A real-time decision-support tool for the integrated airline recovery using a machine learning approach

Air Transport Operations MSc thesis

Berend Eikelenboom

TU Delft

A real-time decision-support tool for the integrated airline recovery using a machine learning approach

Air Transport Operations MSc thesis

by

Berend Eikelenboom

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 05/08/2022.

Student number: 4538870
Project duration:
10/2021 – 08/2022

Thesis committee:	Dr. B.F. Santos	TU Delft, Supervisor
	Dr. O.A. Sharpanskykh	TU Delft, Chair
	Dr. G. la Rocca	TU Delft, Examiner

TU Delft

Acknowledgements

By submitting this thesis, I will finish my two-year master program Air Transport Operations at TU Delft. One year ago, I was approaching the end of my internship. It was approximately at that time that I started studying potential subjects for my Master's thesis. Rather quickly, I became interested in conducting research at a university abroad. However, upon sharing my ideas with Bruno Santos, I learned that this would not be a possibility. At first, I was a little disappointed, but the disappointment quickly faded when Bruno proposed me an alternative thesis subject: A challenging exercise on airline disruption management, including the use of operations research and machine learning. This immediately sparked my interest, as I greatly enjoyed Bruno's courses on operations research throughout my master, and had always wanted to learn more about the combination of operations research and machine learning. At that time, my knowledge of machine learning was very limited, however the increasing popularity of machine learning made me take a leap of faith and take on this opportunity. And this decision has turned out to be a great one: I've very much enjoyed the past 10 months conducting this research project. Looking back, I'm very glad that Bruno has convinced me to stay in Delft!

In addition to proposing this subject to me, I'd like to thank Bruno for all the appreciation shown for my work, but also the critical feedback on parts to be improved. Your feedback has greatly helped me improve my final work. Furthermore, your experience, knowledge and interest in the airline industry and in the field of operations research have kept me enthusiastic on the topic throughout the entire project. Finally, I'd like to thank you for the good talks and guidance in the past year.

This research is built upon a series of projects about airline disruption management conducted at the Air Transport Operations department. In the past years, Wieger, Jeroen, Lotfy and Andrej have worked very hard on finding solutions for the airline recovery problem, which have set the basis for this research. I would like to thank Lotfy and Andrej for the many meetings, discussing possible directions for my thesis, answering all my questions and still being sincerely interested in the project. The provided information was very useful last year and helped me to successfully complete the work.

Lastly, I would like to thank Lishuai Li, who was involved during the machine learning part of my thesis. Since my knowledge of machine learning was very limited at the start of the project, the meetings with you about the fundamentals of machine learning and about improving the algorithms used in the research were very helpful. I'm thankful for your willingness to help me improve my thesis.

Berend Eikelenboom
Delft, July 2022

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Aircraft
ADP	Average Delay per Delayed Passenger
AOCC	Airport Operations Control Center
AS	Aircraft Swaps
ASV	Aircraft Sink Violations
CDH	Crews Deadheading
CF	Cancelled Flights
CFT	Crew Flight Time Violations
CP	Passengers on Cancelled Flights
CPM	Connecting Passenger Matrix
CS	Crew Swaps
CSV	Crew Sink Violations
DOC	Direct Operating Cost
DPC	Delayed Passengers Count
FTM	Maximum Additional Flight Time
IARM	Integrated Airline Recovery Model
IDS	Integrated Disruption Solver
IF	Infeasibilities
KPI	Key Performance Indicator
MAP	Mean Average Precision
MCP	Missed Connecting Passengers
MILP	Mixed-integer Linear Programming
ML	Machine Learning
ML IDS	Machine-learned Ranker in combination with the Integrated Disruption Solver
MRR	Mean Reciprocal Rank
NDCG	Normalised Discounted Cumulative Gain
RCU	Reserve Crews Used

Introduction

Mechanical failures, bad weather conditions or crew absence are all causes for disruptions in airline schedules. Almost 24% of all flights in Europe were delayed in the third quarter of 2017 as a result of disruptions [15]. Associated costs are the result of additional fuel expenses, crew overtime and passenger monetary compensation, which can have a significant impact on the airline business. Delay costs in the US airline industry equalled approximately $32.9 billion in 2007. $8.3 billion of the total delay costs were expenses for additional fuel, crew and maintenance [1]. In the event of a disruption, Airline Operations Control Centers (AOCC) must ensure the resumption of operations by resolving the disruption through the appropriate interventions. In the dynamic operational environment, any intervention is based on real-time decisions. Automated systems that support the decision-making, are therefore only useful if they provide fast, realistic and high-quality recovery solutions. That means that the computational complexity must be limited, all important resources should be considered and disruption costs need to be minimized.

In recent years, several studies on airline disruption management have been performed in the department of Air Transport Operations at the faculty of Aerospace Engineering, TU Delft. Vos et al. [14] started with developing a dynamic aircraft recovery model. Vink et al. [13] extended the work of Vos et al. [14] by using a heuristic to enable faster runtimes. Hassan [8] used machine learning classifiers instead of a heuristic to reduce the computational complexity and enable faster runtimes. Hoeben [9] developed a crew recovery model. Recently, Nikolajević [10] extended the work of Hassan [8] by adding the crew recovery model sequentially with a machine learning classifier. The computational complexity of the integrated recovery formulation and the lack of machine learning accuracy, forced the authors to use a sequential approach to obtain fast recovery times.

The last goal in the series of projects on airline disruption management is to develop an integrated airline recovery model, that is able to provide solutions in real-time. The use of machine learning in the previous projects efficiently decreased the computational complexity and also has the potential to make an integrated model tractable. The objective of this research is to develop a better-performing machine learning model in combination with an integrated recovery approach to realize a fast and more efficient disruption management model.

This project was carried out from September 2021 until July 2022 as a Master's thesis for the Air Transport Operations track at the Aerospace Engineering faculty. The main contributions of this research are the following:

- A tractable integrated airline recovery model, that recovers the aircraft and crew, and considers passenger missed connections.
- Machine-learned ranking algorithms that efficiently reduce the computational complexity of the model, resulting in real-time recovery solutions.

The thesis is divided into three main parts. Part I consists of the scientific article and is the main part of the report, containing background information, the problem description, methodology and experimental results. Part I ends with the conclusions and recommendations for future work. Part II consists of an extensive literature study, which was performed at the start of the project in order to discover the state-of-the-art in the context of airline disruption management and to find potential literature gaps. Part III contains supporting work and provides additional information not given in the scientific article.

I

Scientific Paper

A Real-Time Decision-Support Tool for the Integrated Airline Recovery using a Machine Learning Approach

Berend Eikelenboom,*

Delft University of Technology, Delft, The Netherlands

Abstract

Airlines frequently deal with unexpected disruptions, which have to be resolved in order to resume their operations again. Decision-support tools help airlines with disruption management. However, long computational times are associated with integrated recovery models that compute globally optimal solutions, which makes these tools unfit for large airlines. Faster sequential approaches have been proposed, which recover one resource at a time, but often only provide local optima. This paper presents a real-time decision-support tool to solve the integrated airline recovery problem. The model simultaneously recovers the schedule and allocates aircraft and pilot pairs to the flights, while minimizing additional incurred costs. Passengers are implicitly recovered by considering missed connections. Recovery actions include delaying and cancelling flights, swapping aircraft, and swapping, deadheading or using reserve crew. A machine-learned ranking algorithm reduces the computational complexity of the problem by selecting a subset of the resources that are likely to be involved in the recovery plan, such that only part of the network is considered. The decision-support tool was evaluated on disruption scenarios of one of the largest airlines in the world: Delta Airlines. The results show that the machine learning selection reduces the average computational time 15-fold compared to the integrated recovery model that uses the complete network, increasing the percentage of solutions computed under two minutes from 13% to 96%. The proposed model is able to find globally optimal solutions in 58% of the cases and yields similar results in terms of delays, but more flight cancellations in comparison with the globally optimal solutions. The proposed integrated model found a feasible solution to all disruption instances, while a benchmarked sequential model returned infeasible solutions in 4.4% of the cases. Besides, the sequential model produces 68% more flight cancellations, 4 times more aircraft sink node violations and twice as many crew sink node violations in comparison with the proposed integrated airline recovery model.

1 Introduction

Airline operations are often disturbed, which could damage the schedule in such a way that it becomes infeasible. Disruptions can have different causes, such as mechanical failures, bad weather conditions or crew absence. Walker (2017) showed that almost 24% of all flights in Europe were delayed in the third quarter of 2017 as a result of disruptions. Associated costs are the result of additional fuel expenses, crew overtime and passenger monetary compensation, which could have a significant impact on the airline business. Ball et al. (2010) estimated that the delay costs in the US airline industry were \$32.9 billion in 2007 of which \$8.3 billion were expenses for additional fuel, crew and maintenance. In order to resolve the disruption, the schedule should be recovered such that all flights can be operated again with the goal of minimizing factors like flight delays, flight cancellations and passengers missing their connection. As an airline possesses many resources that are critical for airline operations, taking into account all the resources is preferred, but makes the recovery more complex. Currently, Airlines Operations Control Center (AOCC) use fast, but not optimal, manual methods to recover the disrupted operations. Human specialists each focus on one resource (aircraft, crew or passengers) and combine the results to solve the problem sequentially (Castro and Ana Paula Rocha (2014)). However, since resources are interdependent, the sequential approach may not always result in optimal and feasible solutions. A resolved schedule may be optimal for one resource, while it might raise conflicts for other resources. The approaches that integrate the resources into one model produce globally optimal recovery solutions. However, these approaches have long computational times and are therefore too slow for real-time use. Generally, the requirement of airlines is to have a recovery plan in under two minutes because of the fast-changing operational environment in which operators are situated (Clausen et al. (2010)). The increase of combinations due to the interdependencies of resources and the large amount of recovery options makes the traditional integrated approach therefore not suited for airlines with large networks.

*Msc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

This paper proposes a real-time integrated airline recover model, that makes use of a machine-learned ranker in order to reach solutions in under two minutes. Often, only a limited number of resources are involved in the optimal recovery plan of an airline. The purpose of using a machine learning model is to efficiently select the resources likely to be involved in the recovery, such that a subnetwork can be constructed with these. By doing so, the recovery model does not have to optimize the airlines' whole network, which reduces the computational complexity. The proposed integrated recovery model explicitly recovers the schedule and allocates aircraft and pilots, and implicitly considers passenger flows to minimize avoidable missed connections.

The structure of the paper is as follows. In Section 2 a literature review is provided to give the reader background information. In Section 3 the problem is described and the scope of the research is defined. In Section 4, the methodology of the integrated recovery model is explained and a framework is provided. In Section 5, a case study on a dataset of Delta Airlines is performed and the associated results are presented. Lastly, in Section 6 the conclusion and recommendations for future work are provided.

2 Literature Review

Airline disruption management (ADM) has been a researched field since the 80s, because of its potential to ameliorate the service of airlines. Not only can an efficient disruption management strategy save airlines a lot of money, also passenger dissatisfaction can be mitigated. Hence, in an increasingly competitive environment, a decision-support tool for disruptions can be of real added value. Integrating aircraft, crew and passenger recovery into one model is the most challenging, as the complexity and solution space increase drastically with increasing networks. However, considering all these resources makes the recovery plan more accurate. Petersen et al. (2012) were one of the first who proposed an integrated problem formulation considering aircraft, crew and passengers. The authors used a Benders decomposition optimization technique to solve the problem. However, the runtime of the model was 30 minutes for a network containing 800 flights. Maher (2015) proposed a column and row generation technique to tackle the integrated airline recovery problem. The model required 45 minutes for a network of 262 flights. Zhu et al. (2016) proposed a sequential approach to obtain faster solutions to the airline recovery problem and also considered aircraft, crew and passengers. Although the authors mention that the model is able to solve problems in real-time, the computational time equals 180 seconds per 5-minute time stage. Hence, for longer recovery periods larger solution times should be expected. Arikan et al. (2017) tried to solve the integrated airline recovery problem with a conic quadratic mixed-integer linear programming model, such that the option of changing the cruise speed could be introduced (non-linear relationship with fuel and costs). The model was tested on a network with 1254 flights, but runtimes of around 20 minutes were required. Recently, Hassan (2018) and Nikolajević (2021) proposed a sequential model to obtain real-time solutions to the airline recovery problem and considered the aircraft, crew and passenger recovery as well. The authors noticed that only a few resources in the network were involved in the recovery. To reduce the computational complexity, they implemented machine-learned classifiers to predict the relevant resources and constructed a subnetwork with these resources. Although the model was able to solve disruptions of Delta Airlines in under two minutes, the model was prone to local optima and returned infeasible solutions in some cases.

Machine learning techniques have the potential to narrow down the airline recovery problem, such that real-time solutions can be obtained. However, the classifiers of Hassan (2018) and Nikolajević (2021) lacked accuracy and therefore could not be efficiently used to solve the integrated airline recovery problem. Recently, a less traditional machine learning technique has been proposed to tackle networked data. This technique is called machine-learned ranking and takes into account the global structure of problems. Instead of predicting the outcome of one data point, it takes a set of data points and ranks the data points within. This technique has already been proven efficient in other domains. Liu et al. (2015) applied a machine-learned ranker to protein remote homology detection and Ai et al. (2018) use the technique for information retrieval. However, to the best of the author's knowledge, these techniques have not been used in the airline industry yet. As the airline recovery problem concerns networked data, a machine-learned ranking approach has the potential to leverage this global structure and select the relevant resources more accurately.

3 Problem Description

The model proposed in this paper focuses on integrating the three most important resources, i.e. aircraft, pilots and passengers, in one recovery model. Doing so makes the decision-support model considerably more aligned with the real-life operational situation in comparison to models recovering only a single resource at a time. The objective is to optimize the new schedule and allocate both aircraft and pilot pairs simultaneously. Passenger flows are also considered, since the model penalizes decisions that result in passengers missing their connection. The disruption types considered in this research are flight delays, flight cancellations, aircraft unavailability and airport unavailability. The model computes recovery solutions from these disruptions by using a combination of several recovery actions. Related to the schedule, the model could delay or cancel flights if this is deemed necessary. Besides, swapping aircraft (i.e. operating a flight with an unscheduled aircraft) is another possible recovery option. Related to the pilots, swapping, deadheading (i.e. relocating pilots by flying crew to another airport as a passenger) and the use of reserve crew are recovery actions that could be taken.

Aircraft, pilots and passengers are considered the most important resources of an airline, because the former two are crucial for operating a flight and the latter experience the services of the airline. Additionally, pilots must follow strict regulations regarding flight, duty and rest times, which makes it difficult to manually create feasible schedules that meet all the labour constraints. Cabin crew are not considered in the model, as they are much more readily available and are less subject to stringent regulations, which makes it easier to allocate cabin crew to flights. Another reason for disregarding cabin crew is that they often propagate disruptions to many more flights, which increases the computational complexity of the model. This is because the cabin crew composition is likely to change during the duty, as the number of cabin crew depends on the number of passengers in the aircraft. On the other hand, pilot pairs often follow the same aircraft during their duty, meaning that disruptions affect a limited number of flights.

Furthermore, the operations of airlines also depend on maintenance availability, gate allocation, ground handling and air traffic control. But since these are less constraining and decrease the computational speed of the decision-support tool, these are not considered.

4 Methodology

This paper proposes an integrated airline recovery model that makes use of a machine learning technique to reduce the computational complexity, such that solution times could be achieved in less than 2 minutes. The disruption and the schedule with the aircraft and crew pairs are pre-processed and analysed by the machine-learned model, which ranks the resources from most to least relevant (i.e. the likelihood of the resource being involved in the optimal recovery solution). A subset consisting of the disrupted resources and the resources selected by the machine-learned ranker is created, with which a subnetwork is constructed. Finally, a mixed-integer linear programming model computes the optimal recovery solution to the disruption in the subnetwork. During post-processing, a visualisation of the recovered schedule is generated and key performance indicators (KPI's) are computed. An overview of the methodology is depicted in Figure 1. In the next sections, each module will be explained in more detail.

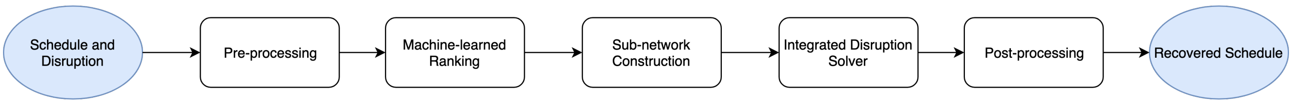

Figure 1: Methodology overview.

4.1 Pre-processing

In the pre-processing stage the schedule and disruptions are interpreted and all the associated costs are computed. The delay costs are based on the work of Cook et al. (2012) and consist of both hard costs and soft costs. The hard costs are defined as the legal compensation that the airline is obliged to return to the customer in case of delays (depending on the duration of the delay). The soft costs, on the other hand, are defined as the expenses not directly related to the repayment, but to the poor experience of the customer due to a delay or cancellation.

During pre-processing, the recovery scope is also determined. The time window indicates the period in which the recovery actions should take place. At the end of the time window, the disruptions should be resolved, such that the airlines' operations can be resumed again. The start of the time window is defined as the time at which the first disruption became known. The end of the time window is defined by the user. Either the airline chooses to specify the length of the time window or the end of the time window. The longer the time window, the more recovery options become available, which increases the computational complexity of the model.

Furthermore, the network is grouped per aircraft family for large airlines to reduce the computational complexity. Instead of recovering all aircraft and crew in the network, only the resources belonging to the same family as the disrupted aircraft are being considered. This implies that aircraft swaps cannot be performed within different families. However, this is not common, as the difference in characteristics of aircraft belonging to different families is often too large making it inefficient to swap. Similarly, crew swaps also become impossible within different families. But as pilots are trained for a specific family, they are not even allowed to operate other families. Examples of aircraft families are Boeing 737 and Airbus A320. Within aircraft families, typically different aircraft types exist, such as the 737-800 and 737 MAX. It should be noted that the differentiation is made between families and not between types.

Lastly, the connecting passenger matrix (CPM) is also constructed in this phase. The CPM is based on the research performed by Vink et al. (2020), who aimed to include passenger considerations as well without explicitly modelling them. Vink et al. (2020) developed a one-sided CPM, which computed the missed connection costs in the case a flight delay caused passengers to miss their connection. Hassan (2018) further extended the CPM, by allowing the connecting flights to be delayed as well if this does not disturb the downstream flights after it. The CPM is generated per flight that has a connection.
Although the connecting passenger matrix implicitly models the flow of passengers and is a great addition to the integrated recovery model, it also has a limitation. The CPM is one-sided in most cases. This means that it only considers the delay of the first flight with respect to the scheduled departure time of the connecting flight (i.e. it assumes that the connecting flight will not be delayed). However, it could happen that the second flight is also delayed due to an upstream disruption. In these cases, the CPM does not consider the imposed delay of the second flight and still counts with the initially scheduled departure time, potentially resulting in additional costs for no missed connection. The two-sided extension of Hassan (2018) fixes this by assigning negative costs to downstream flights which have to be delayed in order to avoid passengers missing their connections. However, this extension only applies to the cases where the first flight is the disrupted flight itself, as it is not computationally feasible to implement this for all flights in the network.

4.2 Machine-learned Ranking

In order to increase the computational speed, a selection of the resources is made before starting with the optimisation. The two-minute requirement of the AOCC cannot be satisfied for most airlines while considering their full network in the integrated optimisation model. Both an aircraft and crew selection is made, with two distinct machine-learned ranking models. These models aim to rank the resources from most relevant to irrelevant in a supervised manner, based on the features describing them. In this context, relevancy means the likelihood that the resource is involved in the recovery through a recovery action (e.g. delayed or swapped). Hereafter, the most relevant resources are selected with which a subnetwork is constructed. The number of resources in the subnetwork should be calibrated in such a way that the total recovery time is limited to under 120 seconds.

Several ranking approaches exist, including pointwise, pairwise and listwise approaches. The pointwise approach reduces to a traditional classification problem. All the data points are independently given a score based on their own features, after which the data points are sorted in a list from high to low scores. The group structure of ranking is neglected in this approach as the data points are not compared to each other in one way or another. In the pairwise approach, pairs of data points are compared with each other. This is done for all data points in the group, such that a global ranking can be achieved at the end of the process. Lastly, the listwise approach uses the complete group structure to rank the data points. The approach directly takes an entire list of data points as an instance and tries to come up with the optimal ordering of it.

Algorithms
As pointwise ranking reduces to a traditional classification problem, the focus will be on the pairwise and listwise approaches. A challenge organised by Yahoo was aimed to benchmark learning to rank algorithms on real-world large datasets (Chapelle and Chang (2011)). The datasets consisted of a subset of the training set used internally by Yahoo to train their own web search engines and included 883,000 data points and 700 features. Burges et al. (2011) won the challenge and used a linear combination of 12 learning to rank algorithms, consisting of 8 LambdaMART, 2 LamdaRank neural networks and 2 logistic regression models. The former two are well-known open-source algorithms developed by Microsoft (Burges (2010)). Next to LambdaRank and LambdaMART, Microsoft first developed RankNet.

- RankNet: This model was first developed by Microsoft and makes use of neural networks. The loss function tries to minimize the number of inversions in ranking, with an inversion meaning an incorrect ordering among pairs. RankNet optimizes the loss function using a stochastic gradient descent in a neural network system.
- LambdaRank: This model was developed based on RankNET, but only uses the gradient (λ, lambda) of the loss, instead of the loss itself. These gradients are attached to the data points and indicate the

direction where the data point has to go (more relevant or less relevant).

- LambdaMART: The last model combines LambdaRank and MART (Multiple Additive Regression Trees). The result is a gradient boosted decision tree with a loss function derived from LambdaRank to perform pairwise ranking. During tests performed by Microsoft, LambdaMART has shown better results than both RankNet and LambdaRank.

Since LambdaMART is the most recently developed algorithm that has shown the best results and was used by Burges et al. (2011) who won the Yahoo challenge, this research uses LambdaMART as a starting point.

Features

The potential of aircraft and crew to help resolve the disruption, depends on different pieces of information, such as the characteristics of the disrupted aircraft, the characteristics of the candidate aircraft, the schedule and information related to the disruption itself. The final feature set consists of 46 features for the aircraft model and 126 for the crew model, and is based on the work of Hassan (2018) and Nikolajević (2021). The ten most important aircraft features are depicted in Table 1. The feature importance is based on the usefulness of the feature in predicting the outcome. All the remaining aircraft features used in the model are presented in Table 10 in Appendix A.

AC Feature	Description
c_ground_time_d_dest_airport	The total time the candidate aircraft spends on the ground at the destination airport of the disrupted aircraft.
c_ground_time_d_orig_airport	The total time the candidate aircraft spends on the ground at the origin airport of the disrupted aircraft.
d_dest_airport_min_after_min	The minimum number of minutes the candidate aircraft is on the destination airport of the disrupted flight after the STA of the disrupted flight.
d_orig_airport_min_after_min	The minimum number of minutes the candidate aircraft is on the origin airport of the disrupted flight after the STD of the disrupted flight.
disruption_duration	The duration of the disruption in minutes.
c_range_vs_d_range_max	Candidate aircraft range - disrupted aircraft range.
d_range_max	Maximum flight distance of the flight string of the disrupted aircraft.
c_flights_duration	The sum of the flight durations for the flights scheduled for the candidate aircraft.
c_econ_lf_mean	The mean economy passenger load factor of all flights scheduled for the candidate aircraft.
c_econ_lf_std	The standard deviation of the economy passenger load factor of all flights scheduled for the candidate aircraft.

Table 1: Most important aircraft features

The ten main crew features are depicted in Table 2. The remaining crew features are based upon the main features below and are presented in Table 11 Appendix A. The critical time, embedded in many features, represents the time and location where a flight's originally scheduled crew will not be able to operate it due to the disruption. A difference in structure can be noted between the two feature sets, as the aircraft features are more comprised of integer and floating values, while many crew features are binary (i.e. true or false).

Crew Feature	Description
time_end_crew	Duty start time.
time_start_crew	Duty end time.
tw_start	TW start time in minutes from midnight.
tw_end	TW end time in minutes as sum of TW start time and TW length.
crit_time	Critical time as measured in minutes after the start of the TW.
c_at_crit	True if crew is at critical location at critical time.
c_at_org	True if candidate crew is at origin of disrupted flight.
c_at_dest	True if candidate crew is at destination of disrupted flight.
future_fl_end	True if crew has future flight to end airport of disrupted crew.
future_fl_next	True if crew has future flight to destination of disrupted flight.
reserve_crew	True if candidate is reserve crew.

Table 2: Main crew features

Training and Evaluation

The learning to rank algorithms are trained with data from solved disruption instances. An aircraft recovery model developed by Hassan (2018) and a crew recovery model developed by Nikolajević (2021) were used to find the optimal solutions to many disruptions instances. The models determined which aircraft and crew pairs were involved in the recovery and computed the features of each individual resource. Hence, with training data containing the features and a binary label indicating the involvement of each resource, the machine learning algorithm is able to learn what the relevant features are to identify helpful resources in disruption instances not yet solved.

Choosing a specific model is realized by comparing their performance in terms of an evaluation metric that fits the purpose of this research. Binary classification evaluation metrics are well-suited for this use case, as the ranking model eventually returns two classes: the selected resources and the non-selected resources. Most binary metrics consider combinations of the true positives (TP), false positives (FP), true negatives (TN) and false negative (FN).

Metric	Formula	Description
Recall	$\frac{TP}{TP+FN}$	The fraction of relevant resources that are selected.
Precision	$\frac{TP}{TP+FP}$	The fraction of selected resources that are relevant.
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	The fraction of correct predictions for all the resources.
Specificity	$\frac{TN}{TN+FP}$	The fraction of irrelevant resources that are kept outside the selection.

Table 3: Evaluation Metrics for Binary Classification

As networks of larger airlines commonly consist of hundreds of aircraft and crew, and only a few resources will be helpful in the recovery, the data will be imbalanced. Because of this, accuracy and specificity seem to be inappropriate metrics. The large number of irrelevant resources will falsely insinuate the good performance of the model. Besides, the changing number of irrelevant resources per disruption instance will give inconsistent and meaningless results. Precision assesses the fraction of selected resources that are relevant, but since the number of relevant resources changes per disruption instance, the metric will also be meaningless. Consider for example a selection size of 20 aircraft, a disruption instance with only one relevant aircraft, correctly identified and an instance with two relevant aircraft, also correctly identified. In both cases the model has done a perfect job, since all relevant aircraft are predicted correctly. However, the precision of the second will be higher than the precision of the first. The recall evaluation metric only considers the relevant resources and is very much suited for this use case. It assesses the fraction of all the relevant resources (i.e. the resources that are involved in the recovery and are included in the selection). This evaluation metric will give a good indication of the performance of the model, regardless of the number of relevant and irrelevant resources in the disruption instances. The recall does increase with increasing selection sizes, thus the recall should always be evaluated at a fixed group size when fairly comparing different models and parameters with each other.

Hyperparameter Optimisation

To increase the performance of the model, the hyperparameters (which define the construction of the model) of machine-learned ranker should also be optimized. Bayesian optimisation is one of the most structured and efficient methods to optimise the hyperparameters. The technique applies a probabilistic surrogate model that tries to approximate an objective function, based on known function values (i.e. the evaluations of models with certain hyperparameters). From the probabilistic model, also called a prior, a posterior distribution is created, which interpolates the known function values with the information gained from the prior. The posterior constructs an acquisition function which provides an indication for the next sample to pick. This process is repeated each time a new sample is evaluated, such that more information can be used to predict the objective function. Generally, a Gaussian process is used for the prior and a method called Kriging, or Gaussian process regression, is used to construct the posterior. The method uses the prior covariances from the Gaussian process to perform the interpolation. The blue line in Figure 2 shows the Kriging method, while the red dashed line depicts a smooth spline. The acquisition function commonly aims to maximise the expected improvement of the evaluation of new samples $f(x)$ in comparison to the best-evaluated sample thus far to find the best next sample to pick.

$$EI(x) = \mathbb{E}(\max(f(x) - \hat{f}, 0)) \text{ with } \hat{f} \text{ the maximum value of } f \text{ thus far} \tag{1}$$

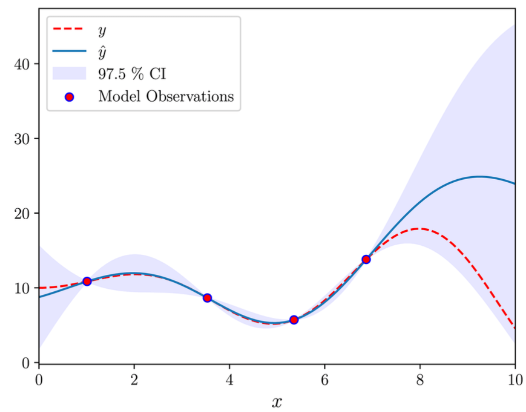

Figure 2: Kriging, creating the posterior from the prior.
Adapted from Huchet et al. (2019).

A good practice during hyperparameter optimisation is to use k-fold cross-validation. The initial training dataset is split into k folds, where $k-1$ folds are used for training and 1 fold is used for validation. A model with certain hyperparameters is trained and evaluated on all the different combinations of training and evaluation folds, such that a more accurate, unbiased analysis can be done when comparing the different hyperparameters with each other. The evaluations of the model with certain hyperparameters could be averaged to get an indication of its performance, but one could also look at the standard deviation to assess its variance on varying datasets. An example of a 3-fold cross validation is given in Figure 3.

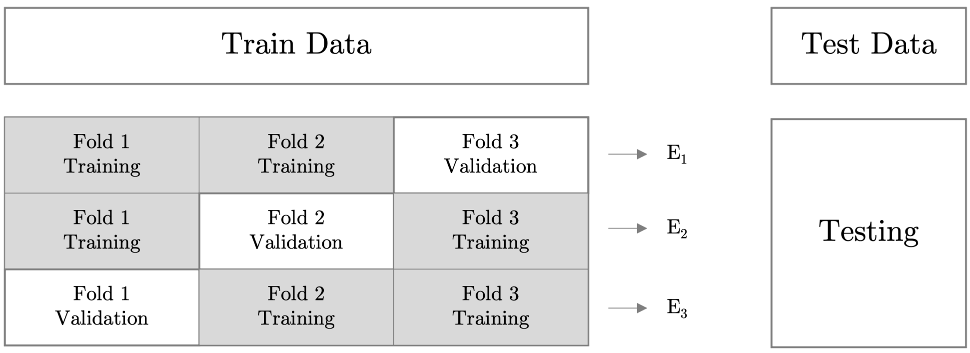

Figure 3: 3-fold cross-validation

4.3 Subnetwork Construction

After the machine learning algorithm has selected the resources, flights should be selected as well in order to create a time-space subnetwork. It is not sufficient to only include the flights operated by the initially selected resources, as other flights could be relevant as well. Consider the schedule in Figure 4. If the initial selection would comprise *AC 1* and *Crew 2*, the flights operated by these resources would be the flights from and to *ATL* and *NY* (i.e. the blue arcs). However, as *Crew 1* has a second flight going to *LA* (i.e. the green arc), this next flight should also be considered. If not, the flight from *ATL* to *NY* has the freedom to be delayed past the scheduled departure time of this second flight (e.g. the orange arc) and hence *Crew 1* would not make its connection. Therefore, it is crucial to consider all the downstream flights of all the resources operating the initially selected flights. Besides, not only the additional flights, but also the additional resources, not yet included, on these flights should be added to the selection. In the example given, not only the flight from *NY* to *LA* should be considered, but also *AC 2* operating this flight. If not, resource allocation could be infeasible due to a shortage of aircraft and crew pairs.

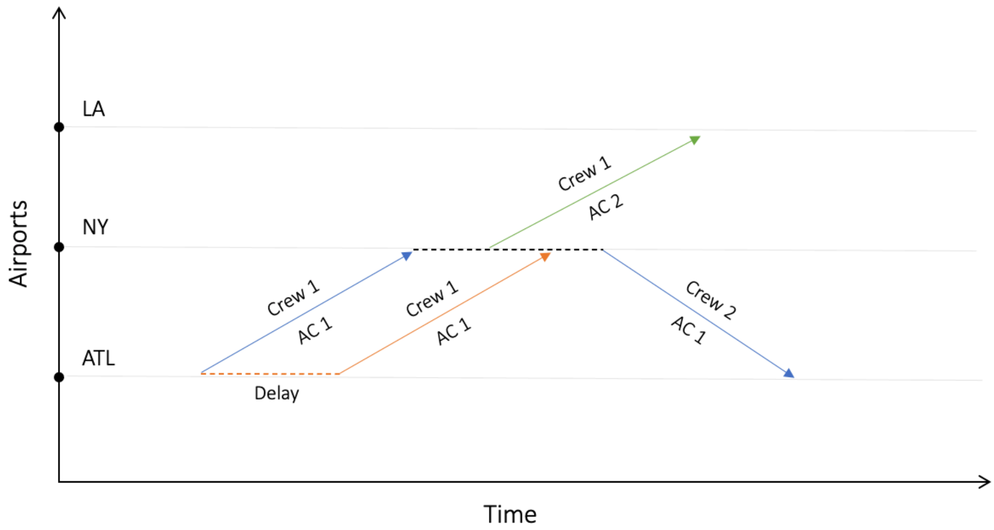

Figure 4: Example incomplete selection

An iterative selection algorithm is implemented that includes all the relevant flights and additional resources in the final selection, given the initial selection determined by the machine-learned ranker. A flowchart that demonstrates this process is depicted in Figure 5.

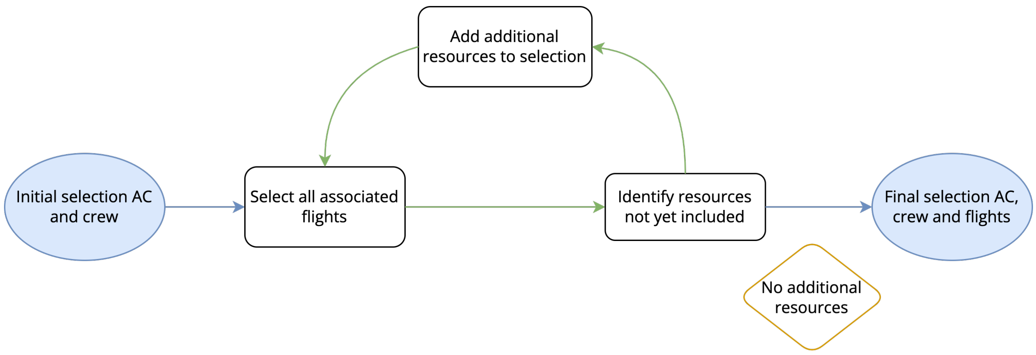

Figure 5: Selection algorithm

The size of the final selection eventually determines the computational speed. If the final selection is too large, the initial selection should be reduced and the iterative selection algorithm should be called again. This process should be repeated until the size of the final selection is appropriate.

Parallel time-space networks, introduced by Thengvall et al. (2003), are constructed. In this way, each resource can be tracked individually, making it easy to implement origin and sink node constraints, crew labour regulations and maintenance activities. All nodes in the network correspond to a unique combination of airport and time. Resources are able to get from one node to the other with the use of two different arcs: flight arcs and ground arcs. Ground arcs connect nodes at the same airport and are used by aircraft and crew that stay on the ground at a specific airport. Flight arcs, on the other hand, are the arcs that connect the nodes not at the same airport and represent both the scheduled and delayed flights.

4.4 Integrated Disruption Solver

The integrated disruption solver (IDS) is a mixed-integer linear programming (MILP) optimisation model that attempts to find the optimal solution to one or multiple disruptions in a given time window. The model optimises the recovery of the schedule and allocates aircraft and pilot pairs to flights simultaneously. Connecting passengers are implicitly included by the connecting passenger matrix (CPM), which imposes an appropriate cost in the objective if passengers cannot make their next flight.

The sets, indices, parameters and decision variables below are used to construct the objective function and the constraints. The origin nodes represent the airport and time at which each resource starts in the time window. From this node, the resources can proceed through the time window via the intermediate nodes until they reach their sink node. The sink nodes are modelled differently for aircraft and crew. Each individual crew has its own sink node, which should be respected. However, aircraft follow a more flexible approach and are not restricted to a specific end airport. Having another aircraft tail of the same type at the sink node is also permitted. This makes sure that more aircraft swap options are possible, which could reduce the disruption costs. Because of the human factor of pilots, this is not allowed for the crew.

Sets and Indices

Sets		*Indices*	
F	flights	i	flight index
K	crews	k	crew index
A	airports	t	delay time index
E	aircraft types	a	airport index
P	aircraft	p	aircraft index
P(e)	aircraft p of type e	e	aircraft type index
N	nodes $= N_O \cup N_I \cup N_S$	n	node index
N_O	origin nodes	j	artificial variable index
N_I	intermediate nodes		
N_S	sink nodes		
T	delay steps		

Parameters

Aircraft		*Crew*	
$C_{OP_{p,i}}$	Operating cost of AC p on flight i	$C_{OP_{k,i}}$	Operating cost of crew k on flight i
$C_{D_{i,t}}$	Delay cost of flight i for delay t	$C_{DH_{k,i}}$	Deadhead cost of crew k on flight i
C_{C_i}	Cancellation cost for flight i	C_{OC}	Unscheduled crew operating penalty
C_{G_n}	Cost of ground arc from node n	C_{CSV_k}	Sink node violation cost for crew k
h_n^e	Number of AC of type e required at node n	C_{FT}	Flight time exceeded penalty
$C_{C_{SCH}}$	Unscheduled AC operating penalty	FT_i	Flight time of flight i
C_{ASV_k}	Sink node violation cost for aircraft k	FTL_k	Flight time remaining in TW for crew k
		FTM_k	Maximum additional flight time for crew k

Decision Variables

Aircraft		*Crew*	
$\delta_{F_{p,i}}$	if flight i is flown by AC p without delay	$\delta_{K_{k,i}}$	if crew k is allocated to flight i without delay
$\delta_{FD_{p,i,t}}$	if flight i is flown by AC p with delay t	$\delta_{KD_{k,i,t}}$	if crew k is allocated to flight i with delay t
δ_{C_i}	if flight i is cancelled	$\delta_{GK_{k,n}}$	if crew k uses ground arc n
$\delta_{GP_{p,n}}$	if AC p uses ground arc n	$\delta_{K'_i}$	if flight i is flown by an unscheduled crew
$\delta_{F'_i}$	if flight i is flown by an unscheduled AC	$\delta_{DH_{k,i}}$	if crew k is deadheaded on flight i without delay
s_j	slack variable for sink constraint violation	$\delta_{DHD_{k,i,t}}$	if crew k is deadheaded on flight i with delay t
		s_k	slack variable for sink constraint violation
		s_{FT_k}	slack variable for exceeding scheduled flight time

Objective Function

$$\begin{aligned}\text{Min} \sum_{p\in P}\sum_{i\in F} C_{OP_{p,i}} \cdot \delta_{F_{p,i}} + \sum_{p\in P}\sum_{i\in F}\sum_{t\in T}\left(C_{OP_{p,i}} + C_{D_{i,t}}\right)\cdot \delta_{FD_{p,i,t}} + \sum_{i\in F} C_{C_i}\cdot\delta_{C_i} + \sum_{p\in P}\sum_{n\in N} C_{G_n}\cdot \delta_{GP_{p,n}} + \sum_{i\in F} C_{C_{SCH}}\cdot\delta_{F'_i} \\ + \sum_{k\in K}\sum_{i\in F}\left(C_{OP_{k,i}}\cdot\delta_{K_{k,i}} + C_{DH_{k,i}}\cdot\delta_{DH_{k,i}} + \sum_{t\in T}\left(C_{OP_{k,i}}\cdot\delta_{KD_{k,i,t}} + C_{DH_{k,i}}\cdot\delta_{DHD_{k,i,t}}\right)\right) + \sum_{k\in K}\sum_{n\in N} C_{G_n}\cdot\delta_{GK_{k,n}} + \sum_{i\in F} C_{OC}\cdot\delta_{K'_i} \\ + \sum_{j\in S} s_j\cdot C_{ASV} + \sum_{k\in K} C_{CSV}\cdot s_k + \sum_{k\in K} C_{FT}\cdot s_{FT_k} \end{aligned} \tag{2}$$

The objective function is a minimisation problem and consists of reducing aircraft, crew and passenger related costs. The first line represents the aircraft and passenger related costs, consisting of the direct operating costs (DOC), delay costs, cancellation costs, the cost of operating a ground arc and the additional cost of operating a flight with an unscheduled aircraft. The passenger missed connection costs are implicitly modelled in CPM, which is incorporated in the delay costs. The second line represents the crew related costs, consisting of the operating and deadheading cost, the cost of operating a ground arc and the additional cost of operating a flight with unscheduled crew. The last line consists of slack variables to ensure feasibility. The first refers to the aircraft sink node violation, the second to the crew sink node violation and the last one to the crew flight time violation. If the aircraft and crew sink node constraints or the flight time constraints cannot be satisfied, these slack variables are activated and impose a penalty in the objective function.

Constraints

$$\delta_{C_i} + \sum_{p \in P} \left(\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} \right) = 1 \quad \forall i \in F \tag{3}$$

$$\sum_{p \in P} \delta_{F_{p,i}} = \sum_{k \in K} \delta_{K_{k,i}} \quad \forall i \in F \tag{4}$$

$$\sum_{p \in P} \delta_{FD_{p,i,t}} = \sum_{k \in K} \delta_{KD_{k,i,t}} \quad \forall i \in F, \forall t \in T \tag{5}$$

$$\sum_{p \in P} \delta_{F_{p,i}} \geq \sum_{k \in K} \delta_{DH_{k,i}} \quad \forall i \in F \tag{6}$$

$$\sum_{p \in P} \delta_{FD_{p,i,t}} \geq \sum_{k \in K} \delta_{DHD_{k,i,t}} \quad \forall i \in F, \forall t \in T \tag{7}$$

$$\delta_{GF_{p,n}} + \sum_{i \in F_{\text{out}}} \delta_{F_{p,i}} + \sum_{i \in F_{\text{out}}, t \in T} \delta_{FD_{p,i,t}} = 1 \quad \forall p \in P, n = \text{scheduled } N_o \text{ of } p \tag{8}$$

$$\left(\delta_{G_{p,n-1}} + \sum_{i \in F_{in}} \delta_{F_{p,i}} + \sum_{i \in F_{in}, t \in T} \delta_{FD_{p,i,t}} \right) - \left(\delta_{G_{p,n}} + \sum_{i \in F_{\text{out}}} \delta_{F_{p,i}} + \sum_{i \in F_{\text{out}}, t \in T} \delta_{FD_{p,i,t}} \right) = 0 \quad \forall p \in P, n \in N_i \tag{9}$$

$$\sum_{p \in P(e)} \left(\delta_{GF_{p,n-1}} + \sum_{i \in F_{in}} \delta_{F_{p,i}} + \sum_{i \in F_{in}, t \in T} \delta_{FD_{p,i,t}} \right) + s_j \geq h_n^e \quad \forall e \in E, n \in N_s \tag{10}$$

$$\delta_{GK_{k,n}} + \sum_{i \in F_{\text{out}}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}} \right) \sum_{i \in F_{\text{out}}, t \in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}} \right) = 1 \quad \forall k \in K, n = \text{ scheduled } N_o \text{ of } k \tag{11}$$

$$\left(\delta_{GK_{k,n-1}} + \sum_{i \in F_{in}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}} \right) + \sum_{i \in F_{in}, t \in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}} \right) \right) -$$
$$\left(\delta_{GK_{k,n}} + \sum_{i \in F_{\text{out}}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}} \right) + \sum_{i \in F_{\text{out}}, t \in T} \delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}} \right) = 0 \quad \forall k \in K, n \in N_i \tag{12}$$

$$\delta_{GK_{k,n-1}} + \sum_{i \in F_{in}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}} \right) + \sum_{i \in F_{in}, t \in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}} \right) + s_k = 1 \quad \forall k \in K, n = \text{ scheduled } N_s \text{ of } k \tag{13}$$

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p, i \text{ where } (\text{SeatsY}_p < \text{PaxY}_i \wedge \text{SeatsJ}_p < \text{PaxJ}_i) \tag{14}$$

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p, i \text{ where } \left(\text{range}_p < \text{dist}_i \right) \tag{15}$$

$$\delta_{F_{p,i}} + \sum_{t\in T} \delta_{FD_{p,i,t}} - \delta_{F'_i} = 1 \quad \forall i \in F, p = \text{ aircraft not scheduled for } i \tag{16}$$

$$\delta_{F_{p,i}} + \sum_{t\in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p \in P, i \text{ where } STD_i - T_{now} < T_{swap} \text{ and i} \neq \text{ flight for p} \tag{17}$$

$$\delta_{K_{k,i}} + \sum_{t\in T} \delta_{KD_{k,i,t}} - \delta_{K'_i} = 1 \quad \forall i \in F, k = \text{ crew not scheduled for } i \tag{18}$$

$$\sum_{i\in F} \left(\delta_{K_{k,i}} + \sum_{t\in T} \delta_{KD_{k,i,t}} \right) \cdot FT_i \leq FTL_k + FTM_k \cdot s_{FT_k} \quad \forall k \in K \tag{19}$$

Constraints 3 ensure that all flights are either flown as scheduled, delayed or cancelled. Constraints 4 ensure that all the on-time flights are allocated a crew pair and Constraints 5 ensure that all the delayed flights are operated by a crew pair. Constraints 6 and 7 prohibit deadheading on cancelled flights and ensure that deadheading can only happen on operated flights. Aircraft node continuity is ensured by the three expressions underneath. Constraints 8 force all aircraft to leave the first node in the time window, either by flying a (delayed) flight or by utilizing a ground arc. Constraints 9 demand all aircraft entering a node to also leave that node. Constraints 10 handle the inflow of the aircraft at the sink node in the time window by fixing the aircraft type at a certain airport. To avoid model infeasibility, a slack variable with a large associated cost is activated in the objective function whenever the constraints cannot be satisfied. The following three expressions manage the crew continuity. Constraints 11 make sure that all crew pairs leave their starting node either by operating or deadheading a (delayed) flight or by utilizing a ground arc. Similar to Constraints 9, Constraints 12 demand all crew pairs entering a node to also leave that node. Lastly, Constraints 13 fix specific crew pairs at the sink node. If this cannot be assured, the slack variable is activated and imposes a large additional cost in the objective function. Again, this is done to avoid model infeasibility whenever the constraints cannot be satisfied. Besides the constraints related to the time-space network, aircraft and airline constraints are also imposed. Constraints 14 make sure that the aircraft that do not satisfy the seat capacity requirement cannot operate the flight. Constraints 15 prohibit an aircraft from operating a flight when its range is less than the distance of the flight. When a tail has been swapped and an unscheduled aircraft operates a flight, Constraints 16 ensure that a penalty is incurred in the objective function. Constraints 17 make sure that aircraft p not scheduled for flight i cannot operate the flight T_{swap} minutes before its time of departure. Constraints 18 impose a penalty in the objective function when a flight is operated by a different crew pair. Lastly, Constraints 19 ensure that crew pairs cannot exceed their maximum flight time during the recovery. If these constraints are violated, a slack variable will be activated to enlarge the crew's maximum flight time, such that the constraints become feasible again. However, this leads to the addition of a penalty in the objective function.

Disruptions

$$\sum_{p\in P} \delta_{F_{p,i}} + \delta_{D_{p,i,t}} = 0 \quad \forall t \in T \leq \text{ delay, } i = \text{ delayed flight} \tag{20}$$

$$\delta_{C_i} = 1, \quad i = \text{ cancelled flight} \tag{21}$$

$$\begin{gathered} \delta_{Fp,i} + \sum_{i\in F} \delta_{D_{p,i,t}} = 0 \\ \forall i \in F \text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i \leq \mathrm{t_{end}} \cup \mathrm{t_{start}} \leq \mathrm{STA}_i \leq \mathrm{t_{end}}), \\ \forall t \in T \text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i + t \leq \mathrm{t_{end}} \cup \mathrm{t_{end}} \leq \mathrm{STA}_i + t \leq \mathrm{t_{end}}) \end{gathered} \tag{22}$$

$$\begin{gathered} \sum_{p\in P} \left(\delta_{Fp,i} + \sum_{i\in F} \delta_{D_{p,i,t}} \right) = 0 \\ \forall i \in F \,(\text{where } (\mathrm{t_{start}} \leq \mathrm{STD}_i \leq \mathrm{t_{end}} \cup \mathrm{t_{start}} \leq \mathrm{STA}_i \leq \mathrm{t_{end}})) \cap (\text{ orig }_i \cup \text{ dest }_i) = \mathrm{a} \\ \forall t \in T \,(\text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i + t \leq \mathrm{t_{end}} \cup \mathrm{t_{end}} \leq \mathrm{STA}_i + t \leq \mathrm{t_{end}})) \cap (\text{ orig }_i \cup \mathrm{dest}_i) = \mathrm{a} \\ \text{where a = unavailable airport} \end{gathered} \tag{23}$$

The actual disruptions should also be included in the MILP model, as otherwise the schedule would be operated as planned. Disruptions are added to the optimisation model in the form of constraints, which cancel the relevant decision variables. When a flight is delayed, Constraints 20 make sure that all flight arcs up until the length of the delay cannot be flown. When a flight is cancelled, Constraints 21 set the cancellation decision variable for that flight equal to one. Another type of disruption is the unavailability of an aircraft for a certain period of time. This could be caused by a mechanical failure, for example. Constraints 22 ensure that all flight arcs up until the duration of the unavailability for the specific aircraft cannot be flown. The last form of disruption considered in this research is the unavailability of an airport. Constraints 23 prohibit the operation of all flights from and to the unavailable airport in the time period.

4.5 Post-processing

After optimisation, the decision variables are interpreted to understand the recovery actions that should be taken. A time-space network of the recovered schedule is created and visualised in a user-friendly interface, displaying the changes with respect to the original schedule. All recovery actions are highlighted such that the airline can quickly react to the disruption(s) that have taken place. Furthermore, KPI's are calculated to get an overview of how the model recovers the schedule and to be able to compare this model to other airline recovery models. The list with KPI's calculated is displayed in Table 4.

KPI	**Unit**	**Description**
Solution Time	seconds	The computational duration of the schedule recovery
Disruption Cost	€	The cost of the disruption(s)
Delayed Passengers	#	The number of passengers delayed
Average Delay	minutes	The average delay per delayed passenger
Passengers Cancelled Flights	#	The number of passengers on cancelled flights
Cancelled Flights	#	The number of cancelled flights
Passengers Missed Connections	#	The number of passengers that miss their connection
Aircraft Swaps	#	The number of aircraft swaps
Crew Swaps	#	The number of crew swaps
Crew Deadheads	#	The number of crew deadheads
Aircraft Sink Violations	#	The number of aircraft violating the sink node constraint
Crew Sink Violations	#	The number of crew pairs violating the sink node constraint
Crew Flight Hour Violations	#	The number of crew pairs violating the flight hour constraint
Reserve Crew Used	#	The number of reserve crew used

Table 4: KPI's

5 Experimental Results

The airline tested in the case study is Delta Airlines, which is the biggest airline in the world by revenue, profit, assets and market capitalization (Forbes (2022)). It is an American hub & spoke airline, with 8 hubs and a total of 242 destinations in 52 countries, as of 2022. The dataset that is used in this research contains roughly one month of operations in January 2015. In that year, the airline operated a fleet of 800 aircraft consisting of Boeing 717, Boeing 757, Boeing 767, McDonnel Douglas MD-88 and McDonnel Douglas MD-90 (Both MD-88 and MD-90 were phased out in 2020) and performed roughly 2,400 domestic flights per day. Since the computational speed of recovery models are susceptible to large networks, Delta Airlines is a good candidate for performing the analyses. Obtaining positive results for this airline means that the model is likely to perform well on disruptions from other airlines as well. Part of the dataset was obtained and modified by Hassan (2018) and included information about the schedule, aircraft and passengers. However, due to the competitive landscape and difficulties in long-term crew scheduling, airlines generally do not publish their crew rosters. Nikolajević (2021) generated this artificially and added the crew roster to the initial dataset created by Hassan (2018).

5.1 Machine-learned Ranker

A study is performed on the Delta Airlines case. In order to train the machine-learned ranking algorithms, an aircraft recovery model developed by Hassan (2018) and a crew recovery model developed by Nikolajević (2021) were used to find the optimal solutions to many disruptions instances. The integrated recovery model was not used, since the large computational times and RAM size limitations made it impossible to obtain enough globally optimal recovery solutions in the time-frame of this research. The dataset consists of 1,817 disruption instances solved by the aircraft recovery model and 7,750 disruption instances solved by the crew recovery model, with which the two machine learning algorithms are trained and evaluated. All features explained before are included in the dataset and a binary label is given to resources denoting their involvement in the recovery. Two

LambdaMART models are trained and optimized with the Bayesian method for both aircraft and crew. The hyperparameters chosen for the models are depicted in Figure 5.

Parameter	**Aircraft**	**Crew**
Learning rate	0.013	0.055
N_estimators	245	310
Number of leaves	30	35
Maximum depth	30	44
Subsample	0.75	0.78
Truncation level	70	76

Table 5: Results hyperparameter optimization LambdaMART

With these hyperparameters, the models are tested on unseen data. The recall is evaluated at different selection sizes both for aircraft and crew. The results are compared with the classifiers used in the SDSS, i.e. a random forest classifier for the aircraft selection and XGBoost for the crew selection, and are depicted in Figure 6 and Figure 7 respectively.

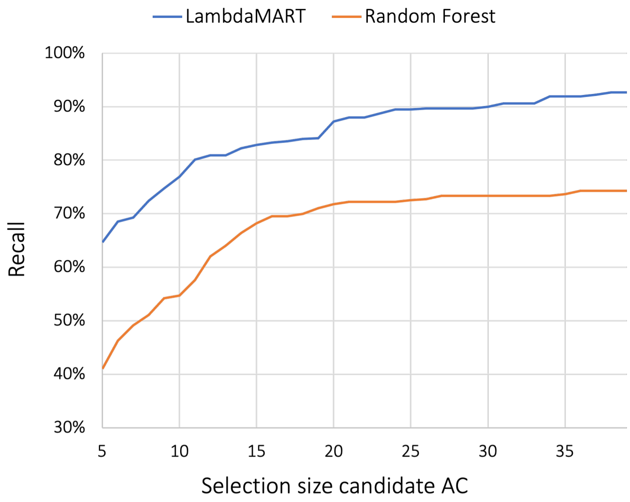

Figure 6: Recall comparison with increasing selection sizes for aircraft machine learning models.

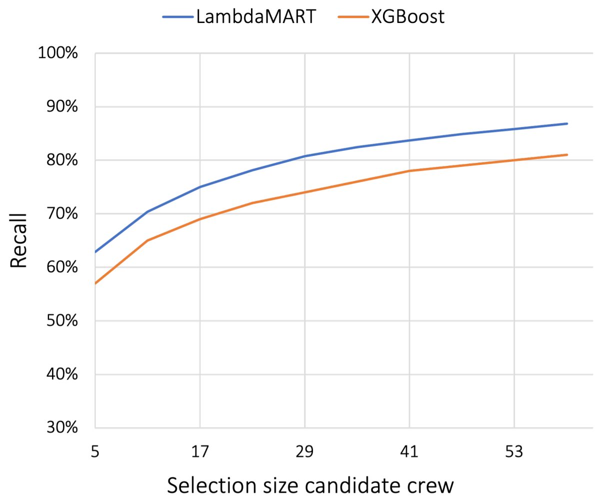

Figure 7: Recall comparison with increasing selection sizes for crew machine learning models.

The results show that both the aircraft and crew LambdaMART models outperform the random forest and XGBoost classifiers respectively. In terms of aircraft, the difference is more significant, since a recall increase of $15 - 25\%$ is realized by using a learning to rank algorithm. The crew models perform more similar, but still an increase of $4 - 8\%$ can be seen.

As the relevancy of resources is expressed in binary terms, recall is a well-suited evaluation metric. However, it does not completely capture the overall performance of the models. This is because no differentiation is made between the irrelevant resources, whilst these resources also have mutual differences. Some resources not in the optimal solution could be a lot more helpful than others, whilst they are both classified with the over-simplified binary label 0. Hence, a model picking the second-best resources or the worst are significantly different, but are both evaluated as bad by the recall metric. To completely understand the performance of the machine-learned ranker, the results of the airline recovery model, with the subnetwork created by the model, should be assessed.

5.2 Integrated Disruption Solver

The integrated disruption solver in combination with the machine-learned ranker will be called the ML IDS and is the proposed model in this research. In order to assess the performance of this model, a comparison is made with the globally optimal solutions retrieved from the integrated disruption solver that considers the full network (without the machine-learned ranker). The last-mentioned model will be referred to as the IDS.

Model Parameters

Several parameters have to be set to configure the model. These are not fixed and can be changed according to the preferences of the airline. The parameters that are used in this case study are depicted in Table 6 and are broken down into three different categories, i.e. the schedule, aircraft and crew model parameters. The delay costs are based on the work of Cook et al. (2012) and are equal to the costs used by Vink et al. (2020) (shown in Figure 8). The names of the costs are defined in Section 4. Aircraft sink violations are a lot more expensive than crew sink node violations, because of the scarcity of the resource. Furthermore, reserve crew sink violations are cheaper, since the strategy of having reserve crew is that they can easily be used in case of unavailable crew. Lastly, crew swapping is more expensive than aircraft swapping, because of the human factor. The cost distribution is chosen in such a way that the most desired recovery actions are more incentivized by the model.

Schedule	**Value**	**Aircraft**	**Value**	**Crew**	**Value**
TW Length	10 hours	C_c	\$250	C_{OC}	\$2,000
Time Step	10 minutes	C_{csch}	\$1,000	C_{FT}	\$20,000
Max Delay	4 hours	C_{ASV}	\$1,000,000	C_{CSV}	\$50,000
Swap Limit	3 hours	Sink Constraint	per AC Type	C_{SV_R}	\$10,000
Business Multiplier	3×	Selection	≤60	C_{DH}	\$200
				Sink Constraint	Per Crew
				Selection	≤65
				FTM	2h

Table 6: Schedule, aircraft and crew model parameters.

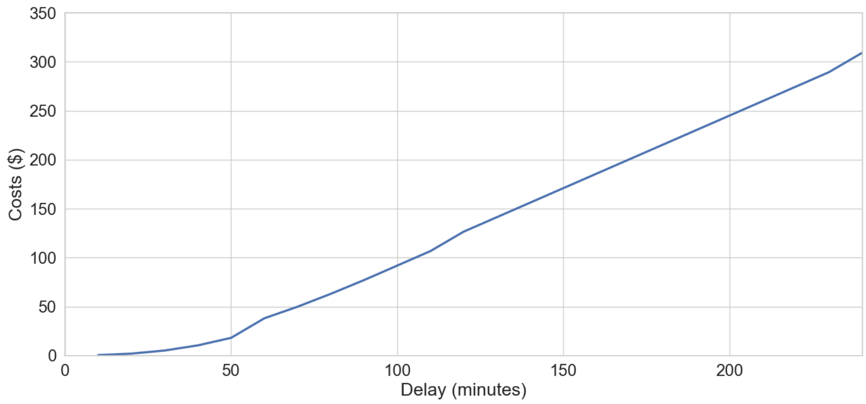

Figure 8: The delay costs per passenger based on the work of Cook et al. (2012).

Results

The proposed integrated model (ML IDS) is compared to the globally optimal solutions from the IDS and to the solutions produced by the sequential disruption set solver (SDSS) developed by Hassan (2018) and Nikolajević (2021). The SDSS does not recover the resources simultaneously, but recovers one resource at a time. This reduces the computational complexity, but does not always result in optimal solutions. The SDSS considers the same recovery actions as the IDS and also makes use of the CPM to take into account passenger flows as well.

The results are gathered by providing the models with disruptions experienced on 05/01/2015 (one day of operations, containing 365 disruptions instances) and analysing the recovery decisions taken using different KPI's. All KPI's used are related either to solution time or solution quality and give a nuanced assessment of the performance of the model. Table 7 depicts the aggregated results for all the 365 instances.

	Optimal	ML IDS	SDSS
Solution Time (Avg.)	1092	70	67
<120s (%)	13%	96%	86%
Cancelled Flights (Sum)	15	25	≥42
Cancelled Passengers (Sum)	1721	2825	≥4937
Average Delay (Avg.)	30.5	31.5	32.1
Delayed Passengers (Avg.)	417.9	310.7	239.1
Aircraft Sink Violations (Sum)	1	3	≥12
Crew Sink Violations (Sum)	57	92	187
Infeasibilities (Count)	0	0	16

Table 7: Results to 365 disruption instances.

	Optimal	ML IDS	SDSS
Aircraft Swaps (Sum)	600	351	214
Crew Swaps (Sum)	220	169	371
Crew Deadheads (Sum)	99	38	105
Reserve Crew Used (Sum)	71	56	118

Table 8: Recovery actions to the 365 disruptions instances.

In terms of solution time, the ML IDS is able to retrieve a recovered schedule in under 120 seconds in 96% of the cases when considering less than 60 aircraft and less than 65 crew pairs, whilst this is only true for 13% of the cases when considering the full network. This shows that the machine-learned ranking model is able to realize a significant computational time reduction. Besides, the memory required to solve the disruptions instances drops from approximately 50 GB for the optimal solutions, to 2 GB for the ML IDS. The SDSS (which selects 50% of all the aircraft in the network and more than 100 crew pairs) is less likely to produce solutions in the required timeframe, as it returns a recovered schedule within 120 seconds in 86% of the cases. Figure 9 depicts the solution times of the disruption instances.

A lower computational runtime is only beneficial if the model still provides high-quality solutions. This is investigated by comparing the solution quality KPI's of the ML IDS with the KPI's of the optimal solutions provided by the IDS. In terms of cancelled flights, the IDS had a total of 15 cancellations, whereas the ML IDS produced 25 cancellations. This increases the total number of passengers on a cancelled flight from 1721 for the IDS to 2825 for the ML IDS. In many of the cases where the ML IDS produced cancellations and the IDS did not, a lot of aircraft and crew swaps were necessary to prevent the cancellations. In these cases, the machine learning algorithm in the ML IDS did not select the correct aircraft and crew in the network, hindering the necessary swaps and resulting in avoidable cancellations. The SDSS produced more than 42 cancellations in all the disruption instances. 17 cancellations occurred in the aircraft stage, but because of the crew allocation infeasibilities in the second stage, more than 25 additional cancellations become inevitable. This comes down to more than 4937 passengers on cancelled flights. Although the ML IDS produces more cancellations than the optimal solutions (67% more cancellations), the SDSS performs significantly worse (280% more cancellations).

It is remarkable that the optimal solution (IDS) has the worst statistics in terms of delayed passengers and average delay per delayed passenger. The SDSS, on the other hand, performs the best on these indicators. Most probably, this depends on the cost factors that are given to the various decision variables in the objective function. The integrated models are more concerned with minimizing cancellations and sink node violations, as the costs associated are much higher than delaying passengers. The model does not penalize short delays too much, as the delay costs start relatively low and increase exponentially with longer waiting times. Since the average delay per delayed passenger is limited to 30 minutes in this case study, these decisions are tolerated. However, if airlines would like to have a more customer-friendly recovery model, this can be achieved by increasing the costs associated with the passengers.

Sink node violations are costly, as this means that those resources are located at an airport not aligned with the schedule. As it is easier to find alternative crew pairs than alternative aircraft to operate flights, aircraft sink node violations are the most costly. The ML IDS is slightly more likely to cause aircraft and crew sink node violations compared with the optimal solution. In the case study, 2 more aircraft sink node violations and 61% more crew sink node violations were produced by the ML IDS. The SDSS performs worse, as it produced more than 11 additional aircraft sink node violations and 228% more crew sink node violations. The reason for this is the crew allocation infeasibilities in the second stage, which causes cancellations and leaves aircraft stranded at undesired airports.

The infeasibilities are a drawback of the SDSS. The reason for this is that the schedule is optimized at first, without considering the crew pairs in the network. This may disturb the crew flow in the network, which requires additional recovery actions in the crew stage to make the schedule feasible again. This phenomenon can also be seen in Table 8, which shows that more crew recovery actions are taken by the SDSS in comparison with the (ML) IDS. In some cases these recovery actions are not sufficient, resulting in crew allocation infeasibilities and inoperable flights. Because of the simultaneous nature of the IDS and the fact that crew flows are considered in the optimization of the schedule, the model is able to generate a feasible schedule in more disruption scenarios.

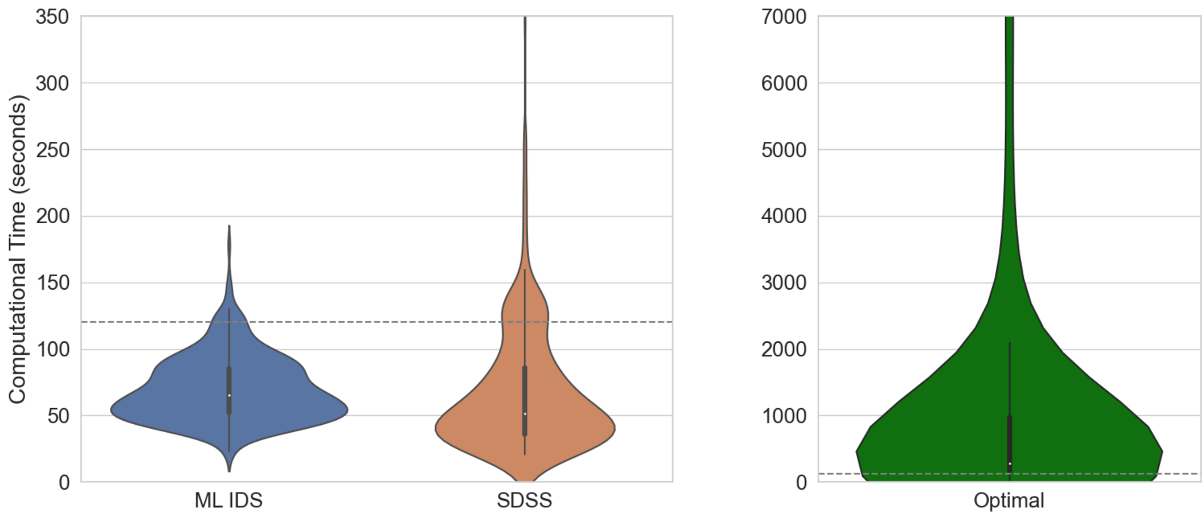

Figure 9: Computational times of the ML IDS, SDSS and optimal solutions, respectively.

Generalization

Table 7 shows the result of solved disruption instances from one day of operations (05/01/2015). However, this is only a fraction of all the disruptions experienced by the airline. Moreover, disruptions may change in duration and severity. Hence, in order to assess the generalization of the machine-learned ranker and to verify the overall performance of the proposed model, disruptions from a whole month of operations are solved. Table 9 depicts the solutions generated by the ML IDS to the disruptions in January 2015, compared to the solutions from disruptions on 05/01/2015.

ML IDS	One Day	One Month
Disruption Instances	365	7416
Solution Time (Avg.)	70	65
<120s (%)	96%	98%
Cancelled Flights (%)	4.4%	4.8%
Average Delay (Avg.)	31.5	36.1
Delayed Passengers (Avg.)	310.7	235.4
Aircraft Sink Node Violations (%)	0.8%	0.4%
Crew Sink Node Violations (%)	13%	14%

Table 9: Results to all disruptions instances in January 2015.

The results to all disruptions in January 2015 are similar to the disruptions on 05/01/2015. The average solution time even dropped and the percentage of instances solved in under two minutes increased from 96% to 98%. The disruption instances containing cancelled flights, aircraft sink node violations and crew sink node violations remained approximately the same. The average delay experienced by passengers slightly increased to 36.1 minutes, but the average number of delayed passengers decreased to 235.4 passengers.

6 Conclusion and Recommendations

The objective of this research was to develop a real-time and integrated airline recovery model that considers the aircraft, crew and passenger recovery. The results show that the ML IDS is efficient both in terms of solution time and solution quality. The subnetwork construction using a machine learning model greatly reduces the computational complexity, while still providing high-quality solutions. The average solution time decreases fifteenfold and 96% of the instances are solved in under 120 seconds, while the model performs similar in terms of delays and only slightly worse in terms of cancellations and sink node violations compared to the optimal solutions. The ML IDS outperformed a sequential model, that recovers the aircraft first after which the crew gets recovered. The sequential model produced more cancellations, sink node violations and returned infeasible solutions in some cases. Furthermore, the ML IDS performs similarly to disruptions experienced during a whole month of operations in comparison to disruptions on one day of operations.

Several limitations and recommendations for future work are provided below. In this research, the training data for the machine learning algorithm was generated by a sequential aircraft and crew recovery model, while the algorithm is implemented in an integrated recovery model. As these approaches differ, the resources involved in the recovery plan may be different as well. This could implicate that a slight bias towards the results of the sequential model will be embedded in the machine learning algorithm of the proposed model. To be more accurate, optimal recovery solutions solved by the integrated recovery model should be used in the future to train the models. Feature engineering could improve the performance of the machine learning models as well. Besides, a careful evaluation and comparison of different machine-learned ranking algorithms and selecting the best-performing in the context of airline recovery could also be useful.

Although passenger missed connections are considered with the use of the connecting passenger matrix, explicit reaccomodation to alternative itineraries is not performed in this model. Furthermore, the recovery of cabin crew is also not considered. Hence, the integrated recovery formulation can still be made more complete. However, limiting the computational complexity will remain a challenge.

References

Ai, Q., Wang, X., Golbandi, N., Bendersky, M., and Najork, M. (2018). Learning groupwise scoring functions using deep neural networks. *CoRR*, abs/1811.04415.

Arikan, U., Gürel, S., and Aktürk, M. S. (2017). Flight network-based approach for integrated airline recovery with cruise speed control. *Transportation Science*, 51:1259–1287.

Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., and Zou, B. (2010). Total delay impact study.

Burges, C., Svore, K., Bennett, P., Pastusiak, A., and Wu, Q. (2011). Learning to rank using an ensemble of lambda-gradient models. *Journal of Machine Learning Research - Proceedings Track*, 14:25–35.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, Microsoft.

Castro, A. and Ana Paula Rocha, E. O. (2014). *A new Approach For Disruption Management In Airline Operations Control*, volume 562. Springer, Berlin Heidelberg.

Chapelle, O. and Chang, Y. (2011). Yahoo! learning to rank challenge overview. In Chapelle, O., Chang, Y., and Liu, T.-Y., editors, *Proceedings of the Learning to Rank Challenge*, volume 14 of *Proceedings of Machine Learning Research*, pages 1–24, Haifa, Israel. PMLR.

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

Cook, A., Tanner, G., and Lawes, A. (2012). The hidden cost of airline unpunctuality. *Journal of Transport Economics and Policy*, 46(2):157–173.

Forbes (2022). The global 2000. https://www.forbes.com/global2000.

Hassan, L. K. (2018). Aircraft disruption management increasing performance with machine learning predictions. Master's thesis, Delft University of Technology.

Huchet, Q., Mattrand, C., Beaurepaire, P., Relun, N., and Gayton, N. (2019). Ak-da: An efficient method for the fatigue assessment of wind turbine structures. *Wind Energy*, 22.

Liu, B., Chen, J., and Wang, X. (2015). Application of learning to rank to protein remote homology detection. *Bioinformatics*, 31(21):3492–3498.

Maher, S. J. (2015). A novel passenger recovery approach for the integrated airline recovery problem. *Computers and Operations Research*, 57:123–137.

Nikolajević, A. (2021). Airline disruption management - a machine learning approach. Master's thesis, Delft University of Technology.

Petersen, J. D., Sölveling, G., Clarke, J. P., Johnson, E. L., and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transportation Science*, 46:482–500.

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

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

Walker, C. (2017). Coda digest q3 2017.

Zhu, B., Clarke, J. P., and Zhu, J. (2016). Real-time integrated flight schedule recovery problem using sampling-based approach. *Journal of Computational and Theoretical Nanoscience*, 13:1458–1467.

Appendices

A Aircraft and Crew Features

AC Feature	Description
Group ID	Group ID necessary for ranking
Iteration	Disruption instance
Disruption_type	The type of disruption
Disruption_cause	The cause of the disruption
Disruption_duration	The duration of the disruption in minutes
d_ac_type	The aircraft type of the disrupted aircraft
d_ac_family	The aircraft family of the disrupted aircraft
d_cap_pax_econ	Economy passengers capacity of the disrupted aircraft
d_cap_pax_buss	Business passengers capacity of the disrupted aircraft
d_pax_econ_fl	Economy passengers on the disrupted flight
d_pax_buss_fl	Business passengers on the disrupted flight
d_pax_econ_max	Maximum No. Economy passengers on the flight string of the disrupted aircraft
d_pax_buss_max	Maximum No. Business passengers on the flight string of the disrupted aircraft
d_range	The range of the disrupted aircraft
d_range_flight	The flight distance of the disrupted flight
d_range_max	Maximum flight distance on the flight string of the disrupted aircraft
d_DOC	Direct Operating Cost per Hour of the disrupted aircraft
d_TAT	Turn Around Time of the disrupted aircraft
c_no_flights	The number of flights scheduled for the candidate aircraft
c_flights_duration	The sum of the flight durations for the flights scheduled for the candidate aircraft
c_ac_type	The aircraft type of the candidate aircraft
c_ac_family	The aircraft family of the candidate aircraft
c_cap_pax_econ	Economy passengers capacity of the candidate aircraft
c_cap_pax_buss	Business passengers capacity of the candidate aircraft
c_econ_lf_mean	The mean economy passenger load factor of all flights scheduled for the candidate aircraft
c_econ_lf_std	The standard deviation of the economy passenger load factor of all flights scheduled for the candidate aircraft
c_buss_lf_mean	The mean business passenger load factor of all flights scheduled for the candidate aircraft
c_buss_lf_std	The standard deviation of the business passenger load factor of all flights scheduled for the candidate aircraft

Table 10: Aircraft Features

AC Feature	Description
c_range	The range of the candidate aircraft
c_DOC	Direct Operating Cost per Hour of the candidate aircraft
c_TAT	Turn Around Time of the candidate aircraft
c_ground_time_d_orig_airport	The total time the candidate aircraft spends on the ground at the origin airport of the disrupted aircraft
d_dest_airport_1_hr_before	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 1 hour before the STA of the disrupted flight
d_dest_airport_2_hr_before	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 2 hour before the STA of the disrupted flight
d_dest_airport_3_hr_before	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 3 hour before the STA of the disrupted flight
d_dest_airport_1_hr_after	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 1 hour after the STA of the disrupted flight
d_dest_airport_2_hr_after	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 2 hour after the STA of the disrupted flight
d_dest_airport_3_hr_after	Indicates if the candidate aircraft is on the destination airport of the disrupted flight 3 hour after the STA of the disrupted flight
d_dest_airport_min_before_min	The minimum number of minutes the candidate aircraft is on the destination airport of the disrupted flight before the STA of the disrupted flight
d_dest_airport_min_after_min	The minimum number of minutes the candidate aircraft is on the destination airport of the disrupted flight after the STA of the disrupted flight
c_ground_time_d_dest_airport	The total time the candidate aircraft spends on the ground at the destination airport of the disrupted aircraft
d_orig_airport_1_hr_before	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 1 hour before the STD of the disrupted flight
d_orig_airport_2_hr_before	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 2 hour before the STD of the disrupted flight
d_orig_airport_3_hr_before	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 3 hour before the STD of the disrupted flight
d_orig_airport_1_hr_after	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 1 hour after the STD of the disrupted flight
d_orig_airport_2_hr_after	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 2 hour after the STD of the disrupted flight
d_orig_airport_3_hr_after	Indicates if the candidate aircraft is on the origin airport of the disrupted flight 3 hour after the STD of the disrupted flight
d_orig_airport_min_before_min	The minimum number of minutes the candidate aircraft is on the origin airport of the disrupted flight before the STD of the disrupted flight
d_orig_airport_min_after_min	The minimum number of minutes the candidate aircraft is on the origin airport of the disrupted flight after the STD of the disrupted flight

Table 10: Aircraft Features

AC Feature	Description
c_DOC_vs_d_DOC	d_DOC - c_DOC
c_TAT_vs_d_TAT	d_TAT - c_TAT
c_range_vs_d_range	c_range - d_range
c_range_vs_d_range_flight	c_range - d_range_flight
c_range_vs_d_range_max	c_range - d_range_max
c_pax_econ_vs_d_cap_econ	c_cap_pax_econ - d_cap_pax_econ
c_pax_buss_vs_d_cap_buss	c_cap_pax_buss - d_cap_pax_buss
c_pax_econ_vs_d_econ_fl	c_cap_pax_econ - d_pax_econ_fl
c_pax_buss_vs_d_buss_fl	c_cap_pax_buss - d_pax_buss_fl
c_pax_econ_vs_d_econ_max	c_cap_pax_econ - d_pax_econ_max
c_pax_buss_vs_d_buss_max	c_cap_pax_buss - d_pax_buss_max
conn_pax_econ_d_to_c	The number of connecting economy passengers from the disrupted aircraft to the candidate aircraft
conn_pax_buss_d_to_c	The number of connecting business passengers from the disrupted aircraft to the candidate aircraft
Result	Indicates if the candidate aircraft was used to recover the disruption

Table 10: Aircraft Features

Crew Feature	Description
Group ID	Group ID necessary for ranking
Iteration	Disruption instance
c_time_start	Duty start time of candidate crew as measured in minutes from time window start time
c_time_end	Duty end time of candidate crew as measured in minutes from time window start time
tw_start	Time window start time as measured in minutes from midnight on day of first departure in time window
tw_end	Time window start time as measured in minutes from midnight on day of first departure in time window
ac_family	Aircraft family, label encoded
type_canx_1*	Represents whether first disruption is a cancellation
type_canx_6	Represents whether sixth or any subsequent disruption is a cancellation
t_del_1*	Represents whether first disruption is a delay
t_del_6	Represents whether sixth or any subsequent disruption is a delay
d_dur_1*	Delay duration in minutes for first disruption
d_dur_6	Maximum delay duration in minutes for sixth or any subsequent disruption
c_at_org_1*	Represents whether candidate is at origin airport of first disrupted flight at time of disruption
c_at_org_6	Represents whether candidate is at origin airport of sixth or any subsequent disrupted flight at time of disruption
c_at_org_1h_1*	Represents whether candidate is at origin airport of first disrupted flight within 1 hour prior to disruption
c_at_org_1h_6	Represents whether candidate is at origin airport of sixth or any subsequent disrupted flight within 1 hour prior to disruption
c_at_org_2h_1*	Represents whether candidate is at origin airport of first disrupted flight within 2 hours prior to disruption
c_at_org_2h_6	Represents whether candidate is at origin airport of sixth or any subsequent disrupted flight within 2 hours prior to disruption
c_at_org_3h_1*	Represents whether candidate is at origin airport of first disrupted flight within 3 hours prior to disruption
c_at_org_3h_6	Represents whether candidate is at origin airport of sixth or any subsequent disrupted flight within 3 hours prior to disruption
c_at_org_before_1*	Represents whether candidate is at origin airport of first disrupted flight at any point prior to disruption
c_at_org_before_6	Represents whether candidate is at origin airport of sixth or any subsequent disrupted flight at any point prior to disruption
c_at_dest_1*	Represents whether candidate is at destination airport of first disrupted flight at time of disruption
c_at_dest_6	Represents whether candidate is at destination airport of sixth or any subsequent disrupted flight at time of disruption

Table 11: Crew Features

Crew Feature	Description
c_at_dest_1h_1*	Represents whether candidate is at destination airport of first disrupted flight within 1 hour prior to disruption
c_at_dest_1h_6	Represents whether candidate is at destination airport of sixth or any subsequent disrupted flight within 1 hour prior to disruption
c_at_dest_2h_1*	Represents whether candidate is at destination airport of first disrupted flight within 2 hours prior to disruption
c_at_dest_2h_6	Represents whether candidate is at destination airport of sixth or any subsequent disrupted flight within 2 hours prior to disruption
c_at_dest_3h_1*	Represents whether candidate is at destination airport of first disrupted flight within 3 hours prior to disruption
c_at_dest_3h_6	Represents whether candidate is at destination airport of sixth or any subsequent disrupted flight within 3 hours prior to disruption
c_at_dest_before_1*	Represents whether candidate is at destination airport of first disrupted flight at any point prior to disruption
c_at_dest_before_6	Represents whether candidate is at destination airport of sixth or any subsequent disrupted flight at any point prior to disruption
c_at_crit_1*	Represents whether candidate is at critical location of first disruption at the critical time
c_at_crit_6	Represents whether candidate is at critical location of sixth or any subsequent disruption at the critical time
c_at_crit_1h_1*	Represents whether candidate is at critical location of first disruption within 1 hour prior to the critical time
c_at_crit_1h_6	Represents whether candidate is at critical location of sixth or any subsequent disruption within 1 hour prior to the critical time
c_at_crit_2h_1	Represents whether candidate is at critical location of first disruption within 2 hours prior to the critical time
c_at_crit_2h_6	Represents whether candidate is at critical location of sixth or any subsequent disruption within 2 hours prior to the critical time
c_at_crit_3h_1	Represents whether candidate is at critical location of first disruption within 3 hours prior to the critical time
c_at_crit_3h_6	Represents whether candidate is at critical location of sixth or any subsequent disruption within 3 hours prior to the critical time
c_at_crit_before_1	Represents whether candidate is at critical location of first disruption at any time prior to the critical time
c_at_crit_before_6	Represents whether candidate is at critical location of sixth or any subsequent disruption at any time prior to the critical time
future_fl_next_1	Represents whether candidate is scheduled to fly to the origin airport of the disrupted crews next scheduled flight after STD of first disrupted flight
future_fl_next_6	Represents whether candidate is scheduled to fly to the origin airport of the disrupted crews next scheduled flight after STD of sixth or any subsequent disrupted flight

Table 11: Crew Features

Crew Feature	Description
future_fl_end_1	Represents whether candidate is scheduled to fly to the end-of-time-window airport of the disrupted crew after STD of first disrupted flight
future_fl_end_6	Represents whether candidate is scheduled to fly to the end-of-time-window airport of the disrupted crew after STD of sixth or any subsequent disrupted flight
crit_time_1	Critical time of first disruption as measured in minutes from time window start time
crit_time_6	Minimum critical time of sixth or any subsequent disruption as measured in minutes from time window start time
reserve_crew	Represents whether candidate is a reserve crew
Result	Represents whether candidate has schedule changed in optimal solution

Table 11: Crew Features

II

Literature Study
previously graded under AE4020

Airline Disruption Management
Literature Study

Berend Eikelenboom, 4538870

Air Transport Operations, Control & Operations, Delft University of Technology

Abstract

Airlines often deal with unexpected circumstances and require efficient recovery strategies to minimize the impact of these disruptions. Various decision-support systems have been developed to help the Airline Operations Control Centers. Preferably, these models should consider all the resources of an airline and should provide solutions within a short period of time, for practical reasons. Different techniques to reach this goal have been proposed. However, it remains a difficult problem, as the number of combinations (and therefore solution time) increases drastically with the increase of the problem size. Machine learning could play an important role, as it has the ability to reduce problem complexities without deteriorating solutions significantly. This literature study reviews the existing decision-support tools, explores how machine learning could be used to simplify the complex structure and aims to discover new promising possibilities in airline disruption management.

1. Introduction

Unexpected events, such as bad weather conditions, mechanical failures or traffic congestion, are unavoidable for airlines and may result in aircraft unavailabilities, delays and airport closure or reduced airflow. Original schedules created for the aircraft and crew, and passenger itineraries may be broken as a consequence. Walker [2017] demonstrated that almost 24% of all flights in Europe were delayed in the third quarter of 2017 as a result of disruptions. Costs associated with these are the result of additional fuel expenses, crew overtime and passenger monetary compensation, which could have a significant impact on the airline business. Ball et al. [2010] estimated that the delay costs in the US airline industry were $32.9 billion of which $8.3 billion were of expenses for additional fuel, crew and maintenance. The Airline Operations Control Center (AOCC) must take appropriate interventions to minimize the severity of the disruption. Currently, most of the airlines have human specialists who monitor the operations of the airline and take action when necessary. It is often too complex to make decisions regarding multiple resources at once, hence the human specialists each focus on one resource (aircraft, crew or passengers) and solve the problem sequentially (Castro and Ana Paula Rocha [2014]). This approach ensures a solution within a short amount of time. However, since resources are interdependent, the sequential approach may not always result in the most optimal solution. A resolved schedule may be optimal for a certain resource, while it might raise conflicts for other resources. Therefore, during the past years research is focused on finding integrated decision-support tools that recover the schedules for the aircraft and crew, and the passenger itineraries altogether. However, modern computers still require too much computational time to solve the available models. This makes it unpractical to implement such systems for airlines in their AOCC, who often require a solution at fleet level within 2 minutes (Vink et al. [2020]). Recent advancements in machine learning technology show great potentials to solve large, complex problems in a shorter period of time. It has the potential to exploit the vast amount of data available in order to make useful predictions that speed up the process. In combination with optimisation problems, it may be a promising direction towards finding a solution to faster, practical models for recovering airline schedules in difficult circumstances.

2. The Airline Recovery Problem

Airline disruption management (ADM) has been a researched field since the 80s, because of its potential to ameliorate the service of airlines. Not only can an efficient disruption management strategy save the airlines a lot of money, also passenger dissatisfaction can be mitigated. Hence, in an increasing competitive environment, a decision-support tool for disruptions can be of real added value.

The first research that has been done on ADM was by Teodorović and Guberinić [1984], who discussed the minimization of passenger delay caused by schedule disruptions. The model allowed aircraft swaps and delaying flights as recovery options. Teodorović and Stojković [1990] extended this work by adding airport curfew as a disruption type and allowing flight cancellations. Later, Teodorović and Stojković [1995] added aircraft maintenance obligations and crew disruptions. During this literature study Google Scholar and Scopus were used as search engines with the following keywords in combination with the terms 'AND' and 'OR' to find relevant articles: airline recovery, aircraft recovery, crew recovery, pairing recovery, passenger recovery, schedule recovery, integrated recovery, disruption management, irregular operations, airline, aircraft. The literature research on the airline recovery problem is largely based on the work of Hassan et al. [2021]. For more details, the reader is referred to their literature review.

The disruption types often considered in recovery models are flight delays (e.g. Aktürk et al. [2014], Arias et al. [2013]), flight cancellations (e.g. Vos et al. [2015], Lee et al. [2020]), aircraft unavailabilities (e.g. Zhu et al. [2015], Sousa et al. [2015]) or crew unavailabilities (e.g. Castro and Oliveira [2009], Chen and Chou [2017]). To react to these, the following recovery actions are frequently taken. For the aircraft recovery, delaying and cancelling flights (e.g. Zhu et al. [2015], Wang et al. [2019]), as well as swapping (e.g. Liang et al. [2018], Zhao and Chen [2018]) and ferrying aircraft (e.g. Zhu et al. [2016], Arikan et al. [2017]) are decisions often taken. For the crew, deadheading (e.g. Petersen et al. [2012], Castro and Ana Paula Rocha [2014]), swapping (e.g. Maher [2015], Zhu et al. [2016]) and using reserve crew (e.g. Le and Wu [2013], Arikan et al. [2017]) are recovery actions that are taken. Lastly, passengers could be re-assigned to another itinerary (e.g. Petersen et al. [2012], Zhu et al. [2016]) if necessary. In the first years, a lot of attention has been placed on solely solving the aircraft recovery problem, while eventually crew and passengers were included as well. Nevertheless, combining all three resources into one model appeared to be a major challenge, both in the industry and in the academia. In order to develop an airline recovery model, network representations are utilized to represent flights and activities during a certain time window.

Briefly, three different types of network representations are used most commonly: time-space, time-band and connection networks. In time-space networks, arcs represent flights and nodes represent activities. Delays of flights are made possible by copying flight arcs and moving them to future time points. Thengvall et al. [2003] extended this method by creating parallel time-space networks to get a more detailed representation, which made it possible to incorporate different fleet types in the model. In time-band networks, nodes represent several activities at an airport during a certain time interval. The network can be created in a dynamic way as disruptions occur. Argüello [1997] proposed this representation and was the first who used it in modelling airline disruptions. Connection network representations use nodes to represent flights legs and arcs for connections between the corresponding flight legs to model the problem. Many solution methodologies are used to solve the model and recover the schedules for the airlines. Among others, exact optimization, (meta-)heuristics, hybrid methods and multi-agents systems are techniques utilized. Argüello [1997] was the first to implement a meta-heuristic and used the greedy randomized adaptive search procedure (GRASP) to solve the airline recovery problem. It is an iterative process in which neighbouring solutions with a local-search procedure are ought to be found from an initial solution. Another novel trend that has been observed in research is the implementation of the dynamic nature of the problem in the modelling framework. Vos et al. [2015] argued that disruptions occur in a dynamic way and are not all known beforehand. Hence, the disruptions should be loaded in the model as soon as they are known by the AOCC. The author concludes that the costs are underestimated by models that utilize an approach in which all disruptions are known at a fixed time before the recovery (static approach).

As mentioned before, it remains difficult to integrate the different resources into a single recovery model. Research has been done on combining aircraft and passengers (e.g. Bisaillon et al. [2010], Jozefowiez et al. [2013]), aircraft and crew (e.g. Le and Wu [2013], Zhang et al. [2015]) and all the three resources together (e.g. Petersen et al. [2012], Zhu et al. [2016]). Generally speaking, two distinct approaches have been implemented in research to accomplish this integration. In the first approach, all resources are combined into one problem formulation and solved with an optimization technique. As all the resources are optimized at once, the solution space increases extremely with larger datasets and therefore the solution time as well. In the second approach, the problem is solved with the use of a sequential modelling framework in which the different resources are modelled in separate stages. In most cases, the schedule recovery and aircraft allocation is done at first in the aircraft stage. After this, the new schedule is fed forward to the crew stage, in which the crew recovery problem is solved. Lastly, the passenger itinerary recovery is performed. Without a solid integration aspect, the schedule in the aircraft stage is recovered optimally solely for the aircraft, without considering crew and passenger itineraries. This often results in local optima and causes conflicts in the crew and passenger stages. In order to integrate this system, the stages are linked with each other by considering the other resources as well when recovering a certain resource or by iterating back to find better solutions. In this literature review the different integration methodologies will be reviewed in order to gain an overview of the state-of-the-art, the difficulties and the promising trends. The datasets and results will be discussed as well to retrieve insights from the different methods used.

2.1. Aircraft and Crew

First of all, the aircraft and crew integration will be discussed. The problem is defined as the recovery of the schedule, and the re-allocation of aircraft and crew members. This section is split up in two, to distinguish the integrated problem formulation from the sequential approach.

2.1.1 Integrated

Le and Wu [2013] used an integrated recovery approach for the aircraft and crew recovery problem (ARP and CRP, respectively). In the objective function, both decision variables for aircraft and crew are included, and the costs exists out of assigning flights to aircraft, assigning crew to flights and the recovery of aircraft (delay and cancellations). The authors mention that the problem is too complex to be solved in real-time by a commercial solver and hence use two approaches to speed up the process. The first one, called node aggregation, combines consecutive arrival nodes and subsequent consecutive departure nodes so that a fewer number of nodes is used in the model. The second one, called island isolation, eliminates unnecessary ground arcs. The model was tested on a Chinese airline with 170 aircraft. However, no solution times were provided.

Maher [2016] used a column and row generation algorithm to solve the aircraft recovery problem and the crew recovery problem. This algorithm extends the column generation algorithm to further reduce the problem complexity and improve runtimes. Next to including more decision variables in the master problem, additional constraints are added as well. This procedure reduces the problem size due to the initial lower number of constraints. In the case study, 441 flights, 123 aircraft, 1 fleet type and 182 crew members were used. Although the column and row generation algorithm was implemented to reduce the complexity, it took the model around 1200 seconds to find solutions.

2.1.2 Sequential

Aguiar et al. [2011] compares different meta-heuristic techniques (hill-climbing, simulated annealing and genetic algorithm) to solve the aircraft and crew recovery separately. The schedule of the ARP serves as an input for the CRP. In the ARP no crew considerations were taken into account and vice versa, which leads to local optima and may cause allocation infeasibilities (and therefore flight cancellations) in the CRP. The authors used 3.521 flights, 51 aircraft. 2 fleet types and 582 crew members in their study. The model was able to determine solutions in 13 seconds.

Zhang et al. [2015] also used a sequential approach to solve the aircraft and crew recovery. The difference with Aguiar et al. [2011] is that crew consideration are taken into account in the ARP by including additional decision variables and aircraft considerations in the CRP by adding extra constraints. The approach that was taken is as follows. First, the schedule is recovered in the ARP without disrupting crew too much. Then, in the CRP the result is rescheduled to decrease costs associated with the crew members. The additional constraints were added in the CRP such that aircraft rotations are kept fixed. The authors used 351 flights, 70 aircraft, 1 fleet type and 134 crew members in their case study. The model obtained solutions in less than 72 seconds.

2.2. Aircraft and Passengers

In this section, the aircraft and passenger recovery problem will be discussed. Like before, the section is split up in works using an integrated problem formulation approach and works using a sequential approach. The problem is defined as the recovery of the schedule, the re-allocation of aircraft to the flights and the re-accommodation of the passengers to itineraries. A lot of work has been done on this type of recovery, partly because of a challenge initiated and organised by the French Operational Research Decision Support Society (ROADEF) in 2009. Relevant and interesting works regarding the integration aspect are discussed below.

2.2.1 Integrated

Hu et al. [2015] modelled the aircraft and passenger itinerary recovery problem with an integer programming (IP) formulation. In this model, it is assumed that the passenger itineraries are comprised of a single flight leg. Because of this assumption, missing connections are disregarded and only passengers with cancelled flights are considered, which simplifies the problem. Aircraft and passenger re-accommodation is integrated in the same IP formulation, whilst using a time-band network to represent the problem. A novelty of this work was that a feasibility study has been conducted. The authors tested the model using a dataset of 188 aircraft, 13 different fleet types and 628 flights. The solutions were found in 172 seconds.

Arıkan et al. [2016] added cruise speed control as a recovery action. By doing so, the model changed to a non-linear type, since the fuel consumption and the aircraft velocity are not linearly dependent. The authors reformulated their approach to a conic quadratic mixed-integer model. The goal of the model was to minimize the passenger related costs and fuel costs by employing recovery actions. The authors tested their approach on a dataset containing 6 fleet types and 1249 passengers, and the model was able to solve it in under 142 seconds.

Vink et al. [2020] modelled the problem as a MILP with a heuristic aircraft selection procedure. The author mentioned that disruptions have a dynamic nature, which means that all the disruptions are most of the time not known up front. Hence, the author included this dynamic nature in its model by allowing the user to input the disruptions as they become known. Passengers were not explicitly re-accommodated, however, a so-called connecting passenger matrix (CPM) was constructed to consider the connection of the passenger itineraries. Additional costs were incurred if passengers would miss their connection in the new schedule. As the problem was still too complex, the author added a heuristic aircraft selection procedure to only consider aircraft that are likely to be chosen in the solution. The model was tested using a dataset with 100 aircraft, 2 fleet types and 600 flights. Solutions were obtained within 44 seconds. The author concluded that static models underestimate the costs and that dynamic models are able to more accurately identify recovery actions and their associated costs.

2.2.2 Sequential

In 2009, ROADEF was initiated to stimulate the development of decision-support tools regarding the simultaneous aircraft and passenger recovery. Bisaillon et al. [2010] won this challenge by creating a large neighbourhood search heuristic. In their model, three stages were considered. In the first one, a new schedule was constructed and was repaired in the second stage based on the constraints imposed. In the third stage, these new schedules were improved by conducting a large neighbourhood search. The ROADEF dataset used consisted of 256 aircraft, 1 fleet type and 1423 passengers. The authors were able to obtain high-quality solutions within 10 minutes.

Mansi et al. [2012] won the second prize in the ROADEF challenge, by approaching the problem with a 2-stage oscillation strategy. In the first stage, the authors attempts to obtain a feasible solution close to the initial schedule by relaxing the model. If no feasible solution could be found, a dynamic programming model would be initiated to find a feasible solution. In the second stage, the model tries to ameliorate the schedule by iteratively destroying and constructing aircraft routes and passenger itineraries. The authors used a dataset with 618 aircraft, 1 fleet type and 2178 flights to test their model. Solutions were found within 10 minutes.

Jozefowiez et al. [2013] also took part in the ROADEF challenge and ended as one of the finalists. The authors formulated a 3-stage heuristic approach. In the first phase, a first feasible solution was obtained by incorporating the disruptions and removing all flight legs and passenger itineraries that were affected. In the next stage, disrupted passengers are re-accommodated in different itineraries. In the last stage, new flight legs were added to potentially re-accommodate the remaining disrupted passengers to new itineraries and hence obtain better solutions. The model was tested on the same dataset Mansi et al. [2012]. The model did not perform as good as the model of the ROADEF winners, however, it could solve the problem instances in less than 4 minutes.

Hu et al. [2016] presented a GRASP meta-heuristic to solve the aircraft and passenger recovery problem. The authors consider both the cancellation of an itinerary and connection time between two flight legs. The procedure consists of three different iterative stages. First, an initial solution, i.e. a schedule with aircraft allocated to the flight legs, has to be generated to start the first iteration. Then, with a local search procedure, neighbouring better solutions are ought to be found. Lastly, passengers are reassigned if their itinerary was broken and had to be recovered. The model is partly integrated, as passenger itineraries are considered in the aircraft recovery stage. The local search procedure consists of three operations that can be carried out, namely inserting, crossing and deleting flights. During the reassignment of passengers, the model considers all feasible itineraries and labels them with seat capacity and delay cost. Next, a minimal cost flow problem is solved to reassign the passengers. The authors tested their model on a dataset with 87 aircraft, 3 fleet types and 340 flights. The model was able to find a solution in less than 100 seconds.

Zhang et al. [2016] used a three-stage math-heuristic to solve the aircraft and passenger itinerary recovery problem. In the first stage the aircraft recovery is solved first, in which the flight schedule and aircraft rotations are recovered without considering passengers with the objective of minimizing delays and cancellations. In the second stage, the flights are rescheduled based on the passenger itineraries, whilst keeping the aircraft rotations from the first stage fixed. In the last stage, the passengers are re-accommodated in the new schedule. The second and third stage are solved in an iterative manner. Costs for the passengers were added in case of a delay, a connection miss for passengers booked on a subsequent flight and a downgrade to a lower class. The model was tested on a dataset provided by ROADEF including 618 aircraft, 1 fleet type and 2178 flights. Solutions were found in 420 seconds. The authors concluded that their model performed the best amongst all benchmarked algorithms tested on the ROADEF dataset.

Yetimoğlu and Aktürk [2021] recently proposed a math-heuristic algorithm for the problem. The authors included cruise speed control as an additional recovery action, next to swapping and cancelling aircraft. The model considers itineraries with a maximum of 2 flight legs, which represents airlines with a hub & spoke network. The algorithm is iterative and consists of three stages. In the first stage, the flights are re-scheduled and aircraft are re-allocated, without considering passengers. In the second stage, passenger itineraries are recovered and passengers' contribution to the objective function is taken into account (e.g. first-class passengers are prioritized). Lastly, cancelled and must-operate flights are fixed and the procedure starts again. The authors tested the model with a dataset consisting of 53 aircraft, 6 fleet types and 208 flights. Solutions were found within 60 seconds.

2.3. Aircraft, Crew and Passengers

Combining all the three resources into one model is the most challenging, as the complexity and solution space increase drastically with increasing dataset sizes. In this section, these complete integrated works will be discussed. Next to the integrated problem formulation and the sequential approach, some have used multi-agent systems to model the problem.

2.3.1 Integrated

Maher [2015] used an integrated problem formulation to solve the complete airline disruption problem and implemented a column and row generation algorithm. The author stated that a Benders decomposition algorithm, an optimization technique often used in large airline-related problems, does not guarantee integer optimality, whilst the column and row generation model does. As stated before, the algorithm extends column generation by also adding additional constraints. Due to a smaller problem size at initiation, the algorithm reduces the computational complexity and improves runtimes. Still, with a dataset of 262 flights, 48 aircraft, 1 fleet type, 30.428 passengers and 85 crew members, the model required around 45 minutes to obtain a solution.

Arikan et al. [2017] introduced the option of changing the cruise speed in the model. As the cruise speed does not have a linear relationship with the fuel consumption and therefore not with costs, the author modelled the problem as a conic quadratic mixed-integer linear program (MILP). In order to decrease the problem complexity and improve runtimes, different methods were implemented. In the first method, partial networks are created for each entity (aircraft, crew and passengers) with the goal of eliminating infeasible recovery actions. The second one being entity aggregation, in which entities with the same partial network are aggregated into one network. The author concluded that the problem size was still too large with the first two methods and implemented a third method in which recovery actions that are not likely to be used were eliminated as well. However, even with these preprocessing methods, the model had to run for around 20 minutes with 1.254 flights, 402 aircraft, 150.118 passengers and 634 crew members.

2.3.2 Sequential

Petersen et al. [2012] proposed a sequential integrated model which was highly related to the model of Lettovsky [1997]. The author used a Benders decomposition optimization technique to solve the problem. In such an approach, the variables are split up in different stages, i.e. in a master problem and in sub-stages. The master problem is optimized first, after which the solution to this problem is fed forward to the following stages. However, if these next stages determine infeasibilities, Benders cuts (additional constraints to tighten the solution space) are generated and the master problem has to be solved again. The author set the schedule recovery model (SRM) as the master problem, whose solution was passed on to the itinerary recovery model (IRM). In this second stage, the problem was solved again and the feasibility was checked, as well as the optimality for the passenger recovery model (PRM). If both of these were not guaranteed, Benders cuts would be added, else the solution was given to the aircraft recovery model (ARM). Again, in this stage the new problem was solved, the feasibility was checked and Benders cuts would be added if an infeasibility was determined. Otherwise, the solution was fed forward to the crew recovery model (CRM). Once more, the problem was solved, the feasibility was checked and Benders cuts would be added if the problem was not feasible. Lastly, the PRM obtained the solution and solved the final problem. In the IRM, all the feasible passenger itineraries that minimize the aggregated delay costs were yielded, and in the PRM the itineraries were assigned to the passenger groups. The authors worked with these distinct passenger stages (IRM and PRM) to reduce computational complexity. The runtime of this model was 30 minutes for a case of 800 flights and 2 fleet types. The number of aircraft, passengers and crew members were not given.

Zhu et al. [2016] opted for a sequential approach to solve the integrated airline disruption management problem. The model uses a sampling-based construction and evaluation approach. In the first stage, a multi-period integer programming (IP) model is used to reconstruct the flight schedule and allocate the aircraft. All feasible solutions are stored. In the second stage, crew schedules (pairings and rosters) and passenger itinerary recovery models evaluate the new schedules of the first stage. Lower bound and upper bound estimations are made for all the solutions in the time period and the non-promising solutions are pruned. A full schedule is made by linking the schedules of all the time-periods together, i.e. a path from start to end. However, the number of paths increase exponentially with increasing time-periods, and the authors concluded that the problem would be over-complex. Hence, random reconstruction path samples were generated to not consider them all. Although the authors mention that the model is able to solve problems in "real-time", it is stated that the computational time is 180 seconds per time stage (5 minutes). Hence, whenever the recovery period becomes longer, larger solution times should be expected. The authors tested the model with 250 flights, 65 aircraft, 6 fleet types, 30.428 passengers and 85 crew members.

Hassan [2018] modelled the aircraft recovery problem with the use of machine learning classifier (random forest) to include the most-promising aircraft in the solution space. Nikolajević [2021] extended this work, by adding the crew recovery problem. The CRP also used a machine learning classifier (extreme gradient boosting) to determine the most-promising crew members and add these to the solution space. The schedule determined by the ARP is used as an input in the CRP. In the ARP, some considerations of passengers and crew are taken into account to integrate the model. With the use of a so-called connection passenger matrix (CPM), the connection time of passengers and crew are examined and a penalty is given if these connections are violated. However, even with the CPM it still occurs that crew allocation is infeasible and flights have to be cancelled. Since the model is not iterative, these cancelled flights will not be recovered. All of this can result in huge expenses that could have been mitigated with a more integrated model. The authors also do not consider passenger re-accommodation. An advantage is that the model could be used as an operational tool, as it was able to find solutions in under 2 minutes for most of the cases. This was partly because the network could be greatly reduced by the machine-learned classifiers, which limits the number of combinations during optimisation. A study of Delta Airlines was used with 327 flights, 8 fleet types and 138 aircraft on average.

2.4. Discussion

Most airlines have an AOCC consisting of human specialists who monitor the operations of the airline and take action in the case of a disruption. Each specialist often focuses on a single resource, as it is too complex to consider all of them together. Then, the solutions are combined and the problem is solved sequentially (Castro and Ana Paula Rocha [2014]). The initial research on automating disruption recovery decision-making also focused on solely one resource at a time. But, advancements in computing power and modelling techniques have enabled new possibilities.

Research is recently focused on combining all the resources in a model and increasing the levels of detail to create a more complete and accurate airline recovery model. However, the added complexity makes it difficult to apply these models in an operational environment, as AOCC controllers require solutions within 2 minutes (Vink et al. [2020]). Combining aircraft and passengers in one model is still doable in terms of problem complexity, as many works described before can obtain solutions in under 3 minutes (e.g. Arikan et al. [2017], Hu et al. [2016] and Vink et al. [2020]). The problem gets more complicated when adding crew, who are highly regulated and restricted by the government, union and airlines itself. The models with an integrated problem formulation, i.e. with decision variables regarding all resources in the objective function, require solution times of 20 minutes or more (e.g. Arikan et al. [2017], Maher [2015]). These models provide a global optimal solution to the problem, but the runtimes are too large and unfit for operational use. The sequential approaches are faster, but often fail at capturing the interdependencies between the resources in the different stages. An overview of the recovery models considering aircraft, passengers and crew are depicted in table 1. There is still a need for an integrated airline recovery model that is able to recover all the resources efficiently and in real-time, usable for operational implementation.

3. Machine Learning

Machine learning (ML), a subset of artificial intelligence (AI), is a technique that leverages data to predict or make decisions without being explicitly programmed to do so. As data is becoming more available nowadays, machine learning is gaining popularity and is used for many complex problems that cannot easily be solved by human decision-makers. Gen-

Table 1: Summary of recovery models considering aircraft, crew and passengers.

Paper	Appr.	Disruption Types				Features		Data	Dimensions			CPU
		Delay	Canx.	AC U/A	Airport	Maint.	MF		AC	Fleet	Flights	(sec)
Petersen et al. [2012]	S	N	N	N	FR & C	Y	Y	RL	-	2	800	1080
Maher [2015]	I	N	N	N	C	Y	Y	G	48	1	262	<2700
Zhu et al. [2016]	S	N	N	N	FR	N	Y	RL	65	6	250	-
Arikan et al. [2017]	I	Y	Y	Y	C	Y	Y	RL	402	-	1254	<1212
Nikolajević [2021]	S	Y	Y	Y	C	N	Y	RL	138	8	327	<120

Abbreviations used in table. Y: Yes, N: No, S: Sequential, I: Integrated, Appr: Approach, Canx: Cancellation, AC: Aircraft, U/A: Unavailable, FR: Flow Restriction, C: Closure, Maint: Maintenance, MF: Multi-Fleet, RL: Real Life, G: Generated, CPU: Central Processing Unit

erally speaking, three types of machine learning exist. The first one is supervised machine learning, which gets trained by using labelled data and learns the mapping from input to output. This trained model can then be used for test data, i.e. data that is not labelled, to estimate or predict its output. Most commonly, this type of machine learning is used for regression and for classification. The second form is unsupervised machine learning. This differs in the sense that it does not learn from labelled data, but instead tries to find patterns and useful information in large, complex data sets that are difficult to analyse. Unsupervised machine learning is often used for data clustering, feature extraction and detecting anomalies. Lastly, reinforcement learning is a form that uses the concept of an agent interacting in its environment, rewarding its desired behaviour and punishing its undesired behaviour, such that it learns a right policy during a training period. This type of machine learning is often useful when one does not have a lot of training data, the ideal end state cannot be clearly defined or when the only way to learn about the environment is to interact with it.

In this chapter, the potential of machine learning in the context of airline disruption management is investigated. First of all, research on airline disruption management that have used AI in their approach are discussed. After this, an overview of the state-of-the-art of machine learning for combinatorial optimization will be provided. Lastly, as the research of Hassan [2018] and Nikolajević [2021] on using machine-learned classifiers showed interesting results, the last sections will be devoted to the investigation of machine learning classification, collective classification and learning to rank. Regression techniques are not considered in this literature study, since generating a sub-network involves labelling (i.e. should this resource be included or not) and not a continuous quantity (which is the output of a regression model).

3.1. AI for the Airline Recovery Problem

Castro and Ana Paula Rocha [2014] used a multi-agent approach to manage the integrated airline disruption problem. The agents monitor the operations of the airline and decide whether the AOCC should take recovery actions. The actions taken by the AOCC are used as input to the system, such that it can learn from the information given. The authors used a dataset of 7.931 flights and 3.028 crew members to test their approach. According to the authors, the integrated approach provides balanced solutions that considers the resources altogether.

Ogunsina et al. [2019] also used a multi-agent approach with an automated learning approach. The authors use a multi-dimensional Markov chain to predict the effect of disruptions on the scheduled airline operations. Two dimensionality reduction techniques for the uncertainty Markov model were discussed afterwards. These techniques are required for the data-driven agent-based approach that uses the uncertainty Markov model to form effective recovery action recommendations to a human controller. The authors did not discuss a case study performed on data of an airline.

Ernst et al. [2020] provided an initial framework on a decision support system for airline operation control hub centre (DiSpAtCH). In this framework, the AOCC gets supported by three machine learning systems. The first one is a machine learning system that uses past operational data to identify disruptions before they occur and their impact on cost and time. The second ML system proposes the 3 most efficient recovery actions that should be taken, based on past recovery actions and their recovery strategies. Lastly, a third ML system tries to accurately predict the impact on cost and time, based on the chosen recovery strategy, the disruption causes, current real prices, and past indications of cost and time impact. The AOCC remains the full control over the decision-making process.

Hondet et al. [2018] implemented a reinforcement learning model for the aircraft recovery problem with aircraft swapping being the single recovery option. A trained agent makes the decision whether to swap certain aircraft with each other or not. The algorithm is eventually compared to the idle strategy, i.e. not swapping and doing nothing. It turned out that the reinforcement learning algorithm was far from optimal and only produced a marginally better solution than the idle strategy. However, the agents takes relevant decisions and the authors mention that the reinforcement learning model might have potential when the agent is trained more or when allowing other recovery options as well.

As discussed before, Hassan [2018] and Nikolajević [2021] used machine learning classifiers in the pre-processing stage to speed up the solution time by making a selection of the resources with the goal of reducing the number of combinations a computer has to evaluate. In their experiments, they showed that in most cases less than 5% of all the aircraft and crew members have a changed schedule in the proposed recovery solution. Hence, not taking all the resources into account in the optimisation model could significantly increase the computational speed. In the ARP, a trained random forest algorithm was used to select a subset of aircraft whose schedule were likely to change. In the CRP, a trained XGBoost (a type of boosted decision trees) was used to classify a subset of crew whose schedule were likely to change. The authors used the Delta Airlines network to test their model. Although the idea was promising, the classifiers were only effective when approximately 50% of all the aircraft and crew members were selected (i.e. around 50% of the resources had to be selected to make sure that the correct 5% of the resources were included as well). For their purpose this was sufficient, as most of the disruption instances were still solved in under two minutes due to their sequential airline recovery approach. However, it is expected that an integrated airline recovery model will require a lot more time using the same ±50% proportion of the resources. A machine learning algorithm should correctly select considerably less resources to make the integrated model fast enough for operational implementation. The authors have not done an extensive analysis on machine learning classifiers or other machine learning algorithms that could be promising for this application.

3.2. Machine Learning for Combinatorial Optimization

According to Bengio et al. [2020], ML can be combined with CO in three different ways. In the pre-processing stage, ML can be used to evaluate the effectiveness of an optimization technique for a problem instance, such as utilizing a certain decomposition scheme to solve the problem. Alternatively, ML can be used to predict the solution time of an optimization problem. Secondly, ML can be used during optimization to help the solver with promising branching decisions during the branch and bound algorithm for a MILP instance. Lastly, ML can be used as an end-to-end method, without the use of a CO solver to optimize a certain problem. Though this has the potential of providing solutions in orders of magnitude faster than traditional solvers, the algorithm does not provide guarantees on optimality or feasibility at all. Most commonly, machine learning is used next to CO to gain information about (how to tackle) a problem instance or to speed up the optimization solution time.

Mazyavkina et al. [2020] have done a survey on the combination of reinforcement learning (RL) with combinatorial optimization. Solving a CO problem with RL requires the formulation of the Markov Decision Process, i.e. the environment, the states & rewards, and the agent itself. The environment is defined by the CO problem instance. The states of the environment should be encoded such that the agents can use it to find a good policy, which can be done by a neural network for example. Lastly, the actions of the agent are defined by the RL algorithm, whose goal is to maximize the rewards. The authors have concluded that RL could be helpful for combinatorial optimization problems in different manners. A trained RL agent can be used to take direct actions, for example building the shortest path from a set of vertices. It can also be used to help the solver internally, by selecting promising nodes in the branch and bound algorithm for example. Another way to categorize RL approaches for CO is by methods that learn the construction heuristics to build a solution incrementally and by methods starting from some arbitrary solution that try to improve the initial solution.

Furthermore, Kenworthy et al. [2021] used reinforcement learning to optimize the assignment of pilots to flight crew schedules with the goal of reducing the impact of disruptions. NICE, abbreviation of Neural Network IP Coefficient Extraction, is the name the authors gave to a model that uses a reinforcement learning algorithm to guide the integer programming (IP) formulation by assigning coefficients weights to the decision variables. The coefficient weights get extracted by a reinforcement learning agent that assigns pilots to slots with a certain probability which are shown in the output layer of the deep neural network. This way, the IP model gets steered in the right direction and has to calculate far fewer combinations to find the optimal solution to this new formulation. They mention that this model could have great advantages over ordinary CO solvers in the case one has to solve complex optimisation problems in a limited solution time for which approximate solutions are sufficient.

3.3. Classification

Classification is a problem that can be tackled by supervised machine-learned classifiers that learn from labelled training data. The data consists of features (the information representing it) and is assigned to a class. During training, classifiers develop a self-built algorithm that can be used to predict the classes of new, unlabelled data. Traditional classifiers have a local nature, i.e. they only consider the features of the data point that is classified and not information about other data points. Olson et al. [2017] benchmarked 13 state-of-the-art machine-learned models in a study consisting of 165 publicly available classification problems. The results showed that decision trees, gradient tree boosting, random forest, support vector machine and neural network (stochastic gradient descent) models yield the best performance. Hence, these models will be discussed in this section. For a more extensive and detailed review of machine learning classifiers, the reader is referred to Akinsola [2017].

3.3.1 Decision Tree

The decision tree algorithm is a popular method for supervised classification. It forms an interpretable tree that splits data-feature values into branches and decision nodes. At each node, a decision is made after which the data gets fed forward to the associated branch. The model can deal with both linear and non-linear data. Trees work well with deterministic data and are generally several orders of magnitude faster than support vector machines and neural networks, according to Kotsiantis [2007].

3.3.2 Random Forest

Random forest form an extension of ordinary decision trees by generating multiple decision trees and taking the majority vote of them to predict the output. This is done by creating bootstrapped datasets, i.e. modifying the dataset by only allowing a subset of data points and features, and making new different trees each time. The goal of using multiple decision trees is to lower the risk of overfitting. However, this technique makes the tree less interpretable. Like ordinary decision trees, random forest can handle non-linear data. Hassan [2018] implemented a random forest classifier in his approach to make a selection of the aircraft in the network likely to be involved in the disruption recovery. The author showed that the model was able to include the relevant aircraft with a probability of 95% if 50% of all the aircraft would be removed.

3.3.3 Gradient-boosting Tree

Gradient-boosting trees also form an extension of decision trees. Decision trees are generated sequentially, where each tree focuses on correcting the errors coming from the previous model. The final result is a combination of the results of all trees. Inherent to ordinary decision trees, the model is able to handle non-linear data points. An example of a gradient-boosting tree is the XGBoost algorithm, a regularized form of a boosted tree which controls over-fitting to ameliorate its performance. Nikolajević [2021] used XGBoost to select a subset of crew during the recovery. The author uses a classifier that is able to correctly identify relevant crew in 99& of the cases, however, this is associated to a selection that contains 50-100% of all the crew in the network (depending on the problem size).

3.3.4 Support Vector Machine

A support vector machine (SVM) is a type of machine learning classification model that draws an optimal linear division between classes. By using a so-called kernel trick, SVM can be generalised to solve non-linear problems as well. In short, kernel functions allow features to be mapped to a higher-dimensional space, such that linear functions are able to classify non-linear data points. According to Jiang et al. [2020], SVM's perform well with unstructured and semi-structured data, and can manage high-dimensional and multicollinear data well, however, they are prone to overfitting. They perform better when the feature data types are continuous, as reported by Kotsiantis [2007].

3.3.5 Neural Network

Neural networks (NN) mimic the behaviour of the human brain and can be trained to handle large amount of complicated and unstructured data. Generally speaking, a NN has three layers, i.e. the input layer, the hidden layer(s) and the output layers. In the input layer the features of the data points get inserted, whilst in the output layer the classification of the data will be shown. The hidden layers try to map the input to the output layers. Neural networks can handle both linear and non-linear data points, depending on the activation functions used and can handle unstructured data very well. NN are known for their accuracy for complex problems, but require a lot of training data. They perform well with multi-dimensional, multi-collinear data and continuous feature types, like SVM's, according to Kotsiantis [2007].

In table 2 an overview of the machine learning models discussed above is depicted. The performance in terms of different criteria are rated with a score from one to five.

Table 2: Performance of machine-learned classifiers on a 1-5 scale. Adapted from Kotsiantis [2007].

	Decision Tree	Random Forest	Boosted Trees	SVM	Neural Networks
Learning Speed	3	2	2	1	1
Classification Speed	4	4	4	4	4
Tolerance to Missing Values	3	2	2	2	1
Tolerance to Redundant Features	2	2	2	3	2
Handling Over-Fitting	2	3	3	2	1
Interpretability	4	2	2	1	1

3.4. Collective Classification

The problem that arises when using a traditional machine learning classifier to select a subset of the resources is that the data is classified independently, without further information about the other data in the problem instance. This is undesired, as the selection of resources is dependent on the other resources in the problem instance as well. A resource with some properties may be a top-candidate in one disruption instance, but may be one of the worst resources in another instance with many other high-potential resources. Ignoring this relational information can confuse a local classifier, as this opens the possibility that a resource with a certain feature vector is selected in one instance, but is not selected in the other. Using a global view to classify an individual data point is called collective classification and may enhance the accuracy of traditional classifiers (Sen et al. [2008], Aggarwal [2014]). Most commonly, the collective classification of nodes depends on its own attributes, the attributes of neighbouring nodes and the labels of neighbouring nodes. Primarily, the process is based on conditional dependence (a label is dependent on the label of another node) or homophily (nodes that share various properties with each other). Several techniques are developed to incorporate this relational information. Sen et al. [2008] provide an overview of three widely used collective classification and inference algorithms.

3.4.1 Iterative Classification Algorithm

The Iterative Classification Algorithm (ICA) seeks to update and revise node attributes based on the labels and features of neighbouring nodes until an equilibrium is reached. If the label of a certain node has to be determined and the labels and/or attributes of all its neighbouring nodes are known, iterative classification work with a local classifier that uses the labels of the neighbouring nodes as input for classification. This method is very flexible, as it uses a local classifier which can be chosen by the user. Hence, anything from decision trees to SVM's would work in such an algorithm. Since it is rare that all labels or attributes of neighbouring nodes are known beforehand, the algorithm iterates the process of classifying nodes using the best estimates of the information of neighbouring nodes. Neville and Jensen [2002] proposed a simple Bayesian classifier, which dynamically updates the attributes of the nodes as inferences are made about related objects. Inferences made with high-confidence in the beginning of the classification process are embedded in the data and used for subsequent inferences to other related nodes. The model has been tested and evaluated on a corporate dataset which consists of data about the intrinsic and relational data of publicly traded corporations. The authors conclude that the iterative approach significantly increase the accuracy of the classification as opposed to traditional machine learning classifiers.

3.4.2 Gibbs Sampling

Gibbs Sampling (GS) is regarded as one of the most accurate inference algorithms, but is very slow in practice. The idea is similar to ICA, as it also estimates the labels of nodes based on labels and attributes of neighbouring nodes. The difference is that GS maintains a count statistic consisting of the number of times label l was given to node Y. After a predefined number of iterations, the algorithm determines the best label for node Y by choosing the label that was assigned the most to node Y. Macskassy and Provost [2007] developed a modular toolkit, called NetKit, for collective classification based on the Gibbs Sampling technique. The authors have tested their work on benchmarked machine learning data sets and conclude that simple collective classification models perform well enough to be used regularly on networked data.

3.4.3 Loopy Belief Propagation

Loopy Belief Propagation (LBP) is a global conditional message-passing classification algorithm in which each node sends a message about its belief of what the message-receiving node should be. The message-receiving node then updates its own beliefs with this additional information. The algorithm differs from other message-passing algorithms in the sense that it discounts the messages with the goal of nodes not receiving their self-generated messages. It is designed to converge on tree graphs, hence does not always converge on non-tree graphs on which it is often used. Sheng et al. [2020] used a loopy belief propagation algorithm to deblur motion in images. The authors show that the deblurring process is convergent and that the number of iterations necessary are acceptable.

More collective classification use cases differ from social network analyses, to document classification and computer vision. Jaafor and Birregah [2017] implement a collective classification algorithm to classify nodes in a social network. In the past, most work that has been done to classify users was done with a local classifier. However, since in practice datasets are unbalanced and some classes have only a few entries, only considering attribute data and neglecting interactions may lead to poor results. The authors propose an iterative classification algorithm for social networks (ICA-SN), which is derived from the traditional ICA. The model aims to detect Jihadi propagandists and malware distributors. The research question of Burford et al. [2015] is how document classifiers can exploit inter-document implicit semantic relationships to improve accuracy. Explicit inter-document links (hyperlinks, name-references and citations) have already been explored, however, documents often do not contain these explicit links. Hence, this research matches pairs of documents based on mutual use of particular n-grams (a contiguous sequence of n items). The authors implemented a dual classifiers approach and iterative classification algorithm. They conclude that the simpler iterative classification algorithm performs better and is suited for such an application.

3.5. Ranking

Collective classification is used to extract additional information about a node from its neighbours. In the context of selecting a subset of resources for the airline disruption recovery, this information is required to determine the relevance of a resource in comparison to the others in the network. Instead of using a collective classification approach, a learning to rank algorithm could also be used to achieve this goal. Learning to rank is a machine learning approach, typically in a supervised setting, which differs from traditional classification and regression in the sense that it does not predict the outcome of one data point, but takes a set of data points (typically referred to as query) and ranks the data points. Hence, if a data point gets a negative predicted score, it means and only means that it is relatively less important than the other data points with a positive score in that group. For a more detailed overview of ranking, the reader is referred to Liu [2011].

Several feature representations can be used to describe the data points in the group.

- Query-dependent: features only depend on data points.
- Query-dependent: features depend on data points and query.
- Query-level: features only depend on query.

Three different ranking methods exist, aiming to rank the data points in the query, as explained by Li [2011], and are discussed below. A fourth, less used method, is developed by Google and is discussed after the three most common ones.

3.5.1 Pointwise Approach

The pointwise approach reduces to a traditional classification problem. All the data points are independently given a score based on their own features, after which the data points are sorted in a list from high to low scores. Existing classification algorithms that output a score or probability can be used, such as boosted decision trees and neural networks. The group structure of ranking is neglected in this approach as the data points are not compared to each other in one way or another (Liu [2009]).

3.5.2 Pairwise Approach

In the pairwise approach, pairs of data points are ranked with the use of a classifier. This is done for all data points in the group, such that a global ranking is achieved at the end of the process. Like the pointwise approach, the overall group structure is still ignored. However, some information regarding the relevance of data points in comparison to others is captured as opposed to the pointwise approach. Some popular models include Ranknet, LambdaRank and LambdaMART developed by Microsoft. As explained by Burges [2010], RankNet was developed first and made use of neural networks. The loss function tries to minimize the number of inversions in ranking, with an inversion meaning an incorrect ordering among pairs. RankNet optimizes the loss function using a stochastic gradient descent. Later, LambdaRank was developed which only uses the gradient (λ, lambda) of the loss, instead of the loss itself (RankNet). These gradients are attached to the data points and indicate the direction where the data point has to go to (more relevant or less relevant). Testing has shown both better and faster results over the original RankNet algorithm. LambdaMART combines LambdaRank and MART (Multiple Additive Regression Trees). The result is a gradient boosted decision tree with a loss function derived from LambdaRank to perform pairwise ranking. During testing, LambdaMART has shown better results than both RankNet and LambdaRank. Furthermore, Ranking SVM (Herbrich et al. [1999]), RankBoost (Freund et al. [2003]), GBRank (Zheng et al. [2008]) and IR SVM (Cao et al. [2006]) are other renowned pairwise methods.

3.5.3 Listwise Approach

The listwise approach does use the overall group structure to rank the data points. The approach directly takes an entire list of data points as an instance and try to come up with the optimal ordering of it. Because of the global view, ranking evaluation metrics can directly be incorporated into the loss function during the training process. Cao et al. [2007] hypothesize that learning to rank should use the listwise approach to increase accuracy. The authors propose two probabilistic methods for a listwise loss function, use it in a neural network with a stochastic gradient descent algorithm and called it ListNet. They conclude that the listwise approach performs better than the pairwise approach. Other renowned models include ListMLE (Xia et al. [2008]), AdaRank (Xu and Li [2007]), SVM MAP (Yue et al. [2007]) and SoftRank (Taylor et al. [2008]).

3.5.4 Groupwise Approach

Pairwise and listwise approaches generally perform well on ranking problems, however, they utilize univariate scoring functions, i.e. the relevance score of a document is computed based on the document itself. Researchers Ai et al. [2018] at Google argue that for some problems a groupwise multivariate scoring function is more appropriate, which takes information of other data points into account when giving relevance scores. The authors evaluate the approach using click logs from one of the largest commercial email search engines, as well as a public benchmark dataset. They conclude that the model leads to better results, especially when the textual features are sparse. Such an approach may be very suitable whenever features of data points do not fully describe the entity and information is missing.

Use cases of learning to rank algorithms differ from information retrieval, recommender systems and machine translations to bio-informatics. Microsoft Research Asia released a benchmark collection for research on learning to rank for information retrieval, as reported by Qin et al. [2010]. Besides, the company has compared several state-of-the-art ranking algorithms in terms of performance: ListNet, AdaRank-MAP, AdaRank-NDCG and SVMMAP. Among those, ListNet produced the best results. Next to information retrieval, learning to rank is also used in other domains. Liu et al. [2015] applied a learning to rank algorithm to protein remote homology detection. This is a problem that requires finding protein sequences in a database that are evolutionarily related to a particular protein. Results on a widely used benchmarked data set show that the learning to rank algorithm outperformed other competing methods.

3.6. Discussion

Machine learning and artificial intelligence are techniques that are being used more often for complex data-driven problems. In airline disruption management, these techniques have also been proposed. Two multi-agent systems with learning capabilities have been developed (Castro and Ana Paula Rocha [2014], Ogunsina et al. [2019]) and a general AI framework for disruption management has been proposed by Ernst et al. [2020]. Besides, Hondet et al. [2018] used reinforcement learning for the aircraft recovery problem, but did not yield good results. Furthermore, machine learning has been proposed to help solve difficult combinatorial optimisation problems. Bengio et al. [2020] explained that ML could be used during pre-processing, in-the-loop and even end-to-end to help solve these issues. Hassan [2018] and Nikolajević [2021] implemented machine-learned classifiers in the pre-processing stage to reduce the computational complexity in their sequential airline recovery model. However, an increase in machine learning performance is necessary to develop a recovery model with an integrated problem formulation that is fit for operational use.

Recently, less traditional machine learning techniques have been proposed to tackle networked data. Collective classification and learning to rank algorithms take into account the global structure of problems and could be of added value. These techniques have already been used in bio-informatics and social networks (Liu et al. [2015] and Jaafor and Birregah [2017]), among others. Table 3 depicts state-of-the-art research using collective classification and learning to rank methods. To the best of the author's knowledge, these techniques have not been used in the airline industry yet. Since networked data is part of many airline related problems, this could be an interesting field to explore. The airline recovery problem is an example of an airline related problem, and could benefit from the use of the machine learning techniques that take into account the global structure.

Table 3: Research using collective classification and learning to rank methods.

Paper	Method	Data	Problem
Neville and Jensen [2002]	ICA	Publicly Traded Corporations	Classification Industry of Company
Qin et al. [2010]	LTR	Documents	Information Retrieval
Burford et al. [2015]	ICA	Inter-Document Links	Document Classification
Liu et al. [2015]	LTR	Evolutionary Proteins	Homology Detection
Jaafor and Birregah [2017]	ICA	Social Network	Detect Jihadi Propagandists and Malware Distributors
Ai et al. [2018]	LTR	Click Logs	E-mail Ranking
Sheng et al. [2020]	LBP	Pictures	Motion Deblurring

Abbreviations used in table. ICA: Iterative Classification Algorithm, LBP: Loopy Belief Propagation, LTR: Learning to Rank

4. Research Goal

Sequential approaches for the airline recovery problem have shown to be time-efficient, but often ignore important interdependencies between aircraft, crew and passengers. This results in solutions that might be optimal for one resource, but unacceptable for the other. Integrated approaches do have a global view and optimise for all the resources in the model at once, but have the drawback that they are computational intractable. Recent advancements in machine learning could be a solution for this, as it has the ability to reduce the complexity and therefore the solution time of large problems. More specifically, learning to rank algorithms show great potential in selecting a small subset of the resources that should be taken into account during optimisation. There is still a need for an integrated recovery model, fit for operational use, and exploring the benefits of an efficient learning to rank algorithm could be of added value to the field of airline disruption management.

At the faculty of Aerospace Engineering in the department Air Transport Operations, several works on airline disruption management have been performed over the last years. Vos et al. [2015] started with developing a dynamic aircraft recovery model. Vink et al. [2020] and Hassan [2018] extended the work of Vos et al. [2015] by using a heuristic and machine learning classifiers respectively to enable faster runtimes. Hoeben [2018] developed a crew recovery model. Recently, Nikolajević [2021] extended the work of Hassan [2018] by adding the crew recovery model sequentially with a machine learning classifier. The lack of accuracy of the machine-learned classifiers, forced the authors to use a sequential approach to obtain fast recovery times. The objective of this research is to develop a better performing machine learning model in combination with an integrated recovery approach to realize a fast and more efficient disruption management model:

Development of a real-time and integrated airline recovery decision-support tool by implementing a learning to rank algorithm that accurately reduces the complexity of the combinatorial optimisation recovery model.

Research questions that indicate what type of knowledge and data is needed to steer the project in the right direction are formulated. The higher-level questions further consist of lower-level questions.

1. What type of machine-learned ranking algorithm is efficient to select the aircraft and crew that should be considered in the integrated airline recovery model?

(a) Is the pairwise, listwise or groupwise approach the most suitable for the problem?

(b) What underlying machine learning model is effective (e.g. neural network or gradient boosted tree)?

2. How can one develop an efficient integrated combinatorial optimisation model?

(a) What is the most computational tractable way to incorporate multiple resources into one problem formulation?

(b) Should the passenger itineraries be explicitly or implicitly modelled in the formulation?

3. How can one further reduce the complexity to obtain real-time solutions?

(a) Which methods can be applied to simplify the problem formulation and speed up the process.

(b) How should the two main performance metrics, solution time and solution quality, be balanced during the model assessment?

5. Conclusions

Literature has been more focused on integrating all the resources of airlines into a single model to cover the complete recovery problem. As the Airline Operations Control Centers require a solution within 2 minutes, the runtime of the decision-support tools are also regarded as one of the key criteria. Sequential approaches have been proposed as they may solve the recovery problem in a faster period of time, but often fail in capturing the interdependencies of the different resources. Models with an integrated problem formulation do guarantee a global optimum, however, their computational complexity increases drastically with larger networks. Selecting a subset of the resources that should be considered in the optimisation model can reduce the computational complexity of the model. Traditional machine learning classifiers have been used in the past, but only solve a prediction problem on a single instance at a time and are therefore not well-suited. The selection of an aircraft or crew member is not only dependent on its own attributes, but also on its relative relevance compared to the other resources available. Hence, this research will focus on implementing a machine-learned ranking algorithm, a method that looks at the relative ordering among all the resources in a disruption instance, to select the resources prone to schedule changes. The objective of the research is the development of a real-time and integrated airline recovery decision-support tool by implementing a learning to rank algorithm that accurately reduces the complexity of the combinatorial optimisation recovery model.

References

C. C. Aggarwal. Data Classification: Algorithms and Applications, chapter Collective Classification of Network Data. Chapman & Hall/CRC, 1st edition, 2014. ISBN 1466586745.

B. Aguiar, J. Torres, and A. J. Castro. Operational problems recovery in airlines - a specialized methodologies approach. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7026 LNAI: 83–97, 2011. ISSN 03029743. doi: 10.1007/978-3-642-24769-9_7.

Q. Ai, X. Wang, N. Golbandi, M. Bendersky, and M. Najork. Learning groupwise scoring functions using deep neural networks. CoRR, abs/1811.04415, 2018. URL `http://arxiv.org/abs/1811.04415`.

J. E. T. Akinsola. Supervised machine learning algorithms: Classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48:128 – 138, 06 2017. doi: 10.14445/22312803/IJCTT-V48P126.

M. S. Aktürk, A. Atamtürk, and S. Gürel. Aircraft rescheduling with cruise speed control. Operations Research, 62(4):829–845, 2014. doi: 10.1287/opre.2014.1279. URL `https://doi.org/10.1287/opre.2014.1279`.

M. Argüello. Framework for Exact Solutions and Heuristics for Approximate Solutions to Airlines' Irregular Operations Control Aircraft Routing Problem. PhD thesis, The University of Texas at Austin, 1997.

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

U. Arikan, S. Gürel, and M. S. Aktürk. Flight network-based approach for integrated airline recovery with cruise speed control. Transportation Science, 51:1259–1287, 11 2017. ISSN 15265447. doi: 10.1287/trsc.2016.0716.

U. Arıkan, S. Gürel, and M. Aktürk. Integrated aircraft and passenger recovery with cruise time controllability. Annals of Operations Research, 236(2):295–317, 2016. URL `https://EconPapers.repec.org/RePEc:spr:annopr:v:236:y:2016:i:2:p:295-317:10.1007/s10479-013-1424-2`.

M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, and B. Zou. Total delay impact study, 10 2010. URL `https://isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf`.

Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon, 2020.

S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin. A large neighbourhood search heuristic for the aircraft and passenger recovery problem. 4OR, 9:139–157, 06 2010. doi: 10.1007/s10288-010-0145-5.

C. Burford, S. Bird, and T. Baldwin. Collective document classification with implicit inter-document semantic relationships. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, pages 106–116, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-1012. URL https://aclanthology.org/S15-1012.

C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, June 2010. URL https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.

Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, page 186–193, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933697. doi: 10.1145/1148170.1148205. URL https://doi.org/10.1145/1148170.1148205.

Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. Technical Report MSR-TR-2007-40, April 2007. URL https://www.microsoft.com/en-us/research/publication/learning-to-rank-from-pairwise-approach-to-listwise-approach/.

A. Castro and E. O. Ana Paula Rocha. A new Approach For Disruption Management In Airline Operations Control, volume 562. Springer, Berlin Heidelberg, 2014.

A. J. M. Castro and E. Oliveira. A multi-agent system for airline operations control. volume 55, pages 159–168, 01 2009. doi: 10.1007/978-3-642-00487-2_17.

C.-H. Chen and J.-H. Chou. Multiobjective optimization of airline crew roster recovery problems under disruption conditions. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(1):133–144, 2017. doi: 10.1109/TSMC.2016.2560130.

J. Ernst, B. Langner, T. Feuerle, and G. Gonzalez. Decision support system for airline operation control hub centre (dispatch), 2020.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res., 4(null):933–969, dec 2003. ISSN 1532-4435.

L. Hassan, B. Santos, and J. Vink. Airline disruption management: A literature review and practical challenges. Computers & Operations Research, 127:105137, 2021. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2020.105137. URL https://www.sciencedirect.com/science/article/pii/S0305054820302549.

L. K. Hassan. Aircraft disruption management increasing performance with machine learning predictions. Master's thesis, Delft University of Technology, 2018. URL http://repository.tudelft.nl/.

R. Herbrich, T. Graepel, and K. Obermayer. Large Margin Rank Boundaries for Ordinal Regression. In Advances in Large Margin Classifiers, chapter 7, pages 115–132. The MIT Press, 1999. URL http://www.herbrich.me/papers/nips98_ordinal.pdf.

N. J. M. Hoeben. Dynamic crew pairing recovery. Master's thesis, Delft University of Technology, 2018. URL http://repository.tudelft.nl/.

G. Hondet, L. Delgado, and G. Gurtner. Airline Disruption Management with Aircraft Swapping and Reinforcement Learning. In SID 2018, 8th SESAR Innovation Days, Salzburg, Austria, Dec. 2018. URL https://hal-enac.archives-ouvertes.fr/hal-01944612.

Y. Hu, B. Xu, J. F. Bard, H. Chi, and M. Gao. Optimization of multi-fleet aircraft routing considering passenger transiting under airline disruption. Computers and Industrial Engineering, 80:132–144, 2015. ISSN 03608352. doi: 10.1016/j.cie.2014.11.026.

Y. Hu, Y. Song, K. Zhao, and B. Xu. 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, 3 2016. ISSN 13665545. doi: 10.1016/j.tre.2016.01.002.

O. Jaafor and B. Birregah. Collective classification in social networks. 07 2017. ISBN 978-1-4503-4993-2. doi: 10.1145/3110025.3110128.

T. Jiang, J. L. Gradus, and A. J. Rosellini. Supervised machine learning: A brief primer. Behavior Therapy, 51(5):675–687, 2020. ISSN 0005-7894. doi: https://doi.org/10.1016/j.beth.2020.05.002. URL https://www.sciencedirect.com/science/article/pii/S0005789420300678.

N. Jozefowiez, C. Mancel, and F. mora camino. A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions. The Journal of the Operational Research Society, 64, 03 2013. doi: 10.1057/jors.2012.20.

L. Kenworthy, S. Nayak, C. Chin, and H. Balakrishnan. Nice: Robust scheduling through reinforcement learning-guided integer programming, 2021.

S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies, page 3–24, NLD, 2007. IOS Press. ISBN 9781586037802.

M. L. Le and C. C. Wu. Solving airlines disruption by considering aircraft and crew recovery simultaneously. Journal of Shanghai Jiaotong University (Science), 18:243–252, 4 2013. ISSN 10071172. doi: 10.1007/s12204-013-1389-y.

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

L. Lettovsky. Airline Operations Recovery: An Optimization Approach. PhD thesis, Georgia Institute of Technology, 1997.

H. Li. A short introduction to learning to rank. IEICE Transactions on Information and Systems, E94.D(10):1854–1862, 2011. doi: 10.1587/transinf.E94.D.1854.

Z. Liang, F. Xiao, X. Qian, L. Zhou, X. Jin, X. Lu, and S. Karichery. A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. Transportation Research Part B Methodological, 113:70–90, 07 2018. doi: 10.1016/j.trb.2018.05.007.

B. Liu, J. Chen, and X. Wang. Application of learning to rank to protein remote homology detection. Bioinformatics, 31(21):3492–3498, 07 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv413. URL https://doi.org/10.1093/bioinformatics/btv413.

T.-Y. Liu. Learning to rank for information retrieval. Found. Trends Inf. Retr., 3(3):225–331, mar 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL https://doi.org/10.1561/1500000016.

T.-Y. Liu. Learning to Rank for Information Retrieval. Springer, 2011. ISBN 978-3-642-14266-6.

S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research, 8(34):935–983, 2007. URL http://jmlr.org/papers/v8/macskassy07a.html.

S. J. Maher. A novel passenger recovery approach for the integrated airline recovery problem. Computers and Operations Research, 57:123–137, 2015. ISSN 03050548. doi: 10.1016/j.cor.2014.11.005.

S. J. Maher. Solving the integrated airline recovery problem using column-and-row generation. Transportation Science, 50(1):216–239, 2016. doi: 10.1287/trsc.2014.0552. URL https://doi.org/10.1287/trsc.2014.0552.

R. Mansi, S. Hanafi, C. Wilbaut, and F. Clautiaux. Disruptions in the airline industry: Math-heuristics for re-assigning aircraft and passengers simultaneously. European J. of Industrial Engineering, 6:690 – 712, 01 2012. doi: 10.1504/EJIE.2012.051074.

N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev. Reinforcement learning for combinatorial optimization: A survey, 2020.

J. Neville and D. Jensen. J. neville and d. jensen (2000). iterative classification in relational data. proceedings of the aaai 2000 workshop learning statistical. 12 2002.

A. Nikolajević. Airline disruption management - a machine learning approach. Master's thesis, Delft University of Technology, 2021. URL http://repository.tudelft.nl/.

K. Ogunsina, N. Davendralingram, I. Bilionis, and D. Delaurentis. Dimensionality reduction in a data-driven model for airline disruption management. AIAA Scitech 2019 Forum, 2019. doi: 10.2514/6.2019-0403. ABM model fro disruption management with leanring aspects.

R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J. H. Moore. Data-driven advice for applying machine learning to bioinformatics problems, 2017. URL https://arxiv.org/abs/1708.05070.

J. D. Petersen, G. Sölveling, J. P. Clarke, E. L. Johnson, and S. Shebalov. An optimization approach to airline integrated recovery. Transportation Science, 46:482–500, 2012. ISSN 15265447. doi: 10.1287/trsc.1120.0414.

T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. Inf. Retr., 13(4):346–374, aug 2010. ISSN 1386-4564. doi: 10.1007/s10791-009-9123-y. URL https://doi.org/10.1007/s10791-009-9123-y.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. AI Magazine, 29(3):93, Sep. 2008. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2157.

B. Sheng, P. Li, X. Fang, P. Tan, and E. Wu. Depth-aware motion deblurring using loopy belief propagation. IEEE Transactions on Circuits and Systems for Video Technology, 30 (4):955–969, 2020. doi: 10.1109/TCSVT.2019.2901629.

H. Sousa, R. Teixeira, H. Lopes Cardoso, and E. Oliveira. Airline disruption management - dynamic aircraft scheduling with ant colony optimization. ICAART 2015 - 7th International Conference on Agents and Artificial Intelligence, Proceedings, 2:398–405, 01 2015. doi: 10.5220/0005205303980405.

M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: Optimizing non-smooth rank metrics. In Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, page 77–86, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939272. doi: 10.1145/1341531.1341544. URL https://doi.org/10.1145/1341531.1341544.

D. Teodorović and S. Guberinić. Optimal dispatching strategy on an airline network after a schedule perturbation. European Journal of Operational Research, 15(2):178–182, 1984. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(84)90207-8. URL https://www.sciencedirect.com/science/article/pii/0377221784902078.

D. Teodorović and G. Stojković. Model for operational daily airline scheduling. Transportation Planning and Technology, 14(4):273–285, 1990. doi: 10.1080/03081069008717431. URL https://doi.org/10.1080/03081069008717431.

D. Teodorović and G. Stojković. Model to reduce airline schedule disturbances. Journal of Transportation Engineering, 121(4):324–331, 1995. doi: 10.1061/(ASCE) 0733-947X(1995)121:4(324). URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-947X%281995%29121%3A4%28324%29.

B. G. Thengvall, J. F. Bard, and G. Yu. A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. Transportation Science, 37(4):392–407, 2003. ISSN 00411655, 15265447. URL http://www.jstor.org/stable/25769165.

J. Vink, B. F. Santos, W. J. Verhagen, I. Medeiros, and R. Filho. Dynamic aircraft recovery problem - an operational decision support framework. Computers and Operations Research, 117, 5 2020. ISSN 03050548. doi: 10.1016/j.cor.2020.104892.

H. W. Vos, B. F. Santos, and T. Omondi. Aircraft schedule recovery problem - a dynamic modeling framework for daily operations. Transportation Research Procedia, 10:931–940, 2015. ISSN 23521465. doi: 10.1016/j.trpro.2015.09.047.

C. Walker. Coda digest q3 2017, 2017. URL https://www.eurocontrol.int/sites/default/files/content/documents/coda-digest-q3-2017.pdf.

D. Wang, Y. Wu, J. Hu, M. Liu, P. Yu, C. Zhang, and Y. Wu. Flight schedule recovery: A simulation-based approach. Asia-Pacific Journal of Operational Research, 36:1940010, 12 2019. doi: 10.1142/S0217595919400104.

F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: Theory and algorithm. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, page 1192–1199, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390306. URL https://doi.org/10.1145/1390156.1390306.

J. Xu and H. Li. Adarank: A boosting algorithm for information retrieval. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, page 391–398, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277809. URL https://doi.org/10.1145/1277741.1277809.

Y. N. Yetimoğlu and M. S. Aktürk. Aircraft and passenger recovery during an aircraft's unexpected unavailability. Journal of Air Transport Management, 91, 3 2021. ISSN 09696997. doi: 10.1016/j.jairtraman.2020.101991.

Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, page 271–278, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277790. URL https://doi.org/10.1145/1277741.1277790.

D. Zhang, H. Y. H. Lau, and C. Yu. A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. Computers and Industrial Engineering, 87:436–453, 6 2015. ISSN 03608352. doi: 10.1016/j.cie.2015.05.033.

D. Zhang, C. Yu, J. Desai, and H. Y. Lau. A math-heuristic algorithm for the integrated air service recovery. Transportation Research Part B: Methodological, 84:211–236, 2 2016. ISSN 01912615. doi: 10.1016/j.trb.2015.11.016.

T. Zhao and X. Chen. A weight-table based heuristic algorithm for aircraft recovery problem. In 2018 37th Chinese Control Conference (CCC), pages 2242–2246, 2018. doi: 10.23919/ChiCC.2018.8484150.

Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems, volume 20. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2007/file/8d317bdcf4aafcfc22149d77babee96d-Paper.pdf.

B. Zhu, J.-f. Zhu, and Q. Gao. A stochastic programming approach on aircraft recovery problem. Mathematical Problems in Engineering, 2015:1–9, 10 2015. doi: 10.1155/2015/680609.

B. Zhu, J. P. Clarke, and J. Zhu. Real-time integrated flight schedule recovery problem using sampling-based approach. Journal of Computational and Theoretical Nanoscience, 13: 1458–1467, 2 2016. ISSN 15461963. doi: 10.1166/jctn.2016.5068.

III

Supporting work

1

Data Input and Pre-Processing

In this section, a more detailed overview of the required data and its pre-processing will be provided. In Figure 1.1 a schematic overview is depicted, which shows what data should be loaded into the model and what pre-processing steps are taken. In the following subsections, the diagram will be explained in detail.

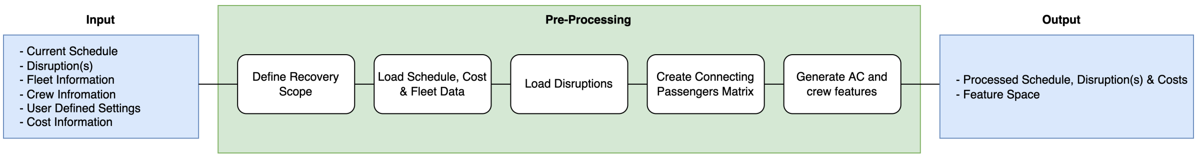

Figure 1.1: Schematic overview of the pre-processing phase.

1.1. Data Input

The airline recovery model requires data regarding the schedule, disruptions, aircraft and crew in order to come up with a recovery strategy. Besides, airline preferences can also be provided to tailor the model to specific needs.

Schedule and Disruption Information

In terms of schedule information, the flight, aircraft and crew schedules are required. That is, each flight should be described by an origin and destination, a scheduled departure time (STD) and scheduled arrival time (STA), and the number of passengers (economy and business) should be given. Moreover, the distances between the airports in the schedule should be provided. Lastly, the connecting passenger itineraries and the minimum required time for passengers transferring on a new flight for each airport are necessary. In terms of disruption(s), two pieces of information are necessary. The first one is the disruption type, which could be a flight delay, a flight cancellation, an aircraft unavailability or an airport closure. Other types of disruptions can not be handled by the proposed model (e.g. reduced flow at an airport). The second piece of information contains the affected flight(s), aircraft, crew (and airport(s) in case of a closure).

Schedule	**Disruption(s)**
Flight schedule	Disruption type
Aircraft schedule	Affected flight(s)
Crew schedule	Affected aircraft
Distances between airports in the schedule	Affected crew
Connecting passenger itineraries	Affected airport(s)
Minimum required passenger connection time	

Table 1.1: Schedule and disruption(s) information.

Fleet and Crew Information

Next to the schedule and disruptions, fleet information is required. All the aircraft tail numbers, types and families in the fleet should be provided. Besides, aircraft characteristics are also necessary. The direct operating cost (DOC), turn-around-time (TAT), range and passenger capacity in both the economy and business class, should be given per aircraft type. Similarly to the fleet information, crew information should also be given. All the crew and reserve crew in the network should be provided, but also the operating costs per crew type, the aircraft families each crew is allowed to operate and the minimum required time for the crew to transfer flights for each airport.

Fleet	**Crew**
Aircraft in fleet	Crew in network
DOC	Crew operating cost
TAT	Aircraft family of crew
Range	Minimum required connection time
Passenger capacity	

Table 1.2: Fleet and crew information.

User Defined Settings

User-defined settings consisting of airline preferences can be given, which tailor the model to the needs of the airline. The length of the time window can be increased, resulting in more recovery options. The time steps may also be increased, which enhances the accuracy of the recovery model. However, these come at the expense of additional computational time. The tail swap time limit is another user-defined setting, which defines a time block before departure in which aircraft may not be swapped anymore. This prohibits aircraft from being swapped minutes before a flight, which is infeasible in reality as ground processes are required to make the aircraft ready for their flight.

Costs

Different cost factors present in the objective function eventually determine what to optimize for. All the cost factors are given in Table 1.3. First of all, local regulations determine the compensation that has to be given to passengers in case of a cancelled flight. The cancellation fee should be set in such a way that it accurately represents these costs. In terms of delay, both hard costs and soft costs are considered in this research. These costs increase with longer delays. The hard costs are defined as the legal compensation that the airline is obliged to return to the customer in case of delays (depending on the duration of the delay). The soft costs, on the other hand, are defined as the expenses not directly related to the repayment, but to the bad experience of the customer due to a delay or cancellation. In the case of a cancellation, the costs equal the soft costs associated with a large delay (can be chosen by the airline), plus the additional cancellation fee.

Cost Factor	**Description**
Cancellation fee	Additional cost per passenger in case of a cancelled flight.
Delay fee	Additional cost per passenger in case of a delay, which varies per delay duration.
Business multiplier	Business passengers have higher associated costs, which equal the economy passenger costs multiplied by the business multiplier.
AC schedule penalty	Penalty assigned when another aircraft than scheduled operates a flight.
Crew schedule penalty	Penalty assigned when another crew than scheduled operates a flight.
AC sink node penalty	Penalty assigned when an aircraft ends at an undesired airport.
Crew sink node penalty	Penalty assigned when crew ends at an undesired airport.
Reserve crew sink node penalty	Penalty assigned when reserve crew ends at an undesired airport.
Crew deadheading	Costs associated to deadheading crew on a flight.
Flight time penalty	Penalty assigned when crew exceeds its maximum flight time.

Table 1.3: Cost factors used in the model.

1.2. Pre-processing

The input data is then pre-processed, such that the information can be interpreted by the computer program. The recovery scope is determined and is defined as the schedule and the known disruption(s) from the start of the time window until the end of the time window. The start of the time window always equals the time at which the first disruption became known. The end of the time window is based on the preference of the airline and is either determined by the given length of the time window or the specific time at which the time window should end. Furthermore, the connecting passenger matrix is constructed, and the aircraft and crew features are generated. The last two steps will be explained in more detail below.

Connecting Passenger Matrix

The connecting passenger matrix (CPM) is based on the research performed by Vink et al. [13], who aimed to include passenger considerations as well without explicitly modelling them. Vink et al. developed a one-sided CPM, which computes the missed connection costs in the case a flight delay causes passengers to miss their connection. The methodology of the CPM will be explained using Figure 1.2. Flight 1 from ATL to LA has a connecting Flight 2 from LA to NY 10 minutes after the arrival of Flight 1. In the case of a 10-minute delay, the passengers are still able to make the connection. However, if the delay is larger than 10 minutes, the passengers will miss their connection. The next flight going to NY from LA is Flight 3, which departs 40 minutes after Flight 2 and is the first possible alternative for the passengers. As a result, if Flight 1 is delayed by more than 10 minutes, the delay experienced at the end destination is equal to 40 minutes. If Flight 1 is delayed by more than 50 minutes, the connection to Flight 3 will also be missed. As this is the last flight of the day, the maximum delay costs will be assigned in this case. All connecting flights are evaluated according to this methodology. The passenger delay at the end destination is calculated for every delay step and the respective delay costs are added to the CPM. The CPM is a matrix of size FxT, with F being the number of connecting flights in the schedule and T the number of delay steps the airline wants to consider.

Hassan [8] further extended the CPM, by allowing the connecting flights to be delayed as well if this does not disturb the downstream flights after it. Again, consider Figure 1.2. If Flight 1 is delayed by 20 minutes, Flight 2 could be delayed by 10 minutes without causing problems. The result is a more optimal solution, as passengers on the 20-minute delayed Flight 1 are able to make their connection to Flight 2, while the connection to Flight 4 can also be made without any problems. Hassan [8] only made this option available for the outbound connecting flights of disrupted inbound flights. The author noted that the computational complexity would increase too much if all subsequent flights would be considered.

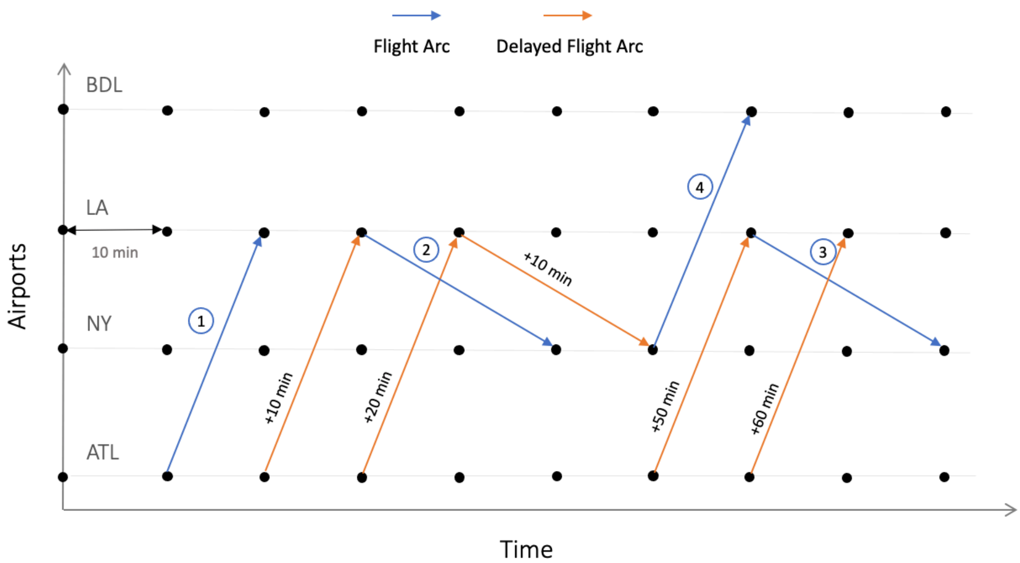

Figure 1.2: Connecting flights example.
Adapted from Hassan [8].

Generation of Features

After pre-processing, the two machine learning algorithms predict which resources are likely to be involved in the optimal recovery plan based on the features describing them. Hence, the last step during pre-processing is to generate these features. The model iterates over all the resources in the network and determines the values of the features, until all resources are considered. The feature values can be categorical (e.g. aircraft type) or numerical (e.g. aircraft range). All features of the resources are determined by either their position in the network (e.g. being at the same airport as the disrupted resources), schedule-related information (e.g. same end-of-duty airport as disrupted crew) or the user-provided information (e.g. turn-around-time at a specific airport). Table 1.4 shows the structure of the feature space. The most right column Y shows the ranks that should be determined by the machine learning algorithms.

	F_1	F_2	F_3	...	F_n	Y
C_1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	...	$v_{1,n}$	y_1
C_2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	...	$v_{2,n}$	y_2
C_3	$v_{3,1}$	$v_{3,2}$	$v_{3,3}$	...	$v_{3,n}$	y_3
...	...	...	...	...	...	...
C_m	$v_{m,1}$	$v_{m,2}$	$v_{m,3}$	...	$v_{m,n}$	$y_{m,n}$

Table 1.4: Structure feature space.

2

Machine-learned Ranking

This section will provide additional information regarding the machine-learned ranking algorithms that are used in the proposed integrated airline recovery model.

2.1. Motivation for Machine-learned Ranking

Learning to rank algorithms aim to retrieve an optimal ordering of items in a group, by training it with labelled data. The labels provide information on the relevance of the data, which could be binary (i.e. 1 is relevant, and 0 is not relevant) or integer to differentiate between distinct levels of relevancy. The items are described by features and belong to a certain group in which the ranking happens. Learning to rank differs from traditional classification and regression in the sense that it does not predict the outcome of one data point, but takes a group and ranks the data points within it. Previous research on airline disruption management has made use of classification algorithms to make the selection of resources, however, these are not effective enough to reduce the computational time for the integrated recovery to under two minutes. A reason for this could be that the context of a disruption instance is not taken into account, whilst this could be realised in the learning to rank algorithms by specifying the groups of resources. Without this context, it is difficult to understand the potential of a resource. For example, an aircraft with certain features may be the most helpful in a particular disruption instance, but that does not guarantee that it also is the most helpful in another instance. The relevance of a resource in comparison to the others in the instance is actually the important information that has to be predicted. Having other more helpful or less helpful resources in the network that could be used in the recovery plan affects the helpfulness of a particular resource. This is why a learning to rank model could be preferred over the traditional machine learning models.

2.2. Feature Manipulation

Some features do not apply to certain resources and are thus left blank. However, those missing values should be carefully treated, as these could have a negative influence on the models. One option could be to delete those resources, but this could remove valuable information from the data. Imputing missing values is in most cases a better option, and could be done in several ways. For numerical data, statistical expressions such as the mean, mode or median over all resources could be used. Another strategy is to assign values outside the range of the feature distribution, expecting that the machine learning model could easily filter these out. The former strategy is chosen and is applied to the missing values in the datasets. Furthermore, another technique was used to handle categorical features, as the open-source libraries for the machine learning models generally do not accept categorical features. The most simple way to deal with this problem is to assign numerical values for each distinct categorical feature, e.g. '737' becomes 1, 'A320' becomes 2 and so on. However, machine learning models could understand this as 2 being greater than 1, whilst nominal categorical features do not have an ordering between the distinct categories. Because of this, another technique known as 'One-hot Encoding' is used, which mitigates this problem by applying binarisation to the categorical features. Table 2.1 shows this principle and, as can be seen, no ordering among the distinct categories is created.

Categorical Feature	One-hot Encoding
A320-200	100
737-700	010
MD-88	001

Table 2.1: One-hot Encoding.

2.3. Evaluation Metrics

Many evaluation metrics are developed nowadays for ranking. Popular ranking metrics include the mean reciprocal rank (MRR), mean average precision (MAP) and the normalized discounted cumulative gain (NDCG). MRR only considers the highest ranked relevant data point in the list and is useful for targeted cases where the lower ranked data points are not of importance.

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \tag{2.1}$$

In Equation 2.1, n is the number of groups and $rank$ is the rank position of the first relevant item in group i. MAP is determined by calculating the precision for increasing recall scores, after which the average is calculated over all groups. The precision is the fraction of relevant items selected and the recall is the fraction of selected items that are relevant.

$$\begin{aligned} \text{MAP} &= \frac{\sum_{i=1}^{n}\text{AveP}(i)}{n} \\ \text{AveP} &= \sum_{k=1}^{n} P(k)\acute{}r(k) \end{aligned} \tag{2.2}$$

NDCG incorporates the actual order of the data points in the ranked list and is able to distinguish the relevance of the different data points in the list (i.e. more relevant data points placed higher in the list are valued greater). The metric penalises relevant items ranked lower in the list. First, the discounted cumulative gain (DCG) is calculated by summing the gain of each item with relevance rel at position i, up until a certain rank position p. Then, the DCG is normalised by dividing it by the ideal discounted cumulative gain (IDCG), which is the DCG for an ideally ranked list.

$$\begin{aligned} \text{DCG}_\text{p} &= \sum_{i-1}^{p}\frac{\text{rel}_i}{\log_2(i+1)} \\ \text{nDCG}_\text{p} &= \frac{DCG_p}{IDCGp} \end{aligned} \tag{2.3}$$

These metrics are relevant for some ranking or information retrieval use cases, however, the metrics do not provide an accurate indication of the performance in the context of this research. This is because the actual order of the relevant resources is not important, so the NDCG is not the most ideal metric. Besides, some disruption instances will contain more than only one helpful resource for the recovery, while MRR only considers the most relevant item. Lastly, precision isn't a well-suited metric for the machine-learned ranker in this context, as explained in part I of this thesis.

2.4. Hyperparameter Optimisation Techniques

Hyperparameters define the construction of the machine learning model and influence its performance. They should be optimised to increase the effectiveness of making a selection before the airline recovery. Since it is computationally expensive to test all the different hyperparameter settings, different techniques are developed to approach this in a structured manner. Grid search is the simplest technique, because it exhaustively searches on a discrete subset of the hyperparameter space. Although simple, the method suffers from the curse of dimensionality, which means that the computational speed greatly reduces with an increasing number of hyperparameters. Random search does not search on a grid, but randomly picks combinations of hyperparameters and eventually chooses the set with the best performance according to an evaluation metric. Bayesian optimisation is the most structured and efficient method to optimise the hyperparameters. The technique applies a probabilistic surrogate model that tries to approximate an objective function, based on known function values (i.e. the evaluations of models with certain hyperparameters). From the probabilistic model, also called a prior, a posterior distribution is created, which interpolates the known function values with the information gained from the prior. The posterior constructs an acquisition function which provides an indication for the next sample to pick. This process is repeated each time a new sample is evaluated, such that more information can be used to predict the objective function.

Bayesian optimisation is useful for finding optima on a continuous function of which the derivative is unknown (gradient descent techniques remain more efficient, but require a derivative) and which is expensive to evaluate at many points. The difference between Bayesian optimisation and random search or grid search is that the first efficiently makes use of the information retrieved during the evaluation of the previous sample points.

2.5. Results

Two LambdaMART models are trained and optimized for both aircraft and crew. The feature importance of both models is depicted below and is based on the usefulness of the feature in predicting the outcome. The description of the features are stated in the appendix in part I of this thesis.

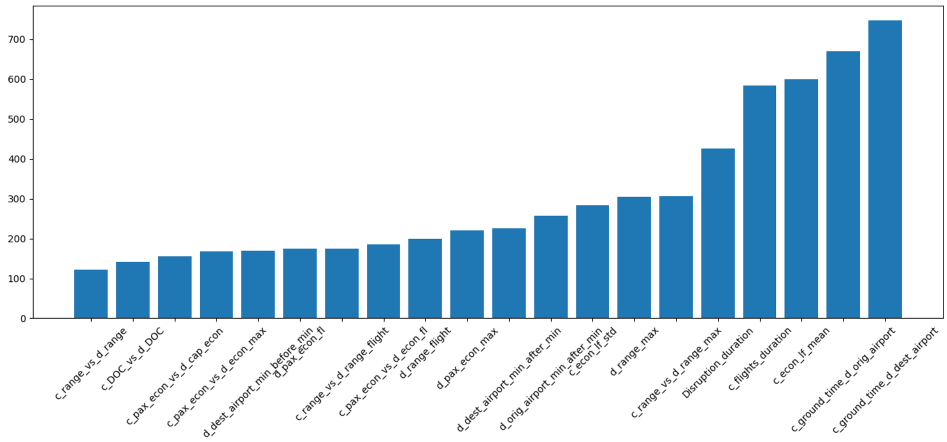

Figure 2.1: Feature importance in the aircraft machine learning algorithm.

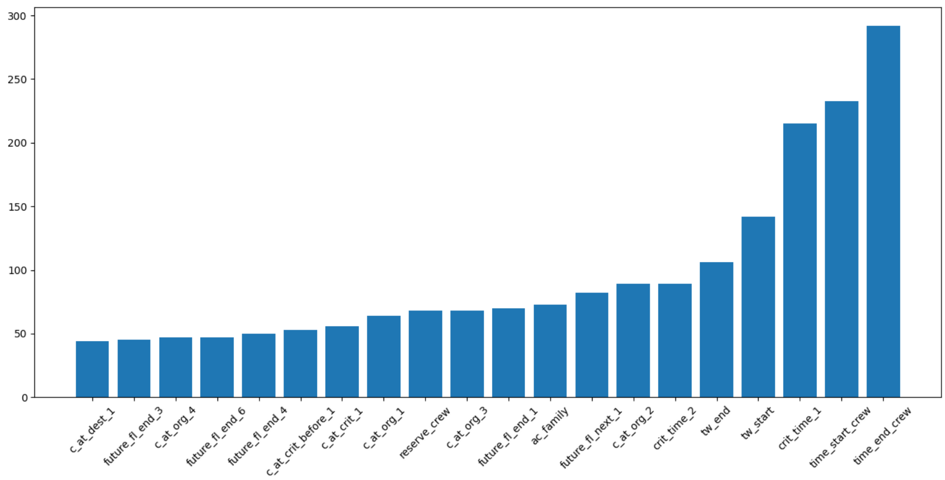

Figure 2.2: Feature importance in the crew machine learning algorithm.

The results of the machine-learned ranking algorithms are compared with the classifiers used in the SDSS, i.e. a random forest classifier for the aircraft selection and XGBoost for the crew selection, and are depicted in Figure 2.3 and Figure 2.4 respectively.

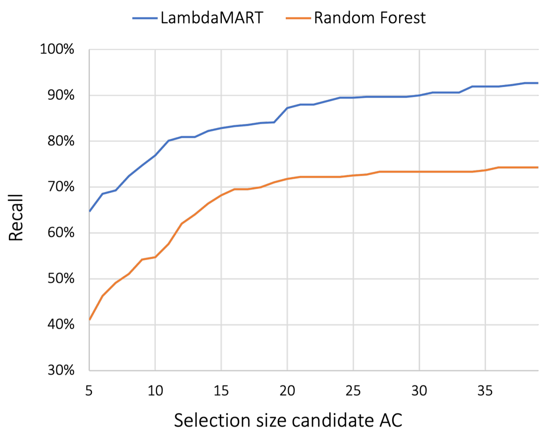

Figure 2.3: Recall comparison with increasing selection sizes for aircraft machine learning models.

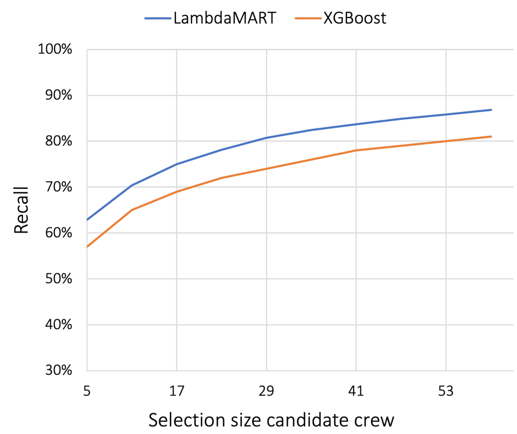

Figure 2.4: Recall comparison with increasing selection sizes for crew machine learning models.

The results show that both the aircraft and crew LambdaMART models outperform the random forest and XGBoost classifiers respectively. In terms of aircraft, the difference is more significant, since a recall increase of 15 – 25% is realized by using a learning to rank algorithm. The crew models perform more similar, but still an increase of 4 – 8% can be seen.

The recall of the aircraft and crew machine-learned rankers, however, does decrease with an increasing number of involved resources in the recovery, as shown in Figure 2.5 and Figure 2.6 respectively. As most disruption instances are recovered by using only a few resources, the problem is not significant. But for more severe disruptions with more resources involved, the models have difficulties identifying the optimal resources.

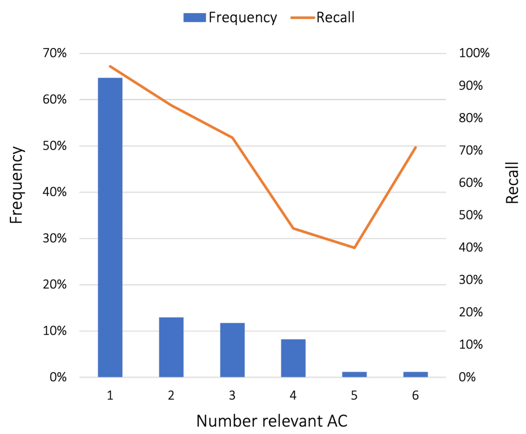

Figure 2.5: Recall for an increasing number of relevant aircraft.

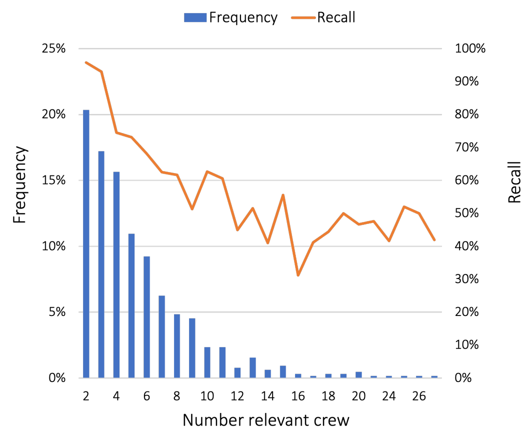

Figure 2.6: Recall for an increasing number of relevant crew.

3

Integrated Recovery Model

This section will provide additional information regarding the mixed-integer linear programming (MILP) formulation used in the integrated airline recovery model. The MILP attempts to find the optimal solution to one or multiple disruptions in a given time window. The model optimises the recovery of the schedule and allocates aircraft and pilot pairs to flights simultaneously. Connecting passengers are implicitly included by the connecting passenger matrix (CPM), which imposes an appropriate cost in the objective if passengers cannot make their next flight. The model is based on a parallel time-space network for every aircraft and crew pair, in which time is discretized on one axis and the airports are located on the second axis. The following sets, indices, parameters and decision variables are used in the optimisation model and will be used throughout this section.

Sets and Indices

Sets		**Indices**	
F	flights	i	flight index
K	crews	k	crew index
A	airports	t	delay time index
E	aircraft types	a	airport index
P	aircraft	p	aircraft index
P(e)	aircraft p of type e	e	aircraft type index
N	nodes = $N_O \cup N_I \cup N_S$	n	node index
N_O	origin nodes	j	artificial variable index
N_I	intermediate nodes		
N_S	sink nodes		
T	delay steps		

Parameters

Aircraft		**Crew**	
$C_{OP_{p,i}}$	Operating cost of AC p on flight i	$C_{OP_{k,i}}$	Operating cost of crew k on flight i
$C_{D_{i,t}}$	Delay cost of flight i for delay t	$C_{DH_{k,i}}$	Deadhead cost of crew k on flight i
C_{C_i}	Cancellation cost for flight i	C_{OC}	Unscheduled crew operating penalty
C_{G_n}	Cost of ground arc from node n	C_{SV_k}	Sink node violation cost for crew k
h_n^e	Number of AC of type e required at node n	C_{FT}	Flight time exceeded penalty
$C_{C_{SCH}}$	Unscheduled AC operating penalty	FT_i	Flight time of flight i
C_{ASV_k}	Sink node violation cost for crew k	FTL_k	Flight time remaining in TW for crew k
		FTM_k	Maximum additional flight time for crew k

Decision Variables

Aircraft

$\delta_{F_{p,i}}$	if flight i is flown by AC p without delay
$\delta_{FD_{p,i,t}}$	if flight i is flown by AC p with delay t
δ_{C_i}	if flight i is cancelled
$\delta_{GP_{p,n}}$	if AC p uses ground arc n
$\delta_{F'_i}$	if flight i is flown by an unscheduled AC
s_j	slack variable for sink constraint violation

Crew

$\delta_{K_{k,i}}$	if crew k is allocated to flight i without delay
$\delta_{KD_{k,i,t}}$	if crew k is allocated to flight i with delay t
$\delta_{GK_{k,n}}$	if crew k uses ground arc n
$\delta_{K'_i}$	if flight i is flown by an unscheduled crew
$\delta_{DH_{k,i}}$	if crew k is deadheaded on flight i without delay
$\delta_{DHD_{k,i,t}}$	if crew k is deadheaded on flight i with delay t
s_k	slack variable for sink constraint violation
s_{FT_k}	slack variable for exceeding scheduled flight time

3.1. Objective Function

$$\begin{aligned}\text{Min} &\sum_{p\in P}\sum_{i\in F} C_{OP_{p,i}}\cdot\delta_{F_{p,i}} + \sum_{p\in P}\sum_{i\in F}\sum_{t\in T}\left(C_{OP_{p,i}} + C_{D_{i,t}}\right)\cdot\delta_{FD_{p,i,t}} + \sum_{i\in F} C_{C_i}\cdot\delta_{C_i} + \sum_{p\in P}\sum_{n\in N} C_{G_n}\cdot\delta_{GP_{p,n}} + \sum_{i\in F} C_{C_{SCH}}\cdot\delta_{F'_i}\\
&+\sum_{k\in K}\sum_{i\in F}\left(C_{OP_{k,i}}\cdot\delta_{K_{k,i}} + C_{DH_{k,i}}\cdot\delta_{DH_{k,i}} + \sum_{t\in T}\left(C_{OP_{k,i}}\cdot\delta_{KD_{k,i,t}} + C_{DH_{k,i}}\cdot\delta_{DHD_{k,i,t}}\right)\right) + \sum_{k\in K}\sum_{n\in N} C_{G_n}\cdot\delta_{GK_{k,n}} + \sum_{i\in F} C_{OC}\cdot\delta_{K'_i}\\
&+\sum_{j\in S} s_j\cdot C_{ASV} + \sum_{k\in K} C_{CSV}\cdot s_k + \sum_{k\in K} C_{FT}\cdot s_{FT_k}\end{aligned} \tag{3.1}$$

The objective function is a minimisation problem and consists of aircraft, crew and passenger related costs. The former consists of the missed connection costs for passengers and is implicitly modeled in the delay cost of a flight through the CPM.

The first line represents all the aircraft related costs, consisting of the direct operating cost (DOC) for the flights operated as scheduled, the DOC and delay cost for the flights operated with a delay, the cancellation cost for the cancelled flights, the cost of operating a ground arc for the aircraft that stay on the ground and the additional cost for operating a flight with an unscheduled aircraft, respectively.

The second line represents all the crew related costs, consisting of the operating and deadheading cost for crew on an on-time flight, the operating, deadheading cost for crew on a flight with a delay, the cost of operating a ground arc and the additional cost for operating a flight with unscheduled crew, respectively.

The last line consists of slack variables to ensure feasibility and prevent unwanted behaviour from the model. The first refers to the aircraft sink node violation, the second to the crew sink node violation and the last one to the crew flight time violation. If the constraints associated to the slack variables cannot be satisfied, the slack variables are activated and a penalty is imposed in the objective function.

3.2. Time-space Network Constraints

The first set of constraints are related to the time-space network. These constraints ensure that all activities in the network occur as expected.

Flight Coverage

Constraints 3.2 ensure that all flights are either flown as scheduled, delayed or cancelled. Without these constraints the model would ground many aircraft, since the costs of staying on the ground are much lower.

$$\delta_{C_i} + \sum_{p\in P}\left(\delta_{F_{p,i}} + \sum_{t\in T}\delta_{FD_{p,i,t}}\right) = 1 \quad \forall i\in F \tag{3.2}$$

Crew Coverage

All flights have to be flown by a crew pair. A differentiation is made between flights flown on time and flights with a delay. Constraints 3.3 ensure that all the on-time flights are allocated a crew pair and Constraints 3.4 ensure that all the delayed flights are operated by a crew pair.

$$\sum_{p\in P}\delta_{F_{p,i}} = \sum_{k\in K}\delta_{K_{k,i}} \quad \forall i\in F \tag{3.3}$$

$$\sum_{p\in P}\delta_{FD_{p,i,t}} = \sum_{k\in K}\delta_{KD_{k,i,t}} \quad \forall i\in F, \forall t\in T \tag{3.4}$$

Crew Deadheading

Deadheading may be of interest in some cases. Crew pairs could either fly with on-time flights or with delayed flights. Constraints 3.5 and 3.6 prohibit deadheading on cancelled flights and ensure that deadheading can only happen on operated flights.

$$\sum_{p\in P} \delta_{F_{p,i}} \geq \sum_{k\in K} \delta_{DH_{k,i}} \quad \forall i \in F \tag{3.5}$$

$$\sum_{p\in P} \delta_{FD_{p,i,t}} \geq \sum_{k\in K} \delta_{DHD_{k,i,t}} \quad \forall i \in F, \forall t \in T \tag{3.6}$$

Aircraft Node Continuity

Aircraft node continuity is ensured by the following three constraints. Constraints 3.7 force all aircraft to leave the first node in the time window, either by flying a (delayed) flight or by utilizing a ground arc. Constraints 3.8 demand all aircraft entering a node to also leave that node. Constraints 3.9 handle the inflow of the aircraft at the last node in the time window. Generally this can be done in two ways, either by fixing the specific aircraft or by fixing the aircraft type at a certain airport. In this model, the former is chosen in order to give the model more flexibility, since tail swaps with aircraft of the same type could be performed in more cases this way. To avoid model infeasibility, a slack variable with a large associated cost is activated in the objective function whenever the constraints cannot be satisfied.

$$\delta_{GF_{p,n}} + \sum_{i\in F_{\text{out}}} \delta_{F_{p,i}} + \sum_{i\in F_{\text{out}}, t\in T} \delta_{FD_{p,i,t}} = 1 \quad \forall p \in P, n = \text{scheduled } N_o \text{ of } p \tag{3.7}$$

$$\left(\delta_{G_{p,n-1}} + \sum_{i\in F_{in}} \delta_{F_{p,i}} + \sum_{i\in F_{in}, t\in T} \delta_{FD_{p,i,t}}\right) - \left(\delta_{G_{p,n}} + \sum_{i\in F_{\text{out}}} \delta_{F_{p,i}} + \sum_{i\in F_{\text{out}}, t\in T} \delta_{FD_{p,i,t}}\right) = 0 \quad \forall p \in P, n \in N_i \tag{3.8}$$

$$\sum_{p\in P(e)} \left(\delta_{GF_{p,n-1}} + \sum_{i\in F_{in}} \delta_{F_{p,i}} + \sum_{i\in F_{in}, t\in T} \delta_{FD_{p,i,t}}\right) + s_j \geq h_n^e \quad \forall e \in E, n \in N_s \tag{3.9}$$

Crew Node Continuity

Crew pairs also have three different constraints ensuring node continuity in the time space network. Constraints 3.10 make sure that all crew pairs leave their starting node either by operating or deadheading a (delayed) flight or by utilizing a ground arc. Similar to Constraints 3.8, Constraints 3.11 demand all crew pairs entering a node to also leave that node. Lastly, Constraints 3.12 fix specific crew pairs at the last node. If this cannot be assured, the slack variable is activated and imposes a large additional cost in the objective function. Again, this is done to avoid model infeasibility whenever the constraints cannot be satisfied.

$$\delta_{GK_{k,n}} + \sum_{i\in F_{\text{out}}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}}\right) \sum_{i\in F_{\text{out}}, t\in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}}\right) = 1 \quad \forall k \in K, n = \text{ scheduled } N_o \text{ of } k \tag{3.10}$$

$$\begin{aligned} &\left(\delta_{GK_{k,n-1}} + \sum_{i\in F_{in}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}}\right) + \sum_{i\in F_{in}, t\in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}}\right)\right) - \\ &\left(\delta_{GK_{k,n}} + \sum_{i\in F_{\text{out}}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}}\right) + \sum_{i\in F_{\text{out}}, t\in T} \delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}}\right) = 0 \quad \forall k \in K, n \in N_i \end{aligned} \tag{3.11}$$

$$\delta_{GK_{k,n-1}} + \sum_{i\in F_{in}} \left(\delta_{K_{k,i}} + \delta_{DH_{k,i}}\right) + \sum_{i\in F_{in}, t\in T} \left(\delta_{KD_{k,i,t}} + \delta_{DHD_{k,i,t}}\right) + s_k = 1 \quad \forall k \in K, n = \text{ scheduled } N_s \text{ of } k \tag{3.12}$$

3.3. Aircraft and Airline Constraints

The second set of equations is related to aircraft properties, airline policy and regulations.

Aircraft Seat Capacity

When swapping planes, the seat capacity of the new aircraft should be large enough to accommodate all the passengers that booked the flight. Constraints 3.13 make sure that the aircraft that do not satisfy the seat capacity requirement cannot operate the flight.

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p, i \text{ where } (\text{SeatsY}_p < \text{PaxY}_i \wedge \text{SeatsJ}_p < \text{PaxJ}_i) \tag{3.13}$$

Aircraft Range
Constraints 3.14 prohibit an aircraft from operating a flight when its range is less than the distance of the flight.

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p, i \text{ where } \left(\text{range}_p < \text{dist}_i\right) \tag{3.14}$$

Penalty Unscheduled Aircraft
When a tail has been swapped and an unscheduled aircraft operates a flight, Constraints 3.15 ensure that a penalty is incurred in the objective function. These constraints make sure that it should be desirable to operate the flights according to the initial schedule. The penalty cost should equal the cost of performing a tail swap.

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} - \delta_{F'_i} = 1 \quad \forall i \in F, p = \text{ aircraft not scheduled for } i \tag{3.15}$$

Tail Swap Time Limit
Because of operational challenges such as turn-around processes and gate allocation, tail swaps cannot be performed T_{swap} minutes before the time of departure of the flight anymore. Constraints 3.16 make sure that aircraft p not scheduled for flight i cannot operate the flight T_{swap} minutes before its time of departure.

$$\delta_{F_{p,i}} + \sum_{t \in T} \delta_{FD_{p,i,t}} = 0 \quad \forall p \in P, i \text{ where } STD_i - T_{now} < T_{swap} \text{ and i} \neq \text{ flight for p} \tag{3.16}$$

Penalty Unscheduled Crew
Similar to Constraints 3.15, Constraints 3.17 impose a penalty in the objective function when a flight is operated by a different crew pair. This favours flights to be operated according to the initial schedule.

$$\delta_{K_{k,i}} + \sum_{t \in T} \delta_{KD_{k,i,t}} - \delta_{K'_i} = 1 \quad \forall i \in F, k = \text{ crew not scheduled for } i \tag{3.17}$$

Flight Time Limit Crew
Strict regulations are imposed on pilots on duty. Constraints 3.18 ensure that crew pairs cannot exceed their maximum flight time during the recovery. If these constraints are violated, a slack variable will be activated to enlarge the crew's maximum flight time, such that the constraints become feasible again. However, this leads to the addition of a penalty in the objective function.

$$\sum_{i \in F} \left(\delta_{K_{k,i}} + \sum_{t \in T} \delta_{KD_{k,i,t}} \right) \cdot FT_i \leq FTL_k + FTM_k \cdot s_{FT_k} \quad \forall k \in K \tag{3.18}$$

3.4. Disruption Implementation

The actual disruptions should also be included in the MILP model, as otherwise the schedule would be operated as planned. Disruptions are added to the optimisation model in the form of constraints, which cancel the relevant decision variables. This makes sure that the disruptions are imposed correctly in the schedule. Four different disruption types can be imposed.

Flight Delay
When a flight is delayed, Constraints 3.19 make sure that all flight arcs up until the length of the delay cannot be flown.

$$\sum_{p \in P} \delta_{F_{p,i}} + \delta_{D_{p,i,t}} = 0 \quad \forall t \in T \leq \text{ delay, } i = \text{ delayed flight} \tag{3.19}$$

Flight Cancellation
When a flight is cancelled, Constraints 3.20 set the cancellation decision variable for that flight equal to one.

$$\delta_{C_i} = 1, \quad i = \text{ cancelled flight} \tag{3.20}$$

Aircraft Unavailability

Another type of disruption is the unavailability of an aircraft for a certain period of time. This could be caused by a mechanical failure, for example. Constraints 3.21 ensure that all flight arcs up until the duration of the unavailability for the specific aircraft cannot be flown.

$$\begin{aligned} \delta_{Fp,i} + \sum_{i \in F} \delta_{D_{p,i,t}} = 0 \\ \forall i \in F \text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i \leq \mathrm{t_{end}} \cup \mathrm{t_{start}} \leq \mathrm{STA}_i \leq \mathrm{t_{end}}), \\ \forall t \in T \text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i + t \leq \mathrm{t_{end}} \cup \mathrm{t_{end}} \leq \mathrm{STA}_i + t \leq \mathrm{t_{end}}) \end{aligned} \tag{3.21}$$

Airport Unavailability

The last form of disruption considered in this research is the unavailability of an airport. Constraints 3.22 prohibit the operation of all flights from and to the unavailable airport in the time period.

$$\begin{aligned} \sum_{p \in P} \left(\delta_{Fp,i} + \sum_{i \in F} \delta_{D_{p,i,t}} \right) = 0 \\ \forall i \in F (\text{where } (\mathrm{t_{start}} \leq \mathrm{STD}_i \leq \mathrm{t_{end}} \cup \mathrm{t_{start}} \leq \mathrm{STA}_i \leq \mathrm{t_{end}})) \cap \left(\mathrm{orig}_i \cup \mathrm{dest}_i \right) = \mathrm{a} \\ \forall t \in T (\text{ where } (\mathrm{t_{start}} \leq \mathrm{STD}_i + t \leq \mathrm{t_{end}} \cup \mathrm{t_{end}} \leq \mathrm{STA}_i + t \leq \mathrm{t_{end}})) \cap \left(\mathrm{orig}_i \cup \mathrm{dest}_i \right) = \mathrm{a} \\ \text{where a = unavailable airport} \end{aligned} \tag{3.22}$$

4
Model Verification

Before performing the case study, it is important to verify the model in order to assess whether it works as designed. Therefore, all the different recovery actions are tested. These are divided into different sections, i.e. schedule, aircraft and crew recovery actions. In total, six scenarios are tested and information regarding these scenarios are depicted in three tables below.

In Table 4.1 a cost overview is shown, which is used in the verification analysis. In the case study the delay costs and the swapping costs differ, but for verification purposes, the values are simplified. Cancellation costs are defined as the maximum delay costs plus an additional $250 per passenger.

Parameter	**Cost**
Delay	$\$1/(min \cdot pax)$
Cancellations	$250/pax
AC Swap	$500
Crew Swap	$1,000
Crew Deadhead	$200
AC Sink Violation	$1,000,000
Crew Sink Violation	$50,000
Reserve Crew Sink Violation	$10,000

Table 4.1: Overview of costs.

In Table 4.2 the number of resources in all the scenario's is shown. The number of passengers is equal to 100 on all flights.

Scenario	1	2	3	4	5	6
AC	1	2	2	2	2	2
Crew	1	3	2	4	1	2
Reserve Crew	0	0	0	0	1	0
Passengers	100	100	100	100	100	100
Connecting Passengers	0	0	0	0	0	50

Table 4.2: Overview of resources in each scenario.

In Table 4.3, the results after recovery are shown in terms of costs. In each scenario, the sum of the individual cost factors should equal the total disruption cost provided by the model.

4.1. Flight schedule recovery

In this section the schedule recovery actions are verified, which include flight delays and flight cancellations.

Scenario	1	2	3	4	5	6
Delay	$4,000	$1,000	$5,000	$2,000	$6,000	$4,000
Canx	$0	$0	$0	$0	$0	$0
Missed Connection	$0	$0	$0	$0	$0	$0
AC Swap	$0	$1,000	$0	$0	$0	$0
Crew Swap	$0	$0	$1,000	$1,000	$1,000	$0
Crew DH	$0	$0	$0	$400	$0	$0
Reserve Crew	$0	$0	$0	$0	$0	$0
AC Sink Viol.	$0	$0	$0	$0	$0	$0
Crew Sink Viol.	$0	$0	$0	$0	$0	$0
Disruption Cost	$4,000	$2,000	$6,000	$3,400	$7,000	$4,000

Table 4.3: Overview of costs in each scenario.

Delay
First of all, the model is tested with a 20-minute delay, without having the opportunity to swap. In such a situation the only available recovery action is to delay the downstream flights as well. Looking at the result of the model in Figure 4.1, the second flight is indeed delayed with the same 20-minute delay.

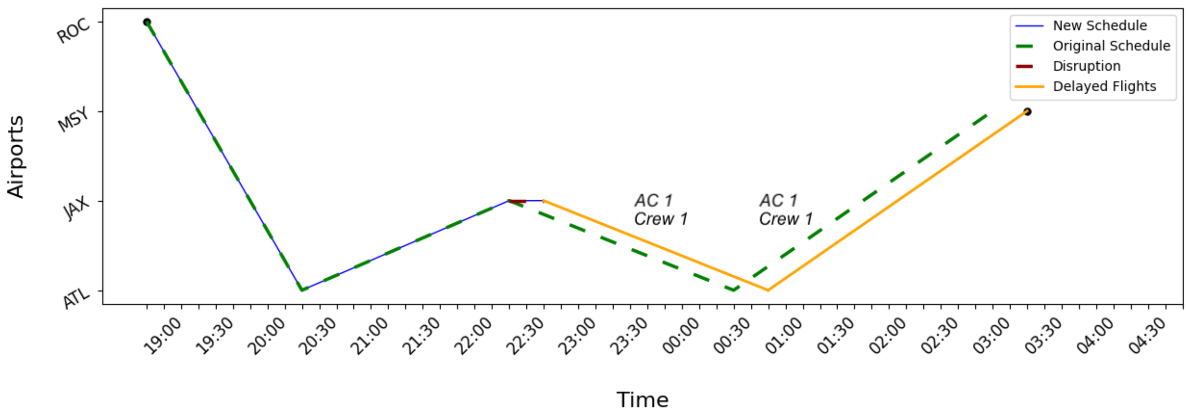

Figure 4.1: 20-minute flight delay.

4.2. Aircraft recovery actions

The only aircraft recovery action considered in the proposed model is swapping, which will be verified in this subsection.

Aircraft Swap
In this scenario, the disruption is a 10-minute delay at New Orleans (MSY). However, since there are no connecting passengers and since there is a swapping opportunity at Los Angeles (LAX), a 10-minute delay for the second flight can be avoided and should be given as the best decision by the recovery model. Indeed, the model uses AC 2 to take over the second flight of AC 1, such that this flight can be operated as scheduled. Crew 3 was originally scheduled to operate this flight and is not disturbed because of the disruption, so in terms of crew pairs no changes were made. The additional costs are $500 with the aircraft swap, whilst the additional costs would equal $1000 if the flight would be delayed.

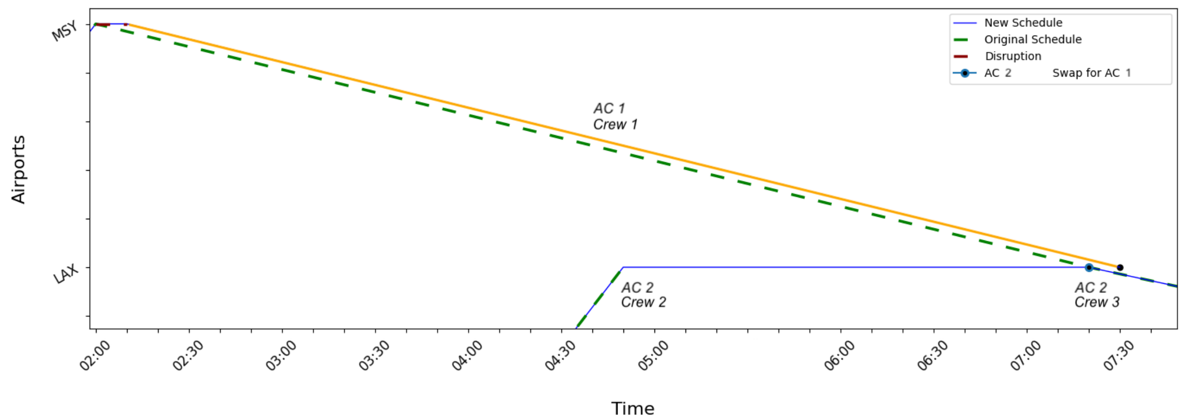

Figure 4.2: Aircraft swap to prevent delay.

4.3. Crew recovery actions

Unlike aircraft, multiple crew recovery actions are considered in the model. Swapping remains an option for crew pairs. Furthermore, deadheading (i.e. changing locations of crew by taking a flight, but not operating it) and using reserve crew, located at strategic airports in case of severe disruptions, are also recovery options.

Crew Swap

In the scenario depicted in Figure 4.3, a flight is delayed by 50 minutes. The problem is that the crew pairs are exceeding their maximum duty time if they would operate the delayed flights, which is not possible. The model takes care of this by swapping the crew originally flying AC 2 with the disrupted crew. As the maximum duty time of Crew 2 lies beyond the arrival time of the disrupted flight, this is a valid recovery action. Both the implementation of the maximum duty time and the swapping recovery action are thereby verified in this scenario.

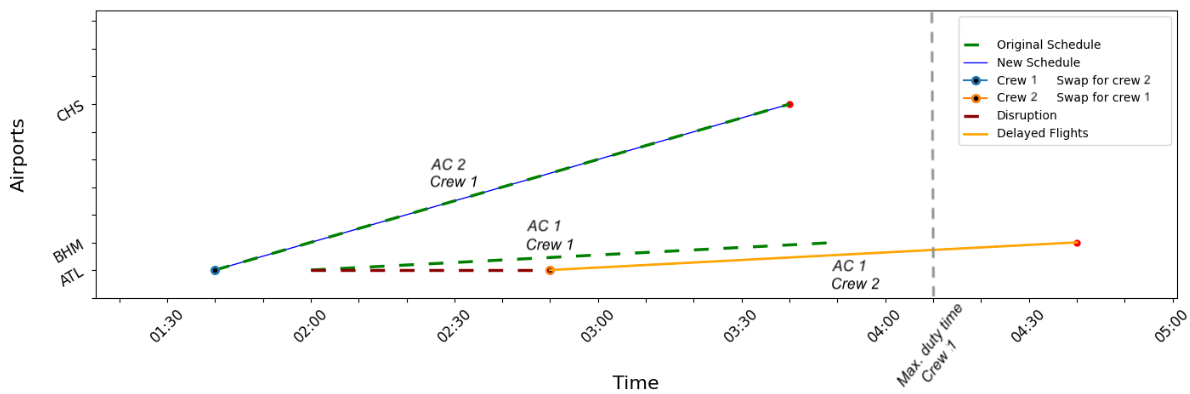

Figure 4.3: Crew pair swap to prevent crossing maximum duty time.

Crew Deadheading

Besides swapping, deadheading is also verified. This case shows a 20-minute delay of the flight from Detroit (DTW) to Orlando (MCO). As the downstream flight requires Crew 1 (but a different aircraft), it cannot be flown as scheduled anymore. However, since a swapping opportunity exists, the model deadheads Crew 2 on the flight from Atlanta (ATL) to Orlando (MCO), such that in Orlando they could be swapped with Crew 1. By doing this, the flight from Orlando (MCO) to Atlanta (ATL) can be flown as scheduled. As Crew 1 should end at Atlanta, it is deadheaded on the last flight from Orlando to Atlanta, such that no crew sink violations occur. The alternative of delaying the first flight from Orlando to Atlanta costs more, as in this case another 20-minute delay will be induced, causing additional costs of $2000. Instead, a crew swap ($1000) and two deadheads ($400) are the actions taken, which results in additional costs of $1400.

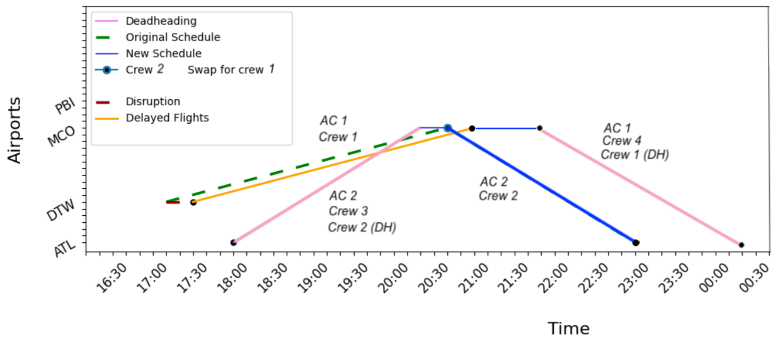

Figure 4.4: Crew deadheading to prevent delay.

Reserve Crew
Lastly, the use of reserve crew is tested. In this scenario depicted in Figure 4.5, a flight from Kansas (MCI) to Atlanta (ATL) is delayed by an hour. This means that the second flight cannot be flown as scheduled, as it requires the disrupted Crew 1 (but different aircraft). However, a reserve crew is located in Atlanta. The model indeed makes this decision and reschedules Reserve Crew 1 on the flight from Atlanta to Fort Lauderdale (FLL), such that this flight can be flown as scheduled. Fortunately, a last flight from Fort Lauderdale to Atlanta is scheduled and operated by Reserve Crew 1 as well, bringing the reserve crew back to its original location. The alternative is delaying two more flights with 20 minutes, resulting in additional costs of $4000, whilst using reserve crew and swapping is limited to additional costs of $1000.

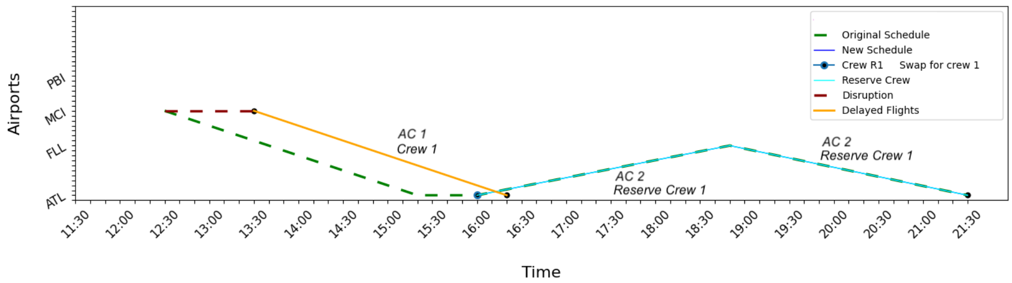

Figure 4.5: Operating reserve crew to prevent delay.

4.4. Connecting Passengers

The model is explicitly programmed to consider aircraft and crew, but it also implicitly considers passenger flows by using the connecting passenger matrix (CPM). In this case, the first flight from New York (JFK) to Atlanta (ATL) is delayed by 30 minutes. In terms of aircraft and crew, this does not have an effect on the second flight shown in Figure 4.6, as this second flight is operated by different resources. However, since many passengers have to take this connecting flight at the hub (ATL) to Tallahassee, the second flight is also delayed. This way, all the connecting passengers could still take their second flight.

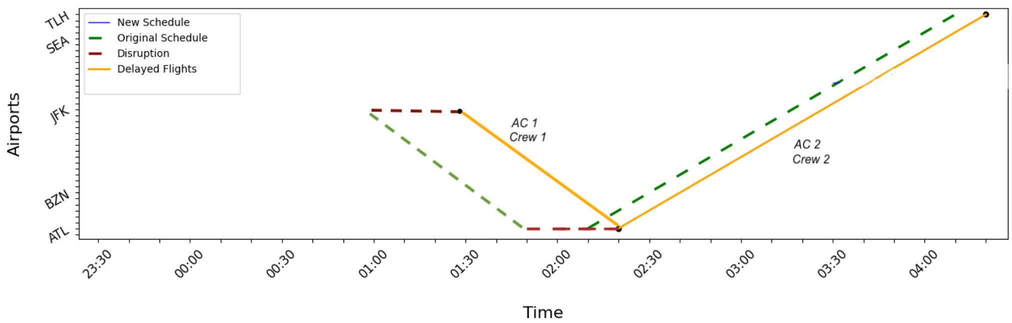

Figure 4.6: Recovering connecting passengers.

5 Case Study

5.1. Delta Airlines Data Set

The airline tested in the case study is Delta Airlines, which is the biggest airline in the world by revenue, profit, assets and market capitalization [7]. It is an American hub & spoke airline, with 8 hubs and a total of 242 destinations in 52 countries, as of 2022. The dataset that is used in this research contains roughly one month of operations in January 2015. In that year, the airline operated a fleet of 800 aircraft consisting of Boeing 717, Boeing 757, Boeing 767, McDonnel Douglas MD-88 and McDonnel Douglas MD-90 (Both MD-88 and MD-90 were phased out in 2020) and performed roughly 2,400 domestic flights per day. Since the computational speed of recovery models is susceptible to large networks, Delta Airlines is a good candidate to perform analyses on. Obtaining positive results for this airline, means that the model is likely to perform well on disruptions from other airlines as well. Part of the dataset was obtained and modified by Hassan [8] and included information about the schedule, aircraft and passengers. However, due to the competitive landscape and difficulties in long-term crew scheduling, airlines generally do not publish their crew rosters. Nikolajević [10] generated this artificially and added the crew roster to the initial dataset created by Hassan.

Flight Schedule

The flight schedule information was downloaded from the 'Reporting Carrier On-Time Performance' database from the United States Department of Transportation [11]. Besides the schedule, the database also contained information on the disruptions that have occurred. Both the consequence (e.g. cancelled and diverted flights) and the causes (e.g. weather conditions) are provided. The complete flight schedule in Q1 2015 was downloaded and processed, which had to be done since the schedule contained timing errors and missing flights. After processing, the schedule was divided in a schedule data set and a disruption data set. Delta Airlines operated 197,000 flights in the first quarter of 2015, which comes down to an average of 2,164 flights per day. Delta Airline has a total of 147 airports, which includes 8 hubs and mainly operates flights from hub to spoke (83% of the flights). The next 13% of the flights are from hub to hub, and the remaining 1% of the flights are between spokes.

Fleet

The flight schedule contains the tail numbers of the aircraft operating the flight, however, information regarding the specific aircraft type was not provided. Information from the N-Number Database of the United States Federal Aviation Administration [6] was used to link tail number to aircraft type. Furthermore, the fleet section on the Delta Website[1] was used to collect data regarding the seat capacity and range of the different aircraft. The direct operating cost (DOC) was computed by using information from the Financial Database of the United States Department of Transportation - Bureau of Transportation Statistics [12]. In 2015, the fleet of Delta Airlines consisted of 800 aircraft, from 8 different families. The distribution over the three largest families is as follows: McDonnel Douglas (32%), Boeing 757 (17%) and Boeing 737 (17%).

Crew

As mentioned earlier, the crew rosters were not publicly available online due to confidentiality and had to

[1] https://www.delta.com/us/en/aircraft/overview

be created by Nikolajević [10] for the purpose of his research. Both an initial crew schedule and a reserve crew schedule were generated. First of all, regulations set by the FAA [5] are stringent and each airline has the possibility to extend these regulations with its own stricter and airline-specific ones. The Delta Airlines additional regulations were publicly available in a crew scheduling handbook [4]. This set of regulations was used to generate a crew schedule, making use of an algorithm that allows flexibility when disruptions occur. The algorithm first generates crew duties, after which a pairing generation algorithm is initiated to complete the scheduling.

After the creation of the crew schedule, Nikolajevi also generated a reserve crew schedule as Delta Airlines also makes use of this strategy to be able to allocate crew in the case of crew absence. Their role is crucial to maintain a feasible flight schedule. The generation of the reserve crew schedule is based on the work of Bayliss et al. [3]. In this work, the probability of scheduled crew being unavailable for a flight was determined, such that a fixed number of reserve crew could be assigned to specific airports to minimize the probability of crew unavailability. Similar to the scheduled crew generation, first the reserve crew duties are generated after which they are combined into pairings.

Passenger
Modelling of the passengers on all flight legs was performed by using the research done by Barnhart et al. [2], who developed methods to model travel for U.S. domestic passengers. Both passengers on direct flights and connecting passengers could be modeled in this way. Moreover, a differentiation was made between economy and business passengers. The average passenger load factor (i.e. the number of passengers divided by the capacity on a flight) is equal to 75.9% and approximately 10% of the flights were full or overbooked. Approximately 26% of all the passengers were passengers on connecting flights.

Disruption
As mentioned before, the flight schedule was divided into an original flight schedule data set and a disruption data set. Only the root causes of disruptions were included and not the ones that were caused by an upstream disruption, as finding a new schedule is the purpose of the integrated recovery model. The data set only holds information on the duration of the delay in minutes (i.e. no cancellations) and the causes of the delays, including carrier delay, late aircraft delay, national air system delay, security delay and weather delay. The dataset does not contain information about the time found out (TFO) of the delays, and was generated by Hassan [8] for the purpose of his research. Aircraft with a delay of more than 120 minutes were classified as aircraft unavailabilities. In Q1 2015, 29,400 disruptions occurred in the Delta Airlines schedule. Aircraft unavailabilities only represent 3% of the disruption in occurrence, but account for 20% of the delays in terms of minutes.

5.2. Results

The proposed integrated model (ML IDS) is compared to the globally optimal solutions from the IDS and to the solutions produced by the sequential disruption set solver (SDSS) developed by Hassan and Nikolajević. The SDSS does not recover the resources simultaneously, but recovers one resource at a time. This reduces the computational complexity, but does not always result in optimal solutions. The SDSS considers the same recovery actions as the IDS and also makes use of the CPM to take into account passenger flows as well.

The results are gathered by providing the models with disruptions experienced on (one day of operations, containing 365 disruptions instances) and analysing the recovery decisions taken using different KPI's. All KPI's used are related either to solution time or solution quality and give a nuanced assessment of the performance of the model. Table 5.2 depicts the aggregated results for all the 365 instances.

The average disruption cost in the optimal solution is equal to $\$11,638$, which is three times less than the average disruption cost of the ML IDS ($32552). This is mainly caused by the additional cancelled flights, and aircraft & crew sink violations, which have the highest associated penalties ($\$550/Pax$, $\$1,000,000$ and $\$50,000$, respectively). Since the SDSS used a slightly different cost set, the disruption costs of the SDSS are not comparable. However, the cost distribution in the objective function associated with all the decision variables is very much the same in all models, and hence a fair comparison can be made in terms of the other KPI's.

	IDS	ML IDS	SDSS
Time (Avg.)	1092	70	65
<120 (%)	13%	96%	85%
Optimal (%)	100%	58%	35%
Disruption Cost (Avg.)	11638	39855	N/A
Aircraft (Avg.)	138	46	70
Crew (Avg.)	196	56	99
Cancelled Flights (Sum)	15	25	≥42
Cancelled Passengers (Sum)	1721	2825	≥4937
Average Delay (Avg.)	30.5	31.5	32.1
Delayed Passengers (Avg.)	417.9	310.7	239.1
Missed Connections (Avg.)	12.1	12.5	11.9
Aircraft Sink Violations (Sum)	1	3	≥12
Crew Sink Violations (Sum)	57	92	187
Crew Flight Time Violations (Sum)	0	0	0
Infeasibilities (Count)	0	0	16
Severe Infeasibilities (Count)	0	0	2

Table 5.1: Results to 365 disruptions instances.

	IDS	ML IDS	SDSS
Aircraft Swaps (Sum)	600	351	214
Crew Swaps (Sum)	220	169	371
Crew Deadheads (Sum)	99	38	105
Reserve Crews used (Sum)	71	56	118

Table 5.2: Recovery actions to the 365 disruptions instances.

The ML IDS considered the fewest resources (around 30% of the network), which is required to compute solutions in under two minutes. The sequential model considers approximately 50% of the network, which is more than the ML IDS but acceptable in terms of computational runtime.

The number of passengers on cancelled flights is in line with the number of cancelled flights. The optimal solution returns 15 cancelled flights and 1721 passengers on a cancelled flight, while the ML IDS returns 25 and 2825 passengers on a cancelled flight. Furthermore, the SDSS returns more than 42 cancellations and more than 4937 passengers on cancelled flights.

On average, the number of passengers that missed their connection is almost the same for all three models. All models use the same connecting passenger matrix (CPM), which induces a large cost whenever passengers miss a connection. Hence, it is not a surprise that the models react the same in terms of passengers that missed their connections. The slightly lower missed connections for the SDSS can be explained by the fact that on average less passengers are delayed using this model, as can be seen by the delayed passengers performance indicator.

Flight hour violations occur when the flight hours flown exceed the initial maximum flight hours. If this happens, a penalty is incurred ($20.000 in this case study) and an additional flight time is added (2 additional hours in this case study) to the initial maximum flight time of the crew. This extended maximum flight time is the stringent limit and cannot be exceeded. However, none of the three models uses this option in the 365 instances of the case study, as it is not deemed necessary.

Infeasibilities are a drawback of solving disruptions sequentially. The reason for this is that the schedule is optimized at first, without considering the crew pairs in the network. This may disturb the crew flow in the network, which requires additional recovery actions in the crew stage to make the schedule feasible again. This phenomenon can also be seen in Table 5.2, which shows that more crew recovery actions are taken by the SDSS in comparison with the (ML) IDS. However, in some cases these recovery actions are not sufficient resulting in crew allocation infeasibilities, meaning that the flight cannot be flown anymore. Moreover, if downstream flights are scheduled to be operated by the aircraft and/or crew pairs scheduled for the now cancelled flight, this will result in even more cancellations. A distinction has been made in terms of the severity of the infeasible schedule, such that an estimate can be made on the additional cancelled flights, passengers on cancelled flights and aircraft sink node violations. Three different types are created.

1. The infeasible crew allocation is related to the last flight of the aircraft in the time window. In this case the result is one additional flight cancellation and one additional aircraft sink node violation.
2. The infeasible crew allocation is related to an aircraft that is scheduled to fly back directly to the airport where the infeasible crew allocation occurred. In this case the result is two additional flight cancellations and no aircraft sink node violation (as the aircraft returns to the airport where it already is stationed).
3. The last case belongs to the rest of the scenarios, i.e. the aircraft related to the infeasible crew allocation is scheduled to fly to other destinations after its first cancelled flight. In this case the result is three or more cancellations and 0 or more aircraft sink node violations. These are the severe infeasibilities, because it may happen that many flights will get cancelled as a result of infeasible aircraft allocations.

In terms of the case study, this resulted in the numbers shown in Table 5.3. These additional outcomes were added to Table 5.1 in order to achieve the final KPI's.

Additional consequences	**SDSS**
Cancellations	≥23
Aircraft Sink Violations	≥9

Table 5.3: Additional consequences due to infeasible crew allocations.

Because of the simultaneous nature of the IDS and the fact that crew flows are considered in the optimization of the schedule, the model is able to generate a feasible schedule in most cases (no infeasible schedules in the case study).

5.3. Sensitivity Analysis

Two sensitivity analyses are performed. In the first analysis, the parameters regarding the delay costs are increased to make the model more customer oriented. In the second analysis, the subnetwork size is increased to assess the effect on disruption costs and solution time.

Customer-Oriented Model

The results in Section 5 are promising, as it shows that it is efficient in terms of computational time and solution quality to use a subnetwork created by a machine-learned ranker. However, the sequential model performs better on the delay indicators in comparison to the optimal solution and the ML IDS. This can be explained by the fact that the decisions of the model depend on the cost factors in the objective function. The IDS is more concerned with minimizing cancellations and sink node violations, as the costs associated are much higher than delaying passengers. Moreover, the model does not penalize short delays too much, as the delay costs increase faster with longer waiting times. Since the average delay per delayed passenger is limited to 30 minutes after recovery in this case study, these decisions are tolerated. However, if airlines would like to have a more customer-friendly recovery model, i.e. a model that considers cancellations and passenger delays even more, this can be achieved by increasing the costs associated with the passengers.

In this sensitivity analysis, the passenger delay costs are increased to assess the change in the decision-making process of the model. In the first study the delay costs are multiplied by five and in the second study 200$ is added to all the delay costs, as depicted in Figure 5.1.
It should be noted that both studies have a different focus. In the first study, the effect of the increased delay costs is significant for the longer delays, whilst in the second study the increase is more significant for the shorter delays. This difference is also visible in the results, which are depicted in Table 5.4. In the first study the number of cancellations decreased from 25 to 22, since cancellation costs are based on the final delay

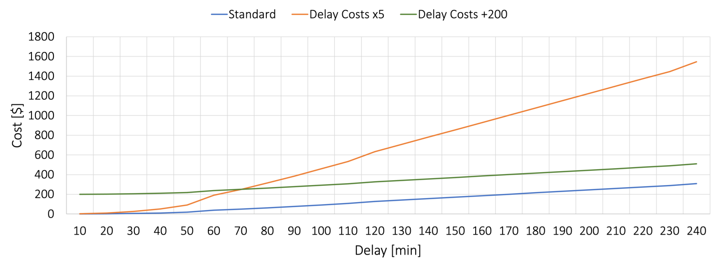

Figure 5.1: Passenger delay cost increase.

costs plus an additional penalty. However, the combination of average delay per passenger and number of delayed passengers has not decreased. This is because an average delay of 30 minutes is still considered short and the costs associated with it are low, even with a delay cost multiplication of five. Airlines seeking fewer cancellations could choose for this set-up, but the positive results regarding the number of cancelled flights have gone at the expense of 2 more aircraft sink violations, 19 more crew sink violations and more recovery actions when compared with the standard configuration. In the second study, the number of cancellations has remained the same, but the number of delayed passengers has decreased significantly, from 310.7 to 244.6. This is great for airlines seeking fewer delays, but has gone at the expense of 10 more crew sink violations and an increase of all recovery actions. As can be concluded from this sensitivity analysis, preventing cancellations is more difficult and causes more airline operations difficulties than preventing delays. Lastly, airlines seeking both fewer cancellations and fewer delays, should increase both the short and long passenger delay costs.

	Standard	**Delay Costs x5**	**Delay Costs +200**
Cancelled Flights (Sum)	25	22	25
Cancelled Passengers (Sum)	2825	2233	2825
Average Delay (Avg.)	31.5	31.3	33.7
Delayed Passengers (Avg.)	310.7	309.1	244.6
Missed Connections (Avg.)	12.5	12.2	12.2
Aircraft Sink Violations (Sum)	3	5	3
Crew Sink Violations (Sum)	92	111	102
Crew Flight Time Violations (Sum)	0	0	0

Table 5.4: Standard and customer-oriented ML IDS comparison.

	Standard	**Delay Costs x5**	**Delay Costs +200**
Aircraft Swaps (Sum)	351	364	429
Crew Swaps (Sum)	169	185	196
Crew Deadheads (Sum)	38	36	40
Reserve Crews used (Sum)	56	61	64

Table 5.5: Standard and customer-oriented ML IDS recovery actions.

Increasing Subnetwork

A disruption instance with four disrupted flights is recovered with different subnetwork sizes to assess the effect on solution time and disruption cost. Although this is only one disruption instance, Figure 5.2 shows that the disruption costs decrease and the solution times increases with larger subnetworks. In this case, the disruption costs have already decreased to near-optimal with less than 50 aircraft in the network, meaning that a high-quality solution is realised with an acceptable runtime.

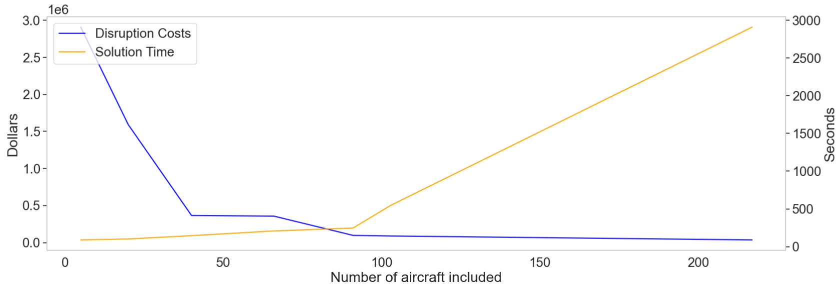

Figure 5.2: Solution time and disruption costs with increasing subnetwork size.

Bibliography

[1] Michael Ball, Cynthia Barnhart, Martin Dresner, Mark Hansen, Kevin Neels, Amedeo Odoni, Everett Peterson, Lance Sherry, Antonio Trani, and Bo Zou. Total delay impact study, 10 2010. URL https://isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf.

[2] Cynthia Barnhart, Douglas Fearing, and Vikrant Vaze. Modeling passenger travel and delays in the national air transportation system. *Operations Research*, 62, 05 2014. doi: 10.1287/opre.2014.1268.

[3] Christopher Bayliss, G. De Maere, Jason Atkin, and M. Paelinck. Probabilistic Airline Reserve Crew Scheduling Model. In Daniel Delling and Leo Liberti, editors, *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 25 of *OpenAccess Series in Informatics (OASIcs)*, pages 132–143, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-45-3. doi: 10.4230/OASIcs.ATMOS.2012.132. URL http://drops.dagstuhl.de/opus/volltexte/2012/3709.

[4] Delta Airlines. *Delta Pilots' Scheduling Reference Handbook*, 2020.

[5] Federal Aviation Administration. *Flightcrew Member Duty and Rest Requirements: RIN 2120-AJ58*, 2011.

[6] Federal Aviation Administration. N-number inquiry database, 2018. URL https://registry.faa.gov/aircraftinquiry/. Online; accessed 23-September-2018.

[7] Forbes. The global 2000, 2022. URL https://www.forbes.com/global2000. Online; accessed 20-June-2022.

[8] Lotfy K. Hassan. Aircraft disruption management increasing performance with machine learning predictions. Master's thesis, Delft University of Technology, 2018. URL http://repository.tudelft.nl/.

[9] N. J. M. Hoeben. Dynamic crew pairing recovery. Master's thesis, Delft University of Technology, 2018. URL http://repository.tudelft.nl/.

[10] Andrej Nikolajević. Airline disruption management - a machine learning approach. Master's thesis, Delft University of Technology, 2021. URL http://repository.tudelft.nl/.

[11] United States Department of Transportation - Bureau of Transportation Statistics. Reporting carrier on-time performance database (1987-present), 2018. URL https://www.transtats.bts.gov/DatabaseInfo.asp?DB_ID=120. Online; accessed 23-September-2018.

[12] United States Department of Transportation - Bureau of Transportation Statistics. Air carrier financial: Schedule p-5.2, 2018. URL https://www.transtats.bts.gov/DatabaseInfo.asp?DB_ID=135. Online; accessed 12-November-2018.

[13] J. Vink, B.F. Santos, W.J.C. Verhagen, I. Medeiros, and R. Filho. Dynamic aircraft recovery problem - an operational decision support framework. *Computers and Operations Research*, 117, 5 2020. ISSN 03050548. doi: 10.1016/j.cor.2020.104892.

[14] Hans-Wieger Vos, Bruno F. Santos, and Thomas Omondi. Aircraft schedule recovery problem - a dynamic modeling framework for daily operations. *Transportation Research Procedia*, 10:931–940, 2015. ISSN 23521465. doi: 10.1016/j.trpro.2015.09.047.

[15] Charles Walker. Coda digest q3 2017, 2017. URL https://www.eurocontrol.int/sites/default/files/content/documents/coda-digest-q3-2017.pdf.