and passenger transiting optimization model, where necessary conditions are given for the existence of feasible solutions to the network model for a given set of practical recovery options. The objective is to minimize costs and solutions can be adjusted afterwards to reflect actual costs if necessary. In Hu et al. (2016), they proposed a new approach to the problem by designing a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic. The goal is to find the optimal trade-off between passenger delay cost, passenger reassignment cost and the cost of refunding tickets. Both studies were tested on real data of a Chinese airline and quality solutions where found within 100s for many instances, they showed superior performance compared to the current methods of the airline in which they implemented the case study.

Santos et al. (2017) presented an ILP approach. To guarantee the linearity of the optimization model and fast computational times, a receding horizon framework is used with a horizon window of 1,5 hours. The objective is the minimization of fuel cost, passenger compensation, and passenger inconvenience costs. To address the fact that many hub airports are heavily congested during peak hours, different airport capacity constraints are included and the model is tested on real-data from a hub-and-spoke carrier. This approach led to 29% costs reductions.

Marla et al. (2017) included the innovative recovery action of flight planning in both exact and approximate mathematical model. Cruise Speed Control (CSC) would allow for longer flying times in combination with less fuel burn, which can be leveraged in disruptive scenarios. The exact model was intractable but an approximate model was found to reduce total recovery costs within a two minute time limit. They showed that their approach could reduce the recovery cost of airline and passengers by also considering passengers' willingness.

Vink et al. (2020) extended the dynamic approach of Vos et al. (2015) by solving for disruptions as they happen and as new information becomes available. They also considered connecting passengers' recovery itinerary operations without compromising the computation time of the model. For this, a costs estimation model was designed to estimate the delay experienced by passengers, the costs of these delays and the itineraries followed by passengers, without adding additional constraints or decision variables in the optimization model. Similar to Vos et al. (2015), the dynamic approach was used to demonstrate that using a static approach generally underestimates disruption costs.

Hu et al. (2021) constructed an integer programming model that minimizes airline recovery costs and passenger recovery loss. The first representing short-term economic costs of the airline. The second representing loss of the impact on passengers' life and business work due to the itinerary changes and long-term reputation loss for airlines. A heuristic combined with multi-directional and stochastic Variable Neighborhood Search (VNS) algorithm is designed to solve the problem and test it on real data.

Yetimoğlu and Aktürk (2021) included seat capacity limitations in the evaluation of each passengers' itinerary to calculate cancellation costs in a math-heuristic method. By doing so, they provided a better financial cost estimate for disrupted passengers and a better trade-off between operational and passenger related recovery costs. They also adopted the CSC decision variable in their framework while specifically addressing aircraft unavailability as disruption source. Sun et al. (2022) modified the time-band representation so that a large number of redundant flight arcs and infeasible recovery flight arcs can be eliminated. They developed a method that can systematically generate passenger candidate itineraries and incorporate them into an optimization framework. Furthermore, they introduced the intermodal concept that considers downstream effects to the ARP. They showcased their methods ability to reduce the complexity of the MILP and demonstrated that considering intermodal recovery options can drastically reduce the number of disrupted passengers.

Cadarso and Vaze (2022) included passenger response behaviour in a model that endogenizes the impacts of recovery decisions on passengers. They developed an optimization model with exact linearization for non-linear passenger costs terms and delayed constraints for aircraft maintenance feasibility. Testing was done with data from a major European airline and the results were promising, showing that incorporating passenger behaviour in the model yields benefits for airline recovery performance.

Wandelt et al. (2023) evaluated node importance in flight network representations to assess the role of airports in airline networks. They proposed a mixed-integer program formulation for airline recovery baseline under node disruptions and designed a (VNS) heuristic to compute solutions. They showcased a method for solving the recovery problem as well as assessing the robustness of an airline network, which was a novel insight for the field of ADM.

Similar to Hu et al. (2017) and Khaled et al. (2018), Chen et al. (2023) focused on passenger and airline preferences in a bi-objective optimization framework. An ϵ-*constraint* method was used to find Pareto optimal solutions. Using a Genetic Algorithm they showcased efficient results.

In Table 3.2, an overview of some of the discussed papers covering aircraft and passenger recovery is presented. Compared to aircraft recovery, delays are predominantly incorporated in aircraft and passenger recovery models as disruption source. Also, all publications used recovery cost or profit in their objective as passenger related objectives usually involve monetary measures (delay compensation, missed revenue, rebooking costs). Flight could be created as recovery option to accommodate passengers (Bisaillon et al. (2010); Jozefowiez et al. (2013); Sinclair et al. (2014); Sinclair et al. (2016); Zhang et al. (2016)) and CSC was considered only by Marla et al. (2017) and Yetimoğlu and Aktürk (2021). Passenger behaviour and preferences were captured in the works of Marla et al. (2017), Cadarso and Vaze (2022), Hu et al. (2017) and Chen et al. (2023) to show the importance of including individual preferences and behaviour into the decision process.

Table 3.2: Overview of publications covering Aircraft and Passenger Recovery

Paper	Disruptions					Recovery Actions						Characteristics		Objectives	Method	Horizon
	AU	AC	RAC	DL	CL	SW	DL	CL	PR	FC	CSC	Maint.	Curfews			
Bisaillon et al. (2010)	✓			✓		✓	✓	✓	✓	✓		✓		min. RC	TS-LNS	36-52 h
Jozefowiez et al. (2013)	✓		✓	✓			✓	✓	✓	✓		✓		min. RC	TS-LNS	1-3 days
Sinclair et al. (2014)	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓		min. RC	TS-LNS	14-78 h
Sinclair et al. (2016)	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓		min. RC	TS-LNS	14-78 h
Zhang et al. (2016)	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓		min. RC	TSH	1-3 days
Hu et al. (2015)	✓					✓	✓	✓	✓				✓	min. RC	CPLEX	1 day
Hu et al. (2016)	✓			✓		✓	✓	✓	✓			✓	✓	min. RC	GRASP	1 day
Santos et al. (2017)			✓	✓			✓		✓					min. RC	RH	1 day
Marla et al. (2017)				✓		✓	✓	✓	✓		✓	✓		min. RC	RH	1.5 days
Vink et al. (2020)	✓	✓		✓	✓	✓	✓	✓				✓		min. RC	Selective-ILP	1 day
Hu et al. (2021)				✓	✓	✓	✓	✓	✓				✓	min. RC	MDS-VNS	1 day
Yetimoğlu and Aktürk (2021)	✓					✓		✓			✓			max. profit	Math-heuristic	1 day
Sun et al. (2022)	✓					✓	✓	✓	✓					min. RC	CPLEX	
Cadarso and Vaze (2022)		✓	✓	✓	✓	✓						✓		min. RC	CPLEX	1-3 days
Wandelt et al. (2023)		✓		✓	✓	✓	✓	✓	✓			✓		min. RC	HVNS	1 day

**PR: Passenger Reaccommodation, FC: Flight Creation, CSC: Cruis Speed Control, TS-LNS: Three-Stage Large Neighbourhood Search, TSH: Three-Stage Heuristic, GRASP: Greedy Randomized Adaptive Search Procedure, ILP: Integer Linear Program, MDS-VNS: Multi-Directional Stochastic Variable Neighbourhood Search*

3.1.3. Aircraft and Crew Recovery

Aircraft and crew recovery is a less researched field than the aircraft and passenger recovery problem. According to Hassan et al. (2021), only four papers were published up until 2021, in addition to Abdelghany et al. (2008), which was not included in the review. However, in recent years, (2021-2024), the number of publications that address this problem has doubled. Aircraft and crew recovery involves making decisions on aircraft recovery while considering crew related factors such as labour constraints and crew costs.

Abdelghany et al. (2008) was the first to address the problem by developing a tool named DSTAR. A rolling horizon framework was used in which a schedule simulation and optimization model were integrated. No horizon window was specified however. They aimed to minimize costs by mitigating flight delays and cancellations. Their tool was found to obtain efficient solutions within one minute when tested on an airport Ground Delay Program (GDP) of a US-carrier. Le and Wu (2013) reused their previous work to include crew in the recovery while minimizing assignment cost associated with delays and cancellations. Maintenance requirements and union regulations are considered in this work, and the problem was solved by means of an iterative tree growing method that relies on node aggregation to simplify routings and decrease computation time. The model was tested on data from a Chinese airline, but no computation times were specified in their results.

Maher (2015b) used Column- and Row-Generation (C&GR) to extend on existing generic methods as an alternative to Benders' Decomposition. Integer optimal solutions were identified with Branch-and-Price, which was also integrated into the method. An extensive set of recovery actions was used with route generation, crew duty generation, crew deadheading, crew reserves, and delays and cancellations. However, they did not include aircraft swapping.

Zhang et al. (2015) proposed a two-stage heuristic method where the two stages consisted of an aircraft recovery model with partial crew considerations and vice-versa. A multi-commodity model was used for crew recovery and new constraints are incorporated for the aircraft recovery to ensure feasibility. The method was proven to outperform benchmark algorithms.

Similar to Hassan (2019), Eikelenboom (2022) used a ML-based selection algorithm named LambdaMART, to rank the most relevant resources to be used in the optimization model. The model was extensive in terms of recovery options and could recover the schedule by delaying, cancelling flights, swapping aircraft, and swapping, deadheading or using reserve crew. The model was tested on real flight data from Delta Airlines and was found to obtain promising results, indicating the potential of ML methods.

Liu et al. (2023) considered costs incurred due to long crew connections. A method was proposed that delayed flights in order to save working hours of crew by enabling crew rest, where a trade-off between operational costs and crew costs was made. An arc-based integer programming model and a set partitioning model with a CG algorithm was proposed that showed to be efficient and effective. Similarly, Eshkevari et al. (2023) proposed a bi-objective model that includes crew rest and sit times. Khiabani et al. (2022) modelled an ILP based on individual flight legs and Benders' Decomposition was used to solve the model. They considered crew swapping, deadheading, aircraft swapping, flight delaying and cancellation as recovery actions. Also ground and sit time requirements for crew were incorporated in their framework.

Zhong et al. (2024) introduced a multi-priority aircraft and crew recovery problem where priority could be given to flights, like flights carrying first-aid items. A particle swarm optimization was done with three repair schemes to prevent low feasibility. This approach provided a new model construction framework for solving the aircraft and crew recovery problem. In addition, their framework can also provide series of recovery schemes to allow for decisions support for airline managers based on their preferences.

Looking at Table 3.3, it can be seen that there is a lot of variation between the publications in term of disruption sources. Le and Wu (2013) and Khiabani et al. (2022) stand out by considering crew related disruptions in their framework. They both only consider DH as crew-related recovery action however. Eikelenboom (2022) proposed the most extensive model in terms of recovery actions, but did not incorporate operational constraints relating to maintenance and curfews. Similar to the publication addressing aircraft and passenger recovery, all publications in the paper considered recovery costs in the objective. This does not come as a surprise, since crew recovery decisions usually are dependent on the compensation schemes of crew (besides labour constraints), and the impacts of aircraft and crew recovery decisions can be translated into one KPI: financial costs.

Table 3.3: Overview of publications covering Aircraft and Crew Recovery

Paper	Disruptions						Recovery Actions							Characteristics		Objectives	Method	Horizon
	AU	AC	RAC	DL	CL	CUA	SW	DL	CL	DH	RC	CSW	CSC	Maint.	Curfews			
Abdelghany et al. (2008)		✓	✓				✓	✓	✓	✓	✓			✓		min. RC	RH-GO	1 day
Le and Wu (2013)	✓					✓		✓	✓	✓				✓		min. RC	ITG-NG Heuristic	1 day
Maher (2015b)		✓						✓	✓	✓	✓			✓		min. RC	C&RG	1 day
Zhang et al. (2015)	✓	✓					✓	✓		✓				✓		min. RC	TSH	1 day
Eikelenboom (2022)	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓				min. RC	ML-selection	1 day
Liu et al. (2023)				✓			✓	✓	✓	✓					✓	min. RC	CG	1 day
Eshkevari et al. (2023)				✓				✓	✓	✓				✓		min. RC	TS	
Khiabani et al. (2022)						✓	✓	✓	✓	✓	✓					min. RC	BD	1 day
Zhong et al. (2024)			✓					✓	✓		✓	✓	✓			min. RC	PSO	4 days

**CUA:Crew Unavailability, DH: Deadheading, RC: Reserve Crew, CSW: Crew Swap, RH-GO: Rolling Horizon with Greedy Optimization, ITG-NG: Iterative Tree Growing with Node-Combination, C&RG: Column- and Row Generation, TS: Tabu-Search, BD: Benders' Decomposition, PSO: Particle Swarm Optimization*

3.1.4. Aircraft, Crew and Passenger Recovery

Integrated recovery frameworks are the most complex both in terms of formulation and computational complexity. Many constraints must be considered for aircraft, crew and passengers to be able to effectively formulate an integrated model. Besides, many of these constraints are specific to airlines and/or countries/regions. This increases the difficulty of formulating an extensive integrated model that can be generalized to many different problem instances. Several efforts have already been made to tackle the integrated recovery problem, while recent technological advancements increase the capabilities of theoretical frameworks to be used in practice.

The first to present a fully integrated approach was Lettovsky et al. (1997). Three sub problems were formulated in a decomposition scheme that were controlled by one master problem. The three sub problems are solved sequentially while retaining feasibility, representing the recovery process in practice as it is done today. Only parts of the problem were actually implemented.

Bratu and Barnhart (2006) further attempted to address the integrated problem by simulating a model that delays and cancels flights and assigns reserve crew and aircraft to flight legs. Although the model does not take crew costs into consideration while minimizing recovery costs, they showed that the model could be used as decision making tool and potentially reduce delays without increasing costs.

Petersen et al. (2012) extended the work of Lettovsky et al. (1997) by adopting the sequential solving algorithm method using Benders' Decomposition. Compared to a truly sequential method (i.e. solving the whole problem for aircraft, then crew, then passengers) they showed to achieve superior results. Maher (2015a) used the same C&RG approach as in their passenger recovery study for integrated method. Using a new definition of cancellation variables that model passenger recovery by prescribing the alternative itineraries, they developed a model in that aims for optimality with aircraft and crew recovery while passenger considerations are explicitly modelled to find re-booking options. This approach showed improved computation time over a CG approach and resulted in less disrupted passengers.

In an extensive framework, Arikan et al. (2017) proposed a method that is based on the flow of each aircraft, crew member, and passenger as entity. Almost all common disruption types and recovery actions are included as well as ferrying aircraft, cruise speed control and ticket cancellations. The model allows for approximation and explicit modelling of passenger delay costs. The computation times were too long for practical implementation however.

Evler et al. (2022) addressed the problem as a vehicle routing problem with homogeneous fleet. Passenger and crew itineraries are modelled as links between flights where transfer times were considered as influence on the delay due to longer turnarounds. They included turnaround time estimations to predict the propagated delays, allowing for proactive schedule changes. The links could be broken if there was a feasible alternative and rebooking and compensation costs where still efficient. A rolling horizon algorithm where the window size was determined by the departure and arrival times of flights was used for solving the model. Results indicate that resilience of the recovery network was improved when considering turnaround processes in comparison to individual aircraft recovery and turnaround models.

Ding et al. (2023) performed an extensive study on integrated recovery focusing on different solution methods for large-scale instances. They proposed three solution methods to compare with each other; one exact method, one with a Variable Neighbourhood Search heuristic, and a third VNS guidance method based on Deep Reinforcement Learning (DRL), where the operations of the VNS heuristic are guided by the RL agent. The solution of the optimization model consists of flight strings belonging to entities (aircraft, crew, passengers). The DRL guided VNS then selects modified solutions by swapping, cutting, inserting and deleting flights in the entities' flight strings to find improved solutions. They tested the solution approaches on multiple sizes of test instances and found that, after training, the DRL guided VNS scales well and outperformed the heuristic method while solving a large instance of 931 flights and 279 aircraft in 14 seconds.

As expected from integrated recovery models, most include an extensive set of recovery actions, focusing on aircraft, crew and passengers simultaneously. Evler et al. (2022) used a unique approach to integrate crew and passenger recovery with aircraft recovery by using turnaround decisions. Following the same reasoning as with aircraft & crew passenger and aircraft & crew recovery, it comes natural that all publication use the recovery costs in their objectives. In Table 3.4, the overview of the publications on integrated recovery is shown.

Table 3.4: Overview of publications covering Aircraft, Crew, and Passenger Recovery

Paper	Disruptions						Recovery Actions								Characteristics			Objectives	Method	Horizon
	AU	AC	RAC	DL	CL	CUA	SW	DL	CL	PR	DH	RC	CSW	CSC	Maint.	Curfews	Proactive			
Petersen et al. (2012)			✓				✓	✓	✓	✓	✓		✓		✓			min. RC	BD	
Maher (2015a)		✓						✓	✓	✓	✓	✓	✓	✓	✓	✓		min. RC	C&RG	6 hours
Arikan et al. (2017)	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓			min. RC	CQ MILP	
Evler et al. (2022)	✓			✓			✓	✓	✓								✓	min. RC	RH	1 day
Ding et al. (2023)		✓		✓	✓		✓	✓	✓	✓	✓		✓	✓	✓			min. RC	RL-VNS	1 day

CQ MILP: Conic Quadratic Mixed Integer Linear Program, RL-VNS: RL guided Variable Neighbourhood Search

3.1.5. Maintenance Requirements

A practically applicable recovery framework should incorporate realistic constraints that represent airlines' daily operational restrictions and challenges. An important part of aircraft routing is the scheduling of different maintenance task for individual aircraft. Since routings change during schedule recovery, maintenance feasibility can be disturbed, which would make a recovery model that disregards maintenance useless in a realistic scenario. In this subsection, several published papers that include maintenance feasibility in their recovery model are discussed. Note that all these papers are discussed in prior subsections, in this subsection however, the focus lies on the way maintenance requirements are incorporated into their frameworks.

To begin with Eggenberg et al. (2010), who developed a constraint specific recovery time-band network to include resource specific constraints for aircraft to include maintenance constraints. They compared results for including and excluding maintenance requirements and concluded that considering maintenance is not only necessary to ensure feasibility of the recovery scheme, but the solution can also be significantly improved by reducing delays when maintenance operations are rescheduled.

In their three-stage heuristic, Bisaillon et al. (2010) included a feasibility check for maintenance in the construction phase. If violations occur, loops in the flight sequence are removed to make the rotation feasible again. If that is not possible, the sequence from the critical flight to the end of the rotation is cancelled. The critical flight is then the last flight that would depart from the maintenance airport before the scheduled maintenance and feasibility is restored. Sinclair et al. (2014), Sinclair et al. (2016) adopted this same method.

In Jozefowiez et al. (2013), maintenance tasks are treated as mandatory flight legs that cannot be cancelled in the recovery process. This ensures feasibility as the algorithm relied on cancelling and creating flight legs. Maher (2015a) considered maintenance tasks at the end-of-day and therefore only set constraints to the sink nodes of aircraft at the end-of-day.

Zhang et al. (2016) specified maintenance duration in their framework. They modelled aircraft having maintenance requirements in a separate time-space network, where maintenance arcs are treated as a "special flight arc" going from and to the same airport, similar to regular ground arcs. Marla et al. (2017) used a similar approach to maintenance where each aircraft has a time-space network where planned maintenance is represented as a "artificial flight leg" with the maintenance station as origin and destination. Maintenance can be delayed by creating copies of these arcs. Liang et al. (2018) considered maintenance swaps as planning actions in case multiple airports are allowed as maintenance stations. The tasks are planned with a large enough buffer to the actual maintenance limit in terms of flight hours, cycles and time interval. They also included maintenance costs in the objective.

Also using a separate time-space network representation, Vink et al. (2020) made a distinction between fixed and flexible maintenance tasks. The tasks can be classified by the airlines themselves for individual preference. They were modelled as ground arcs, and at least one of the set of sequential ground arcs for flexible tasks need to be used to guarantee maintenance feasibility, meaning that flexible tasks can be postponed to other available maintenance opportunities, while fixed task must be performed as originally planned. Zang et al. (2024) also adopted the separate time-space representation to account for aircraft specific maintenance constraints. In their framework, maintenance task had to be performed at one of the available stations before the flying limits were reached (flight hours, flight cycles, time interval).

Huang et al. (2022) considered unplanned maintenance as a disruption source resulting in aircraft unavailability, in addition to planned maintenance. They also considered maintenance retimings as recovery action, modelled as copies of maintenance arcs (i.e. special flight arcs, Zhang et al. (2016), Marla et al. (2017)) in a time-space network. Maintenance swaps were not allowed however. They also included maintenance retiming costs in their objective.

In their model, Ding et al. (2023) incorporated maintenance tasks as must-visit nodes in the connection network. Additionally to planned maintenance, unplanned maintenance formed a disruption source. They did not consider flexible maintenance planning but recognized the improvements this could bring to their model.

Overall, maintenance is an important operational constraint that poses a daily challenge for airlines, and it can not be neglected during airline recovery. Most airline recovery papers consider maintenance requirements to some extend. Eggenberg et al. (2010); Liang et al. (2018) and Huang et al. (2022) considered maintenance related recovery decisions, allowing for more flexible models. Vink et al. (2020) defined two classes of maintenance tasks: fixed and flexible, where the airline can define which tasks can be postponed (the flexible tasks) and which cannot (the fixed tasks). Others considered maintenance as hard constraints on the routings by modelling maintenance tasks as mandatory arcs (Bisaillon et al. (2010); Sinclair et al. (2014); Sinclair et al. (2016); Jozefowiez et al. (2013); Zhang et al. (2016); Marla et al. (2017); Ding et al. (2023)), while some only required aircraft to be at their designated maintenance stations at the end of the recovery window after the day of operations (Maher, 2015a).

3.1.6. Anticipatory Disruption Management

Currently, disruption management in done reactively, i.e. acting after disruptions occur. It is assumed that more efficient recovery is expected in cases when potential future disruptions can be anticipated and the potential disruption effects can be proactively mitigated at an earlier stage. There are two sides from which proactive disruption management can be regarded. The first one being in the scheduling phase, where buffers and slack are incorporated in the schedule to make it less vulnerable to disruptions. The other way to view proactive disruption management is by utilization of frameworks where potential future disruptions are anticipated. This could be done in conjunction with forecasts of disruptive events, and dynamic and stochastic processes that characterise them. In this subsection, literature regarding this second aspect view to proactive recovery in ADM is discussed.

Anticipating disruptions by predicting them is extremely hard as disruptions are various in source, type, duration and magnitude. However, historical data can be leveraged to find patterns in disruption data and flight operations/recovery data. Additionally, partial and probabilistic forecasts can be made on certain disruption sources like weather events, airport congestion or equipment health, that translated into flight delays and aircraft unavailability. This is done by Lee et al. (2020) in an aircraft recovery framework where partial and probabilistic forecasts are made based on an stochastic queuing model for airport congestions. The queuing model passed probability distributions as inputs to the optimization model, which is based on the model from Marla et al. (2017), this is done by constructing stochastic parallel time-space networks. The model is solved in a rolling horizon approach with one-hour windows, but cannot be solved directly via backwards induction because the problem size becomes intractable. They used an look-ahead approximation and sample average approximation method to tackle this, and by Monte Carlo sampling, the future value of taking a decision at a certain time step is approximated. It therefore aims to minimize the expected recovery costs based on the probabilistic inputs. The results were compared to a myopic baseline solution (i.e. where potential future disruptions are not anticipated) and that the model yields significant improvements over the myopic solution.

Furthermore, Evler et al. (2022) focused on uncertainty in the turnaround process. They proactively estimated delay propagation by combining a heterogeneous vehicle routing problem with a resource-constrained project schedule problem applied to the turnaround process. With this approach, flight-specific delay cost functions could be calculated and substantial dependencies about the time of the day, the number of succeeding flight legs and particular downstream destinations where identified, these dependencies could be used to proactively mitigate delays.

Zang et al. (2024), like Lee et al. (2020), included predictions on disruption probabilities, particularly reduced airport capacity, in a recovery framework that dynamically reschedules the disruptions. Based on delay probabilities, they formulate expected costs associated with recovery decisions, which they use in their objective function. Following a case-study with a Chinese airline, results indicate that their method effectively reduces airline delays and operating costs in actual operations.

Other papers do not address anticipatory disruption management directly, but propose methods that can be used in combination with anticipatory models (Zhao et al. (2023);Ogunsina et al. (2019); Ogunsina et al. (2021); Ogunsina et al. (2022)). Zhao et al. (2023) addressed two uncertainties regarding disruptions: Disruption duration, and time that the disruption duration becomes known. Their aim was to develop different recovery strategies as a function of the (unknown) disruption length, such that the recovery scheme could be easily modified with the arrival of new information. Although this is not strictly a proactive approach, a comprehensive scenario analysis indicated which aircraft or flights are likely to be affected when certain disruptions occur. This analysis could allow for the employment of proactive strategies.

Ogunsina et al. (2019) focused on discovering patterns in historical flight- and disruption data such that it can be leveraged for use in anticipatory mechanisms. In particular, the adoption of Hidden Markov Models (HMM's) for learning temporal patterns from historical data. A framework is described that models the uncertainty in operations recovery by capturing the manner in which the schedule changes with respect to recovery decisions made by the AOCC operators. Ogunsina et al. (2022), extends Ogunsina et al. (2019) by implementing an Uncertainty Transfer Function Model (UTFM) for managing disruptions in airline operations. The UTFM is based on the models described in Ogunsina et al. (2019) and are implemented and assessed with real-world data in this paper. In Ogunsina et al. (2021), they focused on Exploratory Data Analysis for scheduling and operations. They highlight ML methods to analyze the data and reveal important features to predict disruptions. These kind of works could be used in conjunction with anticipatory recovery models.

In addition to increase in attention to anticipatory disruption management methods in recent years, it is worth mentioning that this aspect of DM is worth exploring according to many researchers. In recent literature reviews, it is noted that, with advancements in data analytics, anticipatory mechanisms could enhance recovery performance (Hassan et al., 2021). Also, solutions are recommended that handle the uncertainty and inherent dynamics of the problem (Santana et al., 2023), which would allow a more proactive approach. Wu et al. (2024) also recommends integrating a reactive-proactive attitude towards disruption management.

3.2. Reinforcement Learning in Operations Management

The fact that most papers address the highly dynamic and uncertain in nature ARP with static and deterministic solution methods, raises the question if there are other solution methods that would yield better results by capturing these dynamics and uncertainties of the problem. By the dynamics of the ARP, we understand the changes that occur *during* the recovery process such as updated information, additional imposed constraints, or changes in the characteristics of disruptions. These dynamics go hand-in-hand with uncertainty, as changes these potential changes that can occur during the process may or may not happen, unknown to the AOCC controller in a practical setting, or to the recovery model in a more theoretical setting.

Some efforts have been made in ADM, as well as in other Operations Management problems, to utilize RL as tool to optimally solve dynamic and uncertain problems. RL is a ML subfield that relies on an *agent* that interacts with an *environment* to learn how to make optimal decisions. The *agent* performs *actions* to go from one *state* to another subsequent state in an *environment* that feeds back a certain *reward* reflecting the quality of the action taken and the corresponding state-transition. These processes rely on a Markov Decision Process (MDP). An MDP is way to describe a sequential decision making problem in which uncertainty of transitioning between states in described (Heinold, 2024). RL is an effective way to solve MDP's.

For every action, the agent receives a reward and tries to maximize the total accumulated reward during a training *episode* consisting of a *trajectory* of many actions-state transition pairs. By following a certain *policy*, and updating this policy after each episode based on the accumulated reward, the agent can eventually learn an optimal policy that maximizes the total reward. This policy is nothing more than a function that maps states to actions.

Three collections of RL algorithms that can be used to solve MDP's can be specified. *Dynamic Programming* (DP), *Monte Carlo* methods (MC) and *Temporal Difference* methods (TD) (Sutton and Barto, 2018). DP is a model-based method, i.e. it requires perfect knowledge of the environment. MC methods do not require full knowledge of the environment and are therefore model-free. They rely on sampling from past actions, states and rewards and therefore learn from experience. TD methods combine DP and MC, by relying on updated estimates based in part on other learned estimates, without waiting for a final outcome (bootstrapping instead of sampling), but without requiring knowledge of the environment.

Motivated by recent success in the application of Reinforcement Learning in Combinatorial Optimization (CO) problems, Mazyavkina et al. (2021) did a survey on recent publications that demonstrate the use of RL algorithms in reformulating and solving several optimization problems. Examples from literature are highlighted and the way they were modelled as MDP's was discussed. Additionally they compared results of the discussed literature to emphasise benefits and shortcomings of different methods. In concluding remarks, they highlight the ability of RL to generalize well to unseen problems, where traditional CO problems can often only be implemented to a concrete set of problems. Also, the potential for improving solution qualities of current heuristic methods is mentioned as an advantage of RL methods.

Bengio et al. (2021) also surveyed literature that leveraged ML methods in an Operations Research (OR) context. They showed how different types of ML can be used for OR problems and to what extend, and at what levels ML methods can be incorporated in OR frameworks. They identified three ways in which learned policies can be combined with optimization models: 1) End-to-End learning, where the models is trained to output solutions directly from the input. 2) Augmented learning, where ML can provide additional information to an algorithm like parameterization or feature selection for the OR algorithm (Hassan (2019); Eikelenboom (2022)). And 3) a parallel framework where the OR or "Master" algorithm repeatedly calls the ML algorithm to make decisions throughout its execution. RL for CO problems can be categorized as such a framework; the OR model determines the possible states, to which the RL agent iteratively alters the states by making decisions until a solution is found. A schematic depiction of this is illustrated in Figure 3.4.

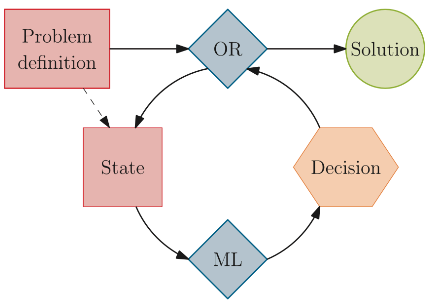


Figure 3.4: Machine Learning alongside Operations Research algorithms, from Bengio et al. (2021)

In the survey, they also discuss some practical challenges in the field, but conclude by mentioning the potential of ML to be the foundation of a new era in Operations Research.

One property of RL is that it can handle complex assumptions, as it is an simulation based method (Lee et al., 2022). Given the fact that RL methods handle complex assumptions as well as uncertain environments, the belief is sparked that RL methods are well suitable for solving ADM problems.

These indications from literature that RL methods are promising in terms of providing efficient and quality solutions for OR problems motivates us to explore the use of these methods in ADM further. In this Section, we will look into literature regarding the application of ML methods, RL in particular, for solving problems in the field of Operations Management. First we will look into general cases where RL is used for some transportation related problems. Then, literature for the use of RL in solving Air Transport Management-related problems is discussed, after which we will zoom in literature on the use of RL in ADM.

3.2.1. General Problems

The use of RL methods has been proven useful in many transportation management related problems in mobility and logistics. In this subsection, literature that addresses transportation/operations management is reviewed, and the way they solve operational problems involving uncertain factors is discussed. How this relates do ADM will become clear later in this section.

Van Heeswijk et al. (2015) studied a Delivery Dispatching Problem with time-windows, where order properties like size, time-window and destination where uncertain. They modelled the problem as an MDP and solved small instances by means of Dynamic Programming (DP), and larger instances with an Approximate Dynamic Problem (ADP) approach. They assumed a known probability distributions of the characteristics of arriving orders and included both deterministic and stochastic orders in their framework. They approximated the Markov model by means of Monte Carlo simulation and used the concept of post-decision states in their implementation, where post-decision state S_t^x is the state immediately after action x_t, but before the arrival of new information ω_{t+1}. Results showed that using ADP to incorporate future information (both deterministic and stochastic) improves dispatching decisions compared to benchmark policies, demonstrating the

Firdausiyah et al. (2019) modelled the behavior of freight carriers and an Urban Consolidation Center operator using Multi-Agent Simulation-Adaptive Dynamic Programming based Reinforcement Learning (MAS-ADP based RL) to evaluate Joint Delivery Systems in an uncertain environment. They compared the MAS-ADP based RL method to Q-learning, a method conventionally used in evaluating the employment of MAS in city logistics. They found MAS-ADP based RL superior in an uncertain environment by adapting better to changes in this environment.

Beirigo et al. (2022) considered an autonomous ridesharing problem in which idle vehicles are hired on-demand in order to meet the service level requirements of a heterogeneous user base, with uncertain demand and idle vehicle supply. In a learning-based optimization framework they proposed an ADP method using Value Function Approximation (VFA), where the dual variables of the underlying optimization model are used to approximate the value functions of having additional vehicles at a certain location at a certain time. The uncertainty of the demand is translated in the uncertainty of the time of the arrival of ride requests. The model considers minimum service-levels such that it is applicable to highly competitive mobility-on-demand service providers, and and experimental study was done on data from New-York City Taxi-cab demand.

Alcaraz et al. (2022) presented a model for decision making in long-distance routes that was capable of making en-route decisions while incorporating driving times, breaks, and rest periods for drivers under uncertain conditions. They used an online model-based RL strategy that included a sample model. Since the model is online, it needed no prior training because learning and planning is done simultaneously and the policy is generated in real-time. The model anticipates future events by generating a set of trajectories (possible sequences of future events). It consists of three subsequent stages: 1) Model Predictive Control (MPC), where the best action within a certain lookahead horizon is decided. 2) A rollout strategy to estimate costs-to-go beyond the lookahead horizon. 3) A Monte Carlo Tree Search (MCTS) to decide how many trajectories (possible sequences of future events) are generated for each action at each stage. They tested their model with data from a Spanish long-distance transport company and compared the results to a baseline policy and to a Deep Q-Networks (DQN), a state-of-the-art model-free algorithm. The method showed superior performance to both in terms of costs reduction, but had longer decision times than DQN, due to the fact that learning and planning was done simultaneously, and DQN was trained prior. However, the decision times were still within margin.

Basso et al. (2022) also proposed a RL method for the Dynamic Stochastic Electric Vehicle Routing Problem for transport and/or mobility providers. The idea is to minimize expected energy consumption in the routing problem, as well as minimizing the risk of battery depletion. Monte Carlo simulation was used to learn about the stochastic energy usage and demand. Based on Q-learning, they propose a solution methodology that has three components: Value Function Approximation for reducing the size of the Q-tables, a safe policy for minimizing energy consumption and failures, and a rollout training strategy that, based on heuristics, explores relevant parts of the state-action space fist as Vehicle Routing problems

(VRP's) are computationally hard. 50 test instances in the city of Luxembourg were generated with both 10 and 20 customers and produced reliable and quality results.

These papers showcase the capabilities of RL methods to solve different types of transportation/operational problems involving one or more uncertain factors. A From Table 3.2, it can be seen that many of the publication use ADP, which is suitable for handling dynamic problems that rely on changing information and uncertain inputs. Q-learning is also used for this in combination with VFA. These examples are discussed to sketch the similarities between these types of frameworks and recovery problems in ADM; They contain a fleet, routings, uncertain factors and dynamically changing environments and can be modelled in IP formulations, as well as MDP formulations. Their effectiveness suggests that similar methods can be of particular interest to ADM applications.

Table 3.5: Overview of publications on Operations Management that use RL

Paper	Theme	Uncertainties						Method			
		Order Characteristics	Delivery Costs	Vehicle Supply	Demand	Travel Times	Energy Consumption	DP	ADP	Q-Learning	DQN
Van Heeswijk et al. (2015)	*Delivery Dispatching*	✓						✓	✓		
Firdausiyah et al. (2019)	*Joint Delivery Systems*		✓						✓		
Beirigo et al. (2022)	*Autonomous Ridesharing*			✓	✓				✓		
Alcaraz et al. (2022)	*Long-distance Routing*					✓			✓		
Basso et al. (2022)	*Vehicle Routing*				✓		✓			✓	

3.2.2. Air Transport Management

Many RL applications in Air Transport Management saw light in recent years as well. Problems that were addressed include fleet planning, maintenance scheduling, crew scheduling, revenue management, Air Traffic Management (ATM) and Disruption Management. One drawback for the use of Machine Learning methods (or Artificial Intelligence (AI) in general) is need for explainability of the models. Strict safety regulations in aviation require transparent and explainable models. This need for explainability is part of the gap between research and practical implementation. Current research is being done on bridging this gap by development of eXplainable Artificial Intelligence (xAI) frameworks in ATM (Hernandez et al., 2021). In a survey on xAI in ATM applications, Degas et al. (2022) found that most papers that use AI in ATM did not focus on explaining their results and that xAI should be used in order to reach end users. Nonetheless, as stated, several problems in Air Transport Management have already been addressed using RL in theoretical frameworks.

Geursen et al. (2023) addressed uncertain factors in fleet planning for airlines by proposing a RL-based multi-stage fleet planning framework incorporating stochastic processes that represent demand and fuel price uncertainty. This RL-based approached would allow managers to determine how the aircraft fleet should evolve based on different scenarios. The state variables at each time step consisted of the number of aircraft in the fleet from a certain type, the demand between all airports in the network, and the fuel price. The decision variables that made up the action space consisted of number of aircraft acquired or disposed as well as decision variables for flight frequencies and number of passengers transported. The problem is solved with ADP, where at each time step, new information on demand and fuel prices become available following a stochastic process. These processes where captured via an mean-reverting Ornstein–Uhlenbeck process. An Advantage Actor-Critic (A2C) algorithm was adopted for training and results show superior performance over deterministic methods. t'Hooft (2024) later focused on addressing reward function design for fleet planning problems, and the use of Graph Neural Networks was explored for this.

Mattila and Virtanen (2011) modelled fighter aircraft maintenance scheduling as a semi-MDP, where both maximum fleet readiness and minimum threshold of fleet readiness were assessed as objectives. Compared to heuristics, the RL method was found to be superior.

Ruan et al. (2021) formulated a network-flow based Integer Linear Program for the Operational Aircraft Maintenance Routing problem (OAMRP). They developed a new RL-based algorithm for solving the problem. In their framework, maximum flying-hour, limit on the number of take-offs between two consecutive maintenance checks and the work-force capacities are incorporated. They modelled the state space as the set of flights in the schedule in chronological order, with information on origin, destination, departure time and arrival time. The action space consisted of the assignment of available aircraft to unassigned flights. The objective was to maximize the through value of the assignments and the reward function was modelled as such. A through value is the desirability of one-stop service between a pair of cities, i.e. the extra revenue that would be gained by the airline from extra passengers who are attracted by this through service. Using a proposed Monte Carlo based algorithm they tested the model on real airline data and found that for large instances, the proposed method performed significantly better than heuristic methods. The authors mention the exclusion of disruptive events in the framework as a major drawback.

Andrade et al. (2021) proposed a Deep Q-Learning method to optimize long term maintenance check scheduling by reducing the total number of checks. The state variables are defined by an extensive set of attributes that contain information on the aircraft with respect to a type of check. The action or decision variable corresponds to selecting an aircraft to have its next check scheduled. The results of the DQN approach are compared to a DP approach from a previous study and historical estimates from an airline. The proposed method produced more efficient A- and C-check schedules compared to the airline estimates, and better C-check schedules compared to a DP approach. In Silva et al. (2023), the same authors extended this work by proposing and adaptive method for long term check scheduling that could reschedule based on new maintenance information like Remaining Useful Life (RUL) predictions of aircraft equipment. Two Deep Q-learning algorithms were proposed for the long term scheduling and the adaptive scheduling respectively.

Tseremoglou and Santos (2024) addressed the disruptive nature of airline operations in a maintenance context by presenting a two-stage dynamic scheduling framework to solve the aircraft fleet maintenance scheduling in a Condition Based Maintenance (CBM) context. CBM considers probabilistic predictions on the health of the equipment and the stochastic arrival of corrective maintenance tasks. The decision making process for regarding health predictions is modelled as a Partially Observable Markov Decision Process (POMDP) and solved using Partially Observable Monte Carlo Planning (POMCP). The defined policy is then integrated with the (unknown beforehand) corrective tasks in a DQN network. The model continuously creates and adjusts the schedule based on new information. Results showed that incorporating the health monitoring predictions led to a 46.2% costs reduction compared to a corrective approach.

Kenworthy et al. (2021) proposed a technique that combined RL and integer programming to achieve robust crew scheduling to reduce disruption impacts. They presented NICE (Neural network IP Coefficient Extraction) to approximately represent complex objectives in an integer programming formulation. With a Monte Carlo approach, weights for the neural network are extracted as probabilities for pilots that captures how likely assigning a pilot to a slot is to maximize reward in a given scheduling episode. The extracted coefficients are used in the objective function while the model is incentivized to pick pilots with higher probabilities, making the solving process much more efficient for the integer program. Compared to a baseline integer programming formulation and a robust integer programming formulation that explicitly tries to minimize the impact of disruptions, NICE performed 33% and 48% better respectively.

Several more applications of Reinforcement Learning mechanisms relating to Air Transport Management have been published: Shihab (2020) and Bondoux et al. (2020) focused on revenue management by using RL-agents to optimize seat inventory control and developing a Revenue Management System (RMS) that does not require demand forecasts, respectively. Balakrishna et al. (2010) used RL to predict taxi-out times, which would allow for more efficient airport operations. Zhao and Liu (2022) used a physics informed DRL model for aircraft conflict resolution.

in Table 3.6, an overview of the publications on Air Transport Management that use RL is provided. Some of which include uncertain factors like demand, fuel price, maintenance tasks and taxi-out times. In terms of solution methods, the algorithmic classes are distinguished by DP, ADP, TD and MC methods - explained earlier in this section. Note that none of the papers consider explainability in their frameworks.

Table 3.6: Overview of publications on Air Transport Management that use RL

Paper	Theme	Uncertainties						Explainability	Method			
		Demand	Fuel Price	Tasks Arrival	RUL Predictions	Taxi-out Times	None		DP	ADP	TD	MC
Geursen et al. (2023)	*Fleet Planning*	✓	✓					x		✓		
t'Hooft (2024)	*Fleet Planning*						✓	x				
Mattila and Virtanen (2011)	*Maintenance*						✓	x			✓	
Ruan et al. (2021)	*Maintenance*						✓	x				✓
Andrade et al. (2021)	*Maintenance*						✓	x	✓		✓	
Silva et al. (2023)	*Maintenance*				✓			x			✓	
Tseremoglou and Santos (2024)	*Maintenance*			✓	✓			x				✓
Balakrishna et al. (2010)	*ATM*					✓		x		✓		
Kenworthy et al. (2021)	*Crew Scheduling*						✓	x				✓

**TD: Temporal Difference, MC: Monte Carlo*

In the next subsection, RL applications in Airline Disruption Management are discussed.

3.2.3. Airline Disruption Management

To the best of the authors' knowledge, Hondet et al. (2018) was the first to utilize RL as solving mechanism for the ARP. They formulated an aircraft recovery problem using a Q-Learning algorithm to train an agent that performs aircraft swaps after a disruptive event created by a simulator. For the states they modelled two different representations, where one considers the flight path and delay of aircraft, while the other does not. The model considers only aircraft swaps as recovery option and was tested on real-world data and compared to an idle scenario. The results were not optimal but the model showed that it takes relevant decisions regarding recovery.

Lee et al. (2020) combined partial and probabilistic predictions on airport delays with and addressed the dynamic and uncertain nature of the problem with a dynamic programming approach. As stated earlier, the model is solved in a rolling horizon approach using a window length of one hour, but cannot be solved directly via backwards induction because the problem size becomes intractable. They used an look-ahead approximation and sample average approximation method to tackle this. Similar to Alcaraz et al. (2022), the cost-to-go function for the look-ahead period at a certain time step is approximated by Monte Carlo sampling. In the rolling horizon, from one period to the next, the state variable is updated based on prior recovery decisions and revealed disruptions at that time. The model aims to minimize the expected recovery costs based on the probabilistic inputs. A comparison is done to a myopic baseline solution for data using 30 scenarios capturing delays at six hubs for a look-ahead period of 4 hours. Results show that using the partial disruption predictions in combination with an approximate algorithm significantly enhances recovery decisions by reducing the total recovery costs while not adding adding risk for the airlines.

In the framework of Lee et al. (2022), the aircraft recovery problem was modelled as a MDP and the environment consisted of aircraft and aircraft routes, and flights. The environment states is a tuple contain the aircraft states and event information. The states of aircraft contains their previous route, future route, and a binary indicator for whether the aircraft is flying or on the ground. The states on event information contain information on on arrivals and departures. The action variable represents an aircraft being swapped with the departing or arriving aircraft. Q-Learning and Double Q-Learning are implemented, the latter to counter an overestimation bias of Q-Learning. In an experimental study, they validated the advantages of Double Q-Learning and identified three features that differentiate from other studies: The RL approach can delay and swap flights without relying on copies of flight arcs, the RL method is able to flexibly adapt to multiple objectives, and the RL methods can handle complex real-world conditions without heavily relying on a mathematical model. The authors mention that it was difficult to reuse the obtained policy for new problem instances because the states and actions were based on routes. Also, unnecessary aircraft swapping was not modelled to have any negative effects, which it can have in real operations.

Ding et al. (2023) proposed a RL-solution method for an integrated model that could be compared with and exact and heuristic method for multiple size instances. They proposed a VNS heuristic, as well as a Deep-RL based guidance framework for the VNS heuristic. In this framework, they modelled the states space as the solution space of the optimization model (i.e. assignments of aircraft and crew to flights). The actions is a selection of flight strings and neighbourhood operators for both aircraft and crew, such that part of the flight strings can be swapped, cut, inserted and deleted. Proximal Policy Optimization (PPO) was used to train the policy network. To make a trade-off between improvement steps on a policy and accidental performance collapse, a PPO-clip variant is used which relies on specialized clipping in the loss function to remove incentives for the new policy to get far from the old policy. After training, the DRL guided VNS scales well and outperformed the exact and heuristic method while solving even large instances of 931 flights and 279 aircraft in 14 seconds.

Although there are a limited number of studies addressing the ARP by means of RL based methods, there is positive sentiment towards the use of these methods in the future. The papers discussed in this subsection, depicted in 3.7, were either able to handle complex assumptions and conditions, solve recovery problems fast and with quality solutions, or do both. Several methods were used, Q-learning, ADP, and DRL. None of the papers addressed explainability in their propositions however.

Table 3.7: Overview of publications on Disruption Management that use RL

Paper	Recovery			Uncertainties		Explainability	Method			
	Aircraft	Crew	Passengers	Airport Delays	None		Q-learning	D-Q-Learning	ADP	DRL
Hondet et al. (2018)	✓				✓	x	✓			
Lee et al. (2020)	✓			✓		x			✓	
Lee et al. (2022)	✓				✓	x	✓	✓		
Ding et al. (2023)	✓	✓	✓		✓	x				✓

**D-Q-Learning: Double Q-Learning*

It is worth mentioning that in the most recent and complete literature studies on DM, the use of RL methods in solving recovery problems was mentioned every time. Hassan et al. (2021) stated that advanced data analytics form a bridge to the use of RL techniques to make optimal decisions while anticipating future consequences. Santana et al. (2023) emphasized the benefit of intelligent data usage as well. Wu et al. (2024) recommended future studies to use such techniques to enable them to adapt to a broad range of scenarios, making them highly applicable.

3.3. Conclusions on Literature Review

In this literature study, two bodies of literature were reviewed and discussed: 1) Publications on ADM in the context of aircraft recovery and 2) Publications on operations management, Air Transport Management, and ADM that use RL as solution method. Regarding the first body of literature, integrated recovery methods have become more popular since advanced computational approaches have paved the way for addressing more complex problems in terms of problem size and scope. Realistic scenarios for integrated recovery can be solved with tailor made solution methods in acceptable computing times. However, not all solution methods are able to generalize well for different problem instances. Some common recovery actions are present in most publications (aircraft swapping, flight delaying, and flight cancellation) and typically one day of operations is used as recovery horizon. Maintenance requirements are considered an important part of the frameworks and maintenance related recovery actions provide some flexibility in case of maintenance constraints. Furthermore, an increased interest in anticipatory/proactive mechanisms is confirmed by the recent exploration of this topic in literature.

Regarding the second body of literature, the use of ML techniques, particularly RL, has been proven to cope well with uncertain and dynamic problems in transportation/operations management. Due to the highly uncertain and dynamic nature of recovery problems, the use of RL techniques is promising for ADM as well. RL methods are also used in other areas of Air Transport Management like fleet planning, maintenance planning, crew scheduling, revenue management and ATM. Some recent publications already explored the use of RL for disruption management by showcasing an excellent ability to improve solution quality and computational efficiency. The potential to capture complex uncertain environments and produce quality results for recovery problems would indicate the usefulness for RL methods in anticipatory disruption management frameworks. Explainability should be considered when proposing RL frameworks for Air Transport Management applications however.

4 Research Proposal

In this chapter, the proposal for the research direction is presented. In Section 4.1, the main research gaps following the literature study in Chapter 3 are stated and discussed. In Section 4.2, the research objectives and questions following the identified research gaps are outlined. Subsequently, the scope and the methodology for the research are discussed in Sections 4.4 and 4.5.

4.1. Research Gaps

4.1.1. Anticipatory Disruption Management

Some studies explored a proactive attitude towards ADM by proposing models for probabilistic predictions of disruptions (Ogunsina et al. (2019); Ogunsina et al. (2021); Ogunsina et al. (2022)). Lee et al. (2020) proposed an anticipatory recovery model based on such predictions. In a similar model, Zang et al. (2024) used turnaround time predictions to allow proactive recovery. However, these models were limited to on type specific type of disruption (taxiing delays and turnaround delays). No publication proposed a general model that could include any disruption probability distribution as input, for multiple disruption sources, in an anticipatory framework.

4.1.2. Learning Based Methods

Although ML and RL have proven its use for solving airline recovery problems in several cases in literature, none of the studies, with the exception of Lee et al. (2020), used a learning based method that considers future uncertain scenarios. The model of Lee et al. (2020), however, is limited to systemic airport delays and does not consider other disruption sources.

4.1.3. Dynamic Disruption Management

A large number of publications aim to tackle airline recovery by proposing novel frameworks and solution methods. However, only few of them consider the inherent dynamics of a disruption scenario in their framework. Vos et al. (2015) addressed this by proposing a model that recovers the schedule dynamically by building forth on previously found solutions as (new) disruptions happen, but no papers yet have used this dynamic solving based on real-time information on future potential disruptions.

4.1.4. Explainability

Additionally, one overlooked aspect of learning based methods for practical applications is explainability. No papers address the limitations that explainability impose on practical use of learning based methods, which is of particular relevance in aviation (Ribeiro et al. (2024); Degas et al. (2022)).

4.1.5. Operational Constraints

Many studies exist that include important operational constraints like maintenance planning or airport curfews in their framework. However, looking at the research gaps identified above, no anticipatory model exists yet that uses considers maintenance constraint while allowing the flexibility of maintenance related recovery decisions.

4.1.6. Summary

To summarize, several gaps have been identified in the literature surrounding Airline Disruption Management. The main gap is the lack of incorporation of anticipatory mechanisms into solution methods for airline recovery problems. Additionally, although a promising field, the number of learning based methods applied to ADM is still fairly limited, as ML is a relatively new field on itself. The use of RL methods for ADM is only addressed in a handful of papers, where no papers actually addressed the aircraft recovery problem taking into account multiple disruption types. Realistic operational constraints such as maintenance requirements of the aircraft fleet and airport curfews are covered in several papers in literature, but not in an anticipatory framework. Furthermore, explainability regarding the use of RL methods is a remarkably overlooked factor in literature. This is of particular importance in a highly regulated industry such as aviation, where new applications are subject to strict certifications. Development of any ADM tool for practical use should always consider certification limitations. These research will be addressed by 1) developing an anticipatory aircraft recovery model using 2) model-based RL while 3) including operational constraints and 4) making efforts to explain the models' actions and assess the robustness of the model.

4.2. Research Objectives

From an industry perspective, the improvement of disruption management tools can be of significant economic and environmental benefit. anticipatory disruption management is potentially a breakthrough approach to the Airline Recovery Problem. When combined with advanced data analytic tools for disruption forecasts, it will allow airlines to drastically reduce disruption impacts in a manner that reactive methods cannot achieve. Furthermore, the belief that RL is a capable method for realizing such anticipatory models motivates us to utilize this in the solution approach. Regarding the use of RL in aviation applications, the use of model-based methods can allow for more explainability than model-free models, since it allow the agent to communicate not only its goals, but also the way it intends to achieve them (Moerland et al., 2018). Given the state-of-the-art, the problem scenario and the identified research gaps the following research objective can be defined:

Research Objective

”Develop an anticipatory aircraft recovery model that proactively and effectively mitigates the effects of potential future disruptions, in a model-based Reinforcement Learning framework”

The research objective can be segmented into the following sub-goals:

1. Develop or adopt and modify a baseline ARP MIP model suitable for the scope of the research;
2. Define disruption sources, characteristics and corresponding probability distributions;
3. Model the problem as a Markov Decision Process and train the model to make relevant recovery decisions when subject to potential future disruptions;
4. Validate the model and the results;
5. Analyse the models’ robustness and explain its actions.

4.3. Research Questions

With the research objective clear, the following research question with corresponding sub-questions are defined.

Research Question

”How can model-based Reinforcement Learning effectively be applied to an proactive Aircraft Recovery Problem that anticipates future disruptions by taking into account short term disruption uncertainty?”

Sub-questions

1. How can anticipated future disruptions be incorporated in an ARP MIP model to effectively mitigate disruption effects before they occur?
2. How can the model be trained to perform optimal recovery actions in the face of potential future disruptions?
3. How can the arrival of new disruption information during the recovery process be incorporated in the decision making process?
4. How can the ARP be modified to include operational constraints such as maintenance and curfews?
5. How can probabilistic propagation effects of disruptions be modelled and mitigated?
6. How can the scalability and long term performance of the proactive recovery model be validated?
7. How can robustness of the model be assessed?
8. How can the models’ actions be explained?

In the next section, the scope of the research will be defined.

4.4. Scope

Having defined the research objective and research questions, the scope of the model can be determined. As already mentioned, this research will focus solely on the recovery of aircraft. The goal is to explore an anticipatory mechanism that improves the solution quality when anticipating future scenarios compared to myopic scenario’s, not to build the most comprehensive model that captures all resources, constraints and recovery actions surrounding airline recovery. Therefore, the scope and size of the problem will be limited. The focus will lie on the definition of the algorithmic design and performance, especially in the way it handles probabilities as input for potential future disruptions. For this purpose, *only aircraft recovery is considered*. In this section, the types of disruptions that will be considered are discussed first, after which the objectives and possible recovery actions are determined.

4.4.1. Disruptions

In the reviewed literature, the most common disruption types are *Airport Closures* (AC) and *Aircraft Unavailability* (AU), as many disruptive events translate into one of these scenarios. For this reason, these two disruption types will be covered in the framework. However, both should be formally defined to avoid ambiguity in their meaning. Regarding *AC*, this will be defined as an airport being restricted of any arrivals or departures in a given time span, defined as the disruption duration. *AU* is defined by an aircraft being unexpectedly grounded during a certain time span, defined as the disruption duration. In both cases, the aircraft and airport become fully operational again after the disruption duration has passed. Optionally, *Reduced Airport Capacity* (RAC) can be considered into the framework as well, which would translate to flight delays as a disruption type.

Most works that tackled an ARP solved the problem on data from one day of operations. Using one day of operations allows for easy implementation of constraints regarding curfews and scheduled overnight airports for individual aircraft to allow for schedule continuity. Naturally, it is also more desirable for airlines to recovery their schedules earlier (the same day) rather than later (next day or later). For this reason, recovery will take place during one day of operations in this framework as well. The model will perform actions at discrete time steps during the recovery period. in this period, the environment is subject to potential disruption realizations based on some assumed probabilities. These probabilities become known to the model at a certain time. At this time, the recovery period begins as the model proactively will try to make optimal recovery decisions when subject to these potential future disruptions. However, no disruptions have actually occurred yet. Theoretically, the model should handle any given input disruption probability distribution. However, Hassan (2019) presented disruption statistics from a case study that could be used to resemble real life disruption probabilities. Vink et al. (2020) also derived disruptions probability distributions for several disruption types based on historical data. Note that except for a scenario with reduced airport capacity, delays are not considered as a disruption type in this framework. They do play a role, however, as they result from the aforementioned disruptions by means of downstream propagation.

4.4.2. Recovery Actions and Objectives

In terms of recovery actions, *aircraft swaps* (SW), *flight delays* (DL), and *flight cancellations* (CL) are most commonly used in literature. Given the fact that maintenance requirements should be part of the framework, *maintenance swaps* (MSW) could be incorporated as a third recovery action. The definitions of the actions are as follows: SW is defined as the swapping of the assigned flights of two aircraft, given that it would still produce a feasible schedule. DL is defined as deliberately delaying a flights' departure time. MSW is defined as swapping the assignment of two flexible (postponable) maintenance tasks to maintenance slots for two aircraft, given that it will still produce a feasible schedule. Given the limited scope for the model, the objective is not based on any financial or economic loss, as this would require detailed background information for the quantification of how schedule disruptions are translated into economic losses. Therefore, the main objective is defined as *the minimization of total flight delay*, accompanied by mitigation of the number of flight cancellations. Additionally, penalties could be given for curfew violations and maintenance tasks violations. Some of these violations are practically possible but highly unfavorable for airlines and should only be relied on in extreme cases as a last resort.

In case of an infeasible schedule (e.g. when a flight has to be cancelled because it cannot be performed), the model should allow this flight to be cancelled while imposing a significant penalty. This ensures the model will still find solutions in heavily disrupted scenarios. Flight cancellations often serve as a last resort to ensure feasibility (and thereby automatically minimizing disruption impacts), or to prevent penalties for maintenance and curfew violations being imposed. They can also, however, be used in case the model thinks that cancelling a flight is better for the long term then allowing the flight to continue, which can potentially result in severe delay propagation through the schedule.

4.5. Methodology

In order to answer the research questions, several steps have to be undertaken. First, sample data must be gathered and the optimization model, MDP formulation, and algorithmic design should be determined. Subsequently, the environment and agent are implemented, after which the agent can be trained. In this phase, the model is very limited in terms of scope and capabilities: Only UA with deterministic duration is considered as a disruption source. Regarding recovery actions, aircraft swapping, delaying flights and cancelling flights will be included. After training, the model will be tested on small problem instances to validate its capabilities. When this is done, the model can be expanded to incorporate more disruptions, constraints and recovery decisions. Specifically, maintenance-related constraints and recovery decisions can be of interests as additional feature. Also, more complex disruption scenarios are to be included in the framework such as different disruption sources with uncertain disruption durations. Incorporating these additional features will be done iteratively by re-training the model on small instance every time a new feature is included. Once the model is able to capture the desired scope, the model will be tested on both small and large problem instances and validation is done. The performance of the model can then be compared to a baseline ARP MIP optimization model that follows a reactive only approach. In parallel, the robustness and explainability of the model will be analyzed.

Below, the methodological steps are listed and in Figure 4.1, the full framework of this MSc Thesis is outlined.

1. Gather sample flight data;
2. Define CO model, MDP, Disruption Probabilities;
3. Define algorithmic design;
4. Implement agent and environment;
5. Train model;
6. Verify model;
7. Expand model;
8. Validate and compare model;
9. Analyze explainability and robustness;
10. Conclusions and recommendations.

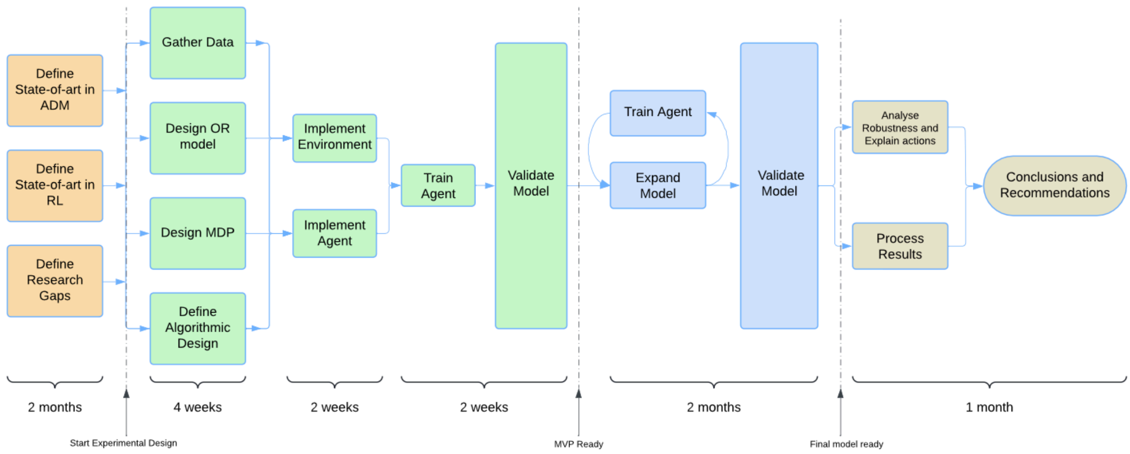


Figure 4.1: MSc Thesis Research Framework

4.6. Planning and Resources

In order to model the problem and answer the research questions, several resources are needed. Mainly Python will be used for implementation of the model. This will be done by using Gurobi, several RL packages, and robustness and explainability packages. A Baseline optimization model is derived from existing models from literature or open source code.

4.6.1. Data and Related Risks

Regarding the required data for the project. Datasets will be either provided by Boeing, taken from publicly available sources (literature, online sources), or by synthesis in case insufficient real-life sample data is available for training. Two main data types are required: Operational data from airlines, and disruption data. In case no conclusions on historical data for disruption probability distributions can be drawn, these will have to be assumed. This might result in the model training for disruption scenarios that would not occur in real-life, thus making it unsuitable for practical application. However, once the model is trained and able to effectively anticipate these potential future disruptions based on assumed distributions, it can easily be re-trained for realistic disruption scenarios when real-world reflecting disruption data becomes available.

4.6.2. Planning

Furthermore, the model can be ran on a server suitable for large optimization problems. The research is split into four phases: Literature Review and Research Definition, Experimental Design, Experimental Campaign, and Research Dissemination. In Figure 4.2, a Gantt chart is shown containing the planning for the entire research divided into the four phases, with milestone moments indicated and vacations included for the author as well as the supervisors.

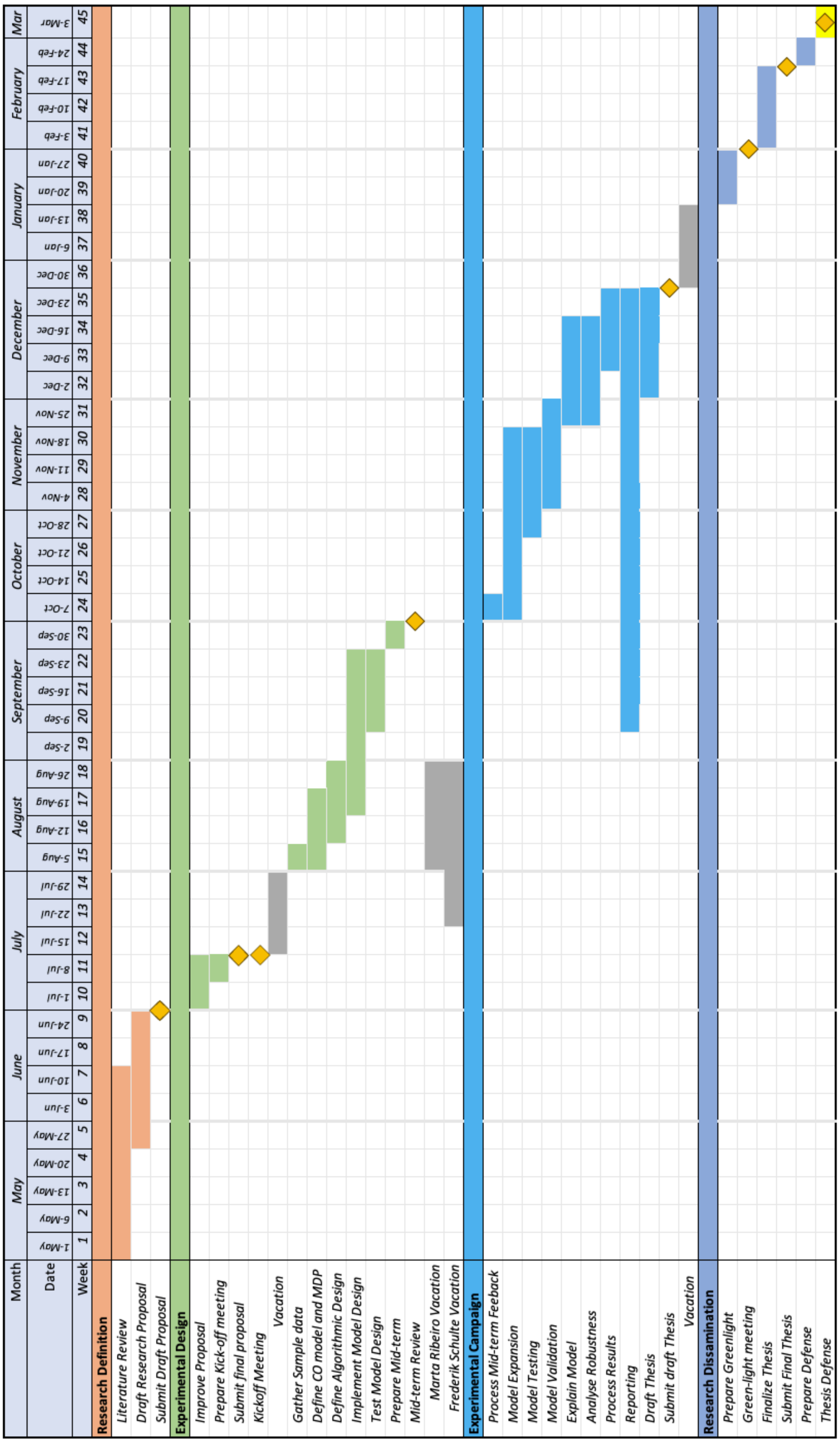


Figure 4.2: MSc Thesis Planning, depicted in a Gantt chart

5 Conclusion

Disruptions pose a significant operational challenge for airlines as their operations run on highly optimized schedules with little room for error. With 8% of worldwide airline revenue being lost to disruption-incurred costs, managing and mitigating disruptions is crucial for airline performance. The competitive nature of commercial air transport amplifies this as airlines risk to lose customers during disruptions as a result of customer dissatisfaction. As of today, the airline recovery problem is solved in a sequential process. Operators act after disruptions happen, focusing first on aircraft recovery, then crew, then passengers. This yields suboptimal solutions because airline resources operate on highly interconnected schedules. In the literature surrounding ADM, many efforts have been made to improve existing models in order to solve more complex, larger problem instances in a matter of minutes, as this is required for airline operations controllers. Only few studies consider looking at potential future disruptions.

This research explores a novel solution approach that integrates anticipated potential future disruptions into the aircraft recovery problem by using Reinforcement Learning as a way to efficiently capture problem dynamics and uncertainties while being effective in terms of solution quality. Thereby showcasing the potential benefits of maintaining a proactive attitude towards disruptions as opposed to a reactive one, as well as the benefits of using Reinforcement Learning as a tool for solving these types of problems fast and effectively.

To achieve this, the following research question is defined:

"How can model-based Reinforcement Learning effectively be applied to an proactive Aircraft Recovery Problem that anticipates future disruptions by taking into account short term disruption uncertainty?"

The focus of the research will lie on the anticipatory mechanisms and the ability to outperform reactive models when subject to potential future disruption scenarios. Therefore, the scope is limited to aircraft recovery. After defining the optimization model, MDP, and algorithmic design, a simple proof of concept model will be developed and trained, with a limited initial scope. After verification of this model, the scope can be expanded to incorporate more operational constraints and potential recovery actions. After retraining and verifying the model, validation is done and a baseline reactive optimization model is used as benchmark comparison. Also, the robustness and explainability of the model will be analyzed.

Bibliography

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

Alcaraz, J., Losilla, F., and Caballero-Arnaldos, L. (2022). Online model-based reinforcement learning for decision-making in long distance routes. *Transportation Research Part E: Logistics and Transportation Review*, 164:102790. https://doi.org/10.1016/j.tre.2022.102790.

Andrade, P., Silva, C., Ribeiro, B., and Santos, B. F. (2021). Aircraft maintenance check scheduling using reinforcement learning. *Aerospace*, 8(4). https://doi.org/10.3390/aerospace8040113.

Arguello, M. (1998). Framework for exact solutions and heuristics for approximate solutions to airlines irregular operations control aircraft routing problem.

Arikan, U., Gürel, S., and Akturk, S. (2017). Flight network-based approach for integrated airline recovery with cruise speed control. *Transportation Science*, 51. https://doi.org/10.1287/trsc.2016.0716.

Balakrishna, P., Ganesan, R., and Sherry, L. (2010). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures. *Transportation Research Part C: Emerging Technologies*, 18(6):950–962. https://doi.org/10.1016/j.trc.20https://doi.org/10.03.003.

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

Basso, R., Kulcsár, B., Sanchez-Diaz, I., and Qu, X. (2022). Dynamic stochastic electric vehicle routing with safe reinforcement learning. *Transportation Research Part E: Logistics and Transportation Review*, 157:102496. https://doi.org/10.1016/j.tre.2021.102496.

Beirigo, B., Schulte, F., and Negenborn, R. (2022). A learning-based optimization approach for autonomous ridesharing platforms with service level contracts and on-demand hiring of idle vehicles. *Transportation Science*. https://doi.org/10.1287/trsc.2021.1069.

Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421. https://doi.org/10.1016/j.ejor.2020.07.063.

Bisaillon, S., Cordeau, J., Laporte, G., and Pasin, F. (2010). A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR*, 9:139–157. https://doi.org/10.1007/s10288-010-0145-5.

Bondoux, N., Nguyen, A., Fiig, T., and Acuna-Agost, R. (2020). Reinforcement learning applied to airline revenue management. *Journal of Revenue and Pricing Management*, 19. https://doi.org/10.1057/s41272-020-00228-4.

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

Cadarso, L. and Vaze, V. (2022). Passenger-centric integrated airline schedule and aircraft recovery. *Transportation Science*, 57. https://doi.org/10.1287/trsc.2022.1174.

Castro, A. J. M., Rocha, A., and Oliveira, E. (2014). *A New Approach for Disruption Management in Airline Operations Control*, volume 562. https://doi.org/10.1007/978-3-662-43373-7.

Chen, K., Wu, J., and Yang, L. (2023). Reassigning departure slots with preferences of the airline and passengers. *Transportation Research Record: Journal of the Transportation Research Board*, 2678. https://doi.org/10.1177/03611981231182714.

Clarke, M. D. D. (1998). Irregular airline operations: a review of the state-of-the-practice in airline operations control centers. *Journal of Air Transport Management*, 4(2):67–76. https://doi.org/10.1016/S0969-6997(98)00012-X.

Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry—concepts, models and methods. *Computers Operations Research*, 37(5):809–821. Disruption Management.

Dang, C. and Ye, Y. (2015). A fixed point iterative approach to integer programming and its distributed computation. *Fixed Point Theory and Applications*, 2015. https://doi.org/10.1186/s13663-015-0429-8.

Degas, A., Islam, M., Hurter, C., Barua, S., Rahman, H., Poudel, M., Ruscio, D., Ahmed, M., Begum, S., Rahman, M., Bonelli, S., Cartocci, G., Di Flumeri, G., Borghini, G., Babiloni, G., and Aricó, P. (2022). A survey on artificial intelligence (ai) and explainable ai in air traffic management: Current trends and development with future research trajectory. *Applied Sciences*, 12(3). https://doi.org/10.3390/app12031295.

Ding, Y., Wandelt, S., Wu, G., Xu, Y., and Sun, X. (2023). Towards efficient airline disruption recovery with reinforcement learning. *Transportation Research Part E: Logistics and Transportation Review*, 179:103295. https://doi.org/10.1016/j.tre.2023.103295.

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

Eikelenboom, B. (2022). A real-time decision-support tool for the integrated airline recovery using a machine learning approach. *MSc. Thesis, Faculty of Aerospace Engineering, Delft University of Technology*. http://resolver.tudelft.nl/uuid:fbe1a4af-54f6-44fc-93ed-6e1069b4ed2a.

Eshkevari, M., Komijan, A., and Baradaran, V. (2023). A bi-objective model for aircraft and crew recovery considering individual flight legs and maintenance requirements. *Engineering Optimization*, 0(0):1–27. https://doi.org/10.1080/0305215X.2023.2284836.

Eurocontrol (2024). All-causes delay to air transport in europe. Accessed on 04.06.2024.

Evler, J., Lindner, M., Fricke, H., and Schultz, M. (2022). Integration of turnaround and aircraft recovery to mitigate delay propagation in airline networks. *Computers Operations Research*, 138:105602. https://doi.org/10.1016/j.cor.2021.105602.

Filar, J., Manyem, P., and White, K. (2001). How airlines and airports recover from schedule perturbations: A survey. *Annals of Operations Research*, 108:315–. https://doi.org/10.1023/A:1016079600083.

Firdausiyah, N., Taniguchi, E., and Qureshi, A. (2019). Modeling city logistics using adaptive dynamic programming based multi-agent simulation. *Transportation Research Part E: Logistics and Transportation Review*, 125:74–96. https://doi.org/10.1016/j.tre.2019.02.011.

Fogaça, L., Henriqson, E., Junior, G., and Lando, F. (2022). Airline disruption management: A naturalistic decision-making perspective in an operational control centre. 16. https://doi.org/10.1177/1555343421106102.

Gershkoff, I. (2016). "airline disruption management". accessed on 20.05.2024. https://amadeus.com/documents/en/blog/pdf/2016/09/airline-disruption-management-whitepaper-2016.pdf.

Geursen, I., Santos, B., and Yorke-Smith, N. (2023). Fleet planning under demand and fuel price uncertainty using actor–critic reinforcement learning. *Journal of Air Transport Management*, 109:102397. https://doi.org/10.1016/j.jairtraman.2023.102397.

Hassan, L. (2019). Aircraft disruption management increasing performance with machine learning predictions. *MSc. Thesis, Faculty of Aerospace Engineering, Delft University of Technology*. http://resolver.tudelft.nl/uuid:07efd193-1d6a-43ad-9461-ccc885dfcb9a.

Hassan, L., Santos, B., and Vink, J. (2021). Airline disruption management: A literature review and practical challenges. *Computers Operations Research*, 127:105137. https://doi.org/10.1016/j.cor.2020.105137.

Heinold, A. (2024). A tutorial on value function approximation for stochastic and dynamic transportation. *4OR*, 22(1):145–173. https://doi.org/10.1007/s10288-023-00539-.

Hernandez, C. S., Ayo, S., and Panagiotakopoulos, D. (2021). An explainable artificial intelligence (xai) framework for improving trust in automated atm tools. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pages 1–10. https://doi.org/10.1109/DASC52595.2021.9594341.

Hondet, G., Delgado, L., and Gurtner, G. (2018). Airline disruption management with aircraft swapping and reinforcement learning. *HAL Open Science*. https://enac.hal.science/hal-01944612.

Hu, Y., Liao, H., Zhang, S., and Song, Y. (2017). Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft. *Expert Systems with Applications*, 83:283 – 299. https://doi.org/10.1016/j.eswa.2017.04.031.

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

Hu, Y., Xu, B., Bard, J., Chi, H., and Gao, M. (2015). Optimization of multi-fleet aircraft routing considering passenger transiting under airline disruption. *Computers and Industrial Engineering*, 80:132 – 144. https://doi.org/10.1016/j.cie.2014.11.026.

Hu, Y., Zhang, P., Fan, B., Zhang, S., and Song, J. (2021). Integrated recovery of aircraft and passengers with passengers' willingness under various itinerary disruption situations. *Computers Industrial Engineering*, 161:107664. https://doi.org/10.1016/j.cie.2021.107664.

Huang, Z., Luo, X., Jin, X., and Karichery, S. (2022). An iterative cost-driven copy generation approach for aircraft recovery problem. *European Journal of Operational Research*, 301(1):334–348. https://doi.org/10.1016/j.ejor.2021.https://doi.org/10.055.

Jozefowiez, N., Mancel, C., and Mora-Camino, F. (2013). A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions. *Journal of the Operational Research Society*, 64(3):384–395. https://doi.org/10.1057/jors.2012.20.

Kenworthy, L., Nayak, S., Chin, C., and Balakrishnan, H. (2021). NICE: robust scheduling through reinforcement learning-guided integer programming. *CoRR*, abs/2109.12171.

Khaled, O., Minoux, M., Mousseau, V., Michel, S., and Ceugniet, X. (2018). A multi-criteria repair/recovery framework for the tail assignment problem in airlines. *Journal of Air Transport Management*, 68:137–151. https://doi.org/10.1016/j.jairtraman.2017.https://doi.org/10.002.

Khiabani, A., Komijan, A., Ghezavati, V., and Mohammadi Bidhandi, H. (2022). A mathematical model for integrated aircraft and crew recovery after a disruption: a benders' decomposition approach. *Journal of Modelling in Management*, 18. https://doi.org/10.1108/JM2-02-2022-0046.

Kohl, N., Larsen, A., Larsen, J., Ross, A., and Tiourine, S. (2007). Airline disruption management—perspectives, experiences and outlook. *Journal of Air Transport Management*, 13:149–162. https://doi.org/10.1016/j.jairtraman.2007.01.001.

Le, M. and Wu, C. (2013). Solving airlines disruption by considering aircraft and crew recovery simultaneously. *J. Shanghai Jiaotong Univ. (Sci.)*, 18:243–252. https://doi.org/10.1007/s12204-013-1389-y.

Lee, J., Lee, K., and Moon, I. (2022). A reinforcement learning approach for multi-fleet aircraft recovery under airline disruption. *Applied Soft Computing*, 129:109556. https://doi.org/10.1016/j.asoc.2022.109556.

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

Lettovsky, L., E, J., and G., N. (1997). Airline crew recovery. *Transportation Science*.

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

Liu, K., Jin, J., and Xiao, F. (2023). Optimizing aircrew recovery considering long connections: A column generation based approach. *Computers Industrial Engineering*, 185:109652. https://doi.org/10.1016/j.cie.2023.109652.

Maher, S. (2015a). A novel passenger recovery approach for the integrated airline recovery problem. *Computers Operations Research*, 57:123–137. https://doi.org/10.1016/j.cor.2014.11.005.

Maher, S. (2015b). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50:150310112838004. https://doi.org/10.1287/trsc.2014.0552.

Marks, J. (2014). "updating airline cancellation costs and customer disruption". accessed on 10.06.2024. https://airinsight.com/wp-content/uploads/2014/10/Updating-airline-cancellation-costs-and-customer-disruption.pdf.

Marla, L., Vaaben, B., and Barnhart, C. (2017). Integrated disruption management and flight planning to trade off delays and fuel burn. *Tranportation Science*, 51. https://doi.org/10.1287/trsc.2015.0609.

Mattila, V. and Virtanen, K. (2011). Scheduling fighter aircraft maintenance with reinforcement learning. *Proceedings - Winter Simulation Conference*, pages 2535–2546. https://doi.org/10.1109/WSC.2011.6147962.

Mazyavkina, N., Sviridov, S., Ivanov, S., and Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers Operations Research*, 134:105400. https://doi.org/10.1016/j.cor.2021.105400.

Moerland, T., Broekens, J., Plaat, A., and Jonker, C. (2018). Model-based reinforcement learning: A survey. https://arxiv.org/abs/2006.16712.

Ogunsina, K., Bilionis, I., and DeLaurentis, D. (2021). Exploratory data analysis for airline disruption management. *Machine Learning with Applications*, 6:100102. https://doi.org/10.1016/j.mlwa.2021.100102.

Ogunsina, K., Papamichalis, M., Bilionis, I., and DeLaurentis, D. (2019). Hidden markov models for pattern learning and recognition in a data-driven model for airline disruption management. https://doi.org/10.2514/6.2019-3508.

Ogunsina, K., Papamichalis, M., and DeLaurentis, D. (2022). Relational dynamic bayesian network modeling for uncertainty quantification and propagation in airline disruption management. *Engineering Applications of Artificial Intelligence*, 112:104846. https://doi.org/10.1016/j.engappai.2022.104846.

Onggo, B., Worthington, D., and Rhodes-Leader, L. (2018). Airline disruption recovery using symbiotic simulation and multi-fidelity modelling. https://doi.org/10.1109/WSC.2017.8248218.

Petersen, J., Sölveling, G., Clarke, J.-P., Johnson, E., and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transportation Science*, 46:482–500. https://doi.org/10.2307/23362876.

Rhodes-Leader, L., Nelson, B., Onggo, S., and Worthington, D. (2022). A multi-fidelity modelling approach for airline disruption management using simulation. *Journal of the Operational Research Society*, 73(10):2228–2241. https://doi.org/10.1080/01605682.2021.1971574.

Ribeiro, M., Tseremoglou, I., and Santos, B. (2024). Certification of reinforcement learning applications for air transport operations based on criticality and autonomy. https://doi.org/10.2514/6.2024-1463.

Rosenberger, J., Johnson, E., and Nemhauser, G. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, 37:408–421. https://doi.org/10.1287/trsc.37.4.408.23271.

Ruan, J., Wang, Z., Chan, F. T., Patnaik, S., and Tiwari, M. (2021). A reinforcement learning-based algorithm for the aircraft maintenance routing problem. *Expert Systems with Applications*, 169:114399. https://doi.org/10.1016/j.eswa.2020.114399.

Santana, M., De La Vega, J., Morabito, R., and Pureza, V. (2023). The aircraft recovery problem: A systematic literature review. *EURO Journal on Transportation and Logistics*, 12:100117. https://doi.org/10.1016/j.ejtl.2023.100117.

Santos, B. F., Wormer, M. M., Achola, T. A., and Curran, R. (2017). Airline delay management problem with airport capacity constraints and priority decisions. *Journal of Air Transport Management*, 63:34–44. https://doi.org/10.1016/j.jairtraman.2017.05.003.

Serrano, J. and Kazda, A. (2017). Airline disruption management: yesterday, today and tomorrow. *Transportation Research Procedia*, 28:3–10. https://doi.org/10.1016/j.trpro.2017.12.162.

Shihab, S. A. M. (2020). *DeepARM: An Airline Revenue Management System for Dynamic Pricing and Seat Inventory Control using Deep Reinforcement Learning*. PhD thesis. https://doi.org/10.13140/RG.2.2.11082.59848.

Silva, C., Andrade, P., Ribeiro, B., and Santos, B. (2023). Adaptive reinforcement learning for task scheduling in aircraft maintenance. *Scientific Reports*, 13. https://doi.org/10.1038/s41598-023-41169-3.

Sinclair, K., Cordeau, J., and Laporte, G. (2014). Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233(1):234–245. https://doi.org/10.1016/j.ejor.2013.08.034.

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

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

Sun, F., Liu, H., and Zhang, Y. (2022). Integrated aircraft and passenger recovery with enhancements in modeling, solution algorithm, and intermodalism. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9046–9061. https://doi.org/10.1109/TITS.2021.3090329.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Teodorovic, D. and Guberinic, S. (1984). Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15(2):178–182.

Teodorovic, D. and Stojkovic, G. (1990). Model for operation delay airline scheduling. *Transportation Planning and Technology*, 14(4):273–285.

Thengvall, B., Bard, J., and Yu, G. (2000). Balancing user preferences for aircraft schedule recovery during irregular operations. *IIE Transactions (Institute of Industrial Engineers)*, 32(3):181 – 193. https://doi.org/10.1080/07408170008963891.

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

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

t'Hooft, C. (2024). Advancing reinforcement learning fleet planning through robust reward design and graph neural networks. *MSc Thesis, Delft University of Technology*.

Tseremoglou, I. and Santos, B. F. (2024). Condition-Based Maintenance scheduling of an aircraft fleet under partial observability: A Deep Reinforcement Learning approach. *Reliability Engineering and System Safety*, 241(C). https://doi.org/10.1016/j.ress.2023.10958.

Van Heeswijk, W., Mes, M., and Schutten, M. (2015). An approximate dynamic programming approach to urban freight distribution with batch arrivals. pages 61–75. https://doi.org/10.1007/978-3-319-24264-4_5.

Vink, J., Santos, B., Verhagen, W., Medeiros, I., and Filho, R. (2020). Dynamic aircraft recovery problem - an operational decision support framework. *Computers Operations Research*, 117:104892. https://doi.org/10.1016/j.cor.2020.104892.

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

Walker, C. (2017). Coda digest q3. Accessed on 17.05.2024.

Wandelt, S., Xu, Y., and Sun, X. (2023). Measuring node importance in air transportation systems: On the quality of complex network estimations. *Reliability Engineering System Safety*, 240:109596. https://doi.org/10.1016/j.ress.2023.109596.

Wang, D., Wu, Y., Hu, J.-Q., Liu, M., Yu, P., Zhang, C., and Wu, Y. (2019). Flight schedule recovery: A simulation-based approach. *Asia-Pacific Journal of Operational Research*, 36(06):1940010. https://doi.org/10.1142/S0217595919400104.

Wen, X., Sun, X., Sun, Y., and Yue, X. (2021). Airline crew scheduling: Models and algorithms. *Transportation Research Part E: Logistics and Transportation Review*, 149:102304. https://doi.org/10.1016/j.tre.2021.102304.

Wu, S., Liu, E., Cao, R., and Bai, Q. (2024). Airline recovery problem under disruptions: A review. https://doi.org/10.48550/arXiv.2401.04866.

Wu, Z., Gao, Q., Li, B., Dang, C., and Hu, F. (2017c). A rapid solving method to large airline disruption problems caused by airports closure. *IEEE Access*, 5:26545 – 26555. https://doi.org/10.1109/ACCESS.2017.2773534.

Wu, Z., Li, B., and Dang, C. (2017b). Solving multiple fleet airline disruption problems using a distributed-computation approach to integer programming. *IEEE Access*, 5:19116–19131. https://doi.org/10.1109/ACCESS.2017.2747155.

Wu, Z., Li, B., Dang, C., Hu, F., Zhu, Q., and Fu, B. (2017a). Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming. *Neurocomputing*, 269:232 – 255. https://doi.org/10.1016/j.neucom.2017.02.091.

Xu, Z., Guo, Y., and Joseph, H. (2021). Accurate remaining useful life prediction with uncertainty quantification: a deep learning and nonstationary gaussian process approach. *CoRR*, abs/2109.12111. https://arxiv.org/abs/2109.12111.

Yetimoğlu, Y. and Aktürk, S. (2021). Aircraft and passenger recovery during an aircraft's unexpected unavailability. *Journal of Air Transport Management*, 91:101991. https://doi.org/10.1016/j.jairtraman.2020.101991.

Zang, H., Zhu, J., Zhu, Q., and Gao, Q. (2024). A proactive aircraft recovery approach based on airport spatiotemporal network supply and demand coordination. *Computers Operations Research*, 165:106599. https://doi.org/10.1016/j.cor.2024.106599.

Zhang, D., Henry Lau, H., and Yu, C. (2015). A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. *Computers Industrial Engineering*, 87:436–453. https://doi.org/10.1016/j.cie.2015.05.033.

Zhang, D., Yu, C., Desai, J., and Henry Lau, H. (2016). A math-heuristic algorithm for the integrated air service recovery. *Transportation Research Part B: Methodological*, 84:211–236. https://doi.org/10.1016/j.trb.2015.11.016.

Zhao, A., Bard, J. F., and Bickel, J. E. (2023). A two-stage approach to aircraft recovery under uncertainty. *Journal of Air Transport Management*, 111:102421. https://doi.org/10.1016/j.jairtraman.2023.102421.

Zhao, P. and Liu, Y. (2022). Physics informed deep reinforcement learning for aircraft conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8288–8301. https://doi.org/10.1109/TITS.2021.3077572.

Zhong, H., Lian, Z., Zhou, T., and Niu, B. (2024). A time-varying competitive swarm optimizer for integrated flight recovery with multi-objective and priority considerations. *Computers Industrial Engineering*, 190:110019. https://doi.org/10.1016/j.cie.2024.110019.

A
Appendix

Abbreviations

Abbreviation	Definition
AC	*Airport Closure*
ADM	*Airline Disruption Management*
ADP	*Approximate Dynamic Programming*
AI	*Artificial Intelligence*
AOCC	*Airline Operations Control Centre*
A2C	*Advantage Actor-Critic*
ARP	*Airline Recovery Problem*
ATM	*Air Traffic Management*
AU	*Aircraft Unavailability*
CBM	*Condition Based Maintenance*
CG	*Column Generation*
C&RG	*Column- and Row-Generation*
CO	*Combinatorial Optimization*
DRL	*Deep Reinforcement Learning*
DM	*Disruption Management*
DP	*Dynamic Programming*
DQN	*Deep Q-Networks*
DSS	*Disruption Set Solver*
DL	*Flight Delaying*
GDP	*Ground Delay Program*
GRASP	*Greedy Randomized Adaptive Search Procedure*
HMM	*Hidden Markov Model*
IROPS	*Irregular Operations*
LP	*Linear Program*
LNS	*Large Neighbourhood Search*
MAS-ADP	*Multi-Agent Simulation-Adaptive Dynamic Programming*
MC	*Monte Carlo*
MCTS	*Monte Carlo Tree Search*
MDP	*Markov Decision Process*
MIP	*Mixed Integer Program*
MILP	*Mixed Integer Linear Program*
ML	*Machine Learning*
MPC	*Model Predictive Control*
MSW	*Maintenance Swapping*
NICE	*Neural network IP Coefficient Extraction*
OR	*Operations Research*
OAMRP	*Operational Aircraft Maintenance Routing problem*
PHM	*Prognostics and Health Monitoring*
POMDP	*Partially Observable Markov Decision Process*
POMCP	*Partially Observable Monte Carlo Planning*
PPO	*Proximal Policy Optimization*
RAC	*Reduced Airport Capacity*
RMS	*Revenue Management System*
RL	*Reinforcement Learning*
RUL	*Remaining Useful Life*
SW	*Aircraft Swapping*
TD	*Temporal Difference*
UTFM	*Uncertainty Transfer Function Model*
UA	*Aircraft unavailability*
VFA	*Value Function Approximation*
VNS	*Variable Neighbourhood Search*
VRP	*Vehicle Routing Problem*
xAI	*eXplainable Artificial Intelligence*

List of Figures

List of Tables