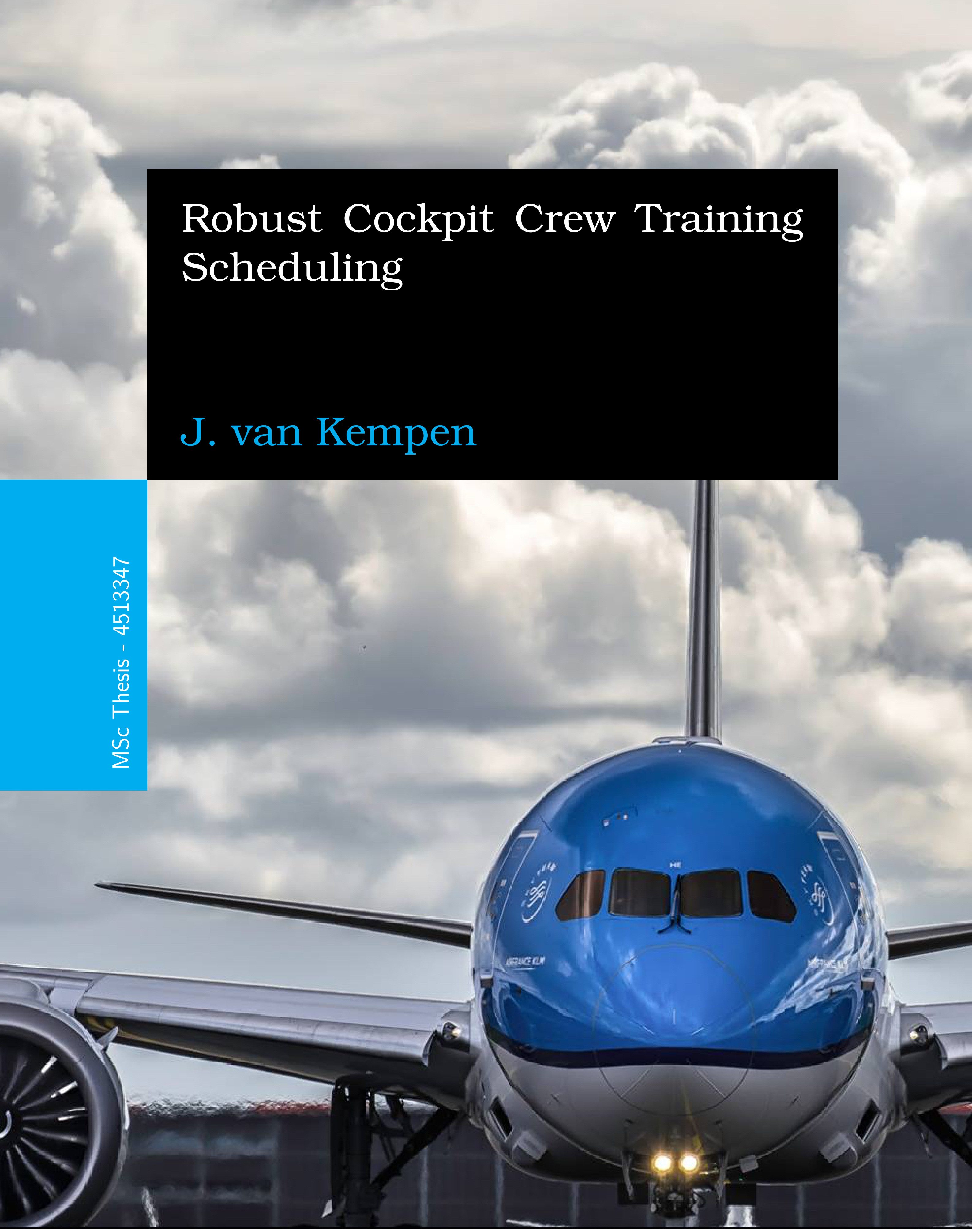

# Robust Cockpit Crew Training Scheduling

J. van Kempen

MSc Thesis - 4513347

TU Delft Delft University of Technology

Faculty of Aerospace Engineering

# Robust Cockpit Crew Training Scheduling

by

J. van Kempen

For the degree of Master of Science at the Delft University of Technology,
to be defended publicly on Friday July 19, 2019 at 10:00 AM.

| | | |
|---|---|---|
| Student ID: | 4513347 | |
| Thesis committee: | Prof. dr. R. Curran | TU Delft |
| | Dr. ir. B. F. Lopes dos Santos | TU Delft |
| | Dr. F. Avallone | TU Delft |
| | Ir. drs. L. Scherp | KLM |

Faculty of Aerospace Engineering (AE) · Delft University of Technology

TU Delft
Delft
University of
Technology

# Executive Summary

Airline schedules are constantly being updated to deal with disruptions originating from crew absence, upstream delays, mechanical failures and other sources. According to Belobaba et al. (2015), disruptions can propagate in the schedule and when resources are scarce. Such disruptions can become very costly. Numerous researchers such as Barnhart et al. (2003) and Kohl et al. (2007) showed that significant savings are possible when efficiently dealing with disruptions. This field of study is known as disruption management and deals with (1) robustness and (2) recovery. Recovery focuses on disruptions re-actively by efficiently restoring schedule feasibility. Robustness is defined as the capability to deal with- or absorb negative effects of unexpected events and is accounted for proactively. Clausen et al. (2010) listed slack, reserve crew, simplified schedules and swap opportunities as examples of robustness. Each of these robustness indicators is studied extensively, especially for crew as this is one of the most expensive resources employed by the airline. However, all researchers examined robustness of crew assigned to flight pairings.

Although the primary task of cockpit crew is to operate these flights, large airlines invest up to thousands of working days in simulator-based crew training for legal licensing purposes (recurrent training) and crew conversion between fleets and ranks (conversion training) (Quintiq, 2017). On top, the crew is trained by a specially qualified subset of cockpit crew members known as instructors. Each training events thus takes away resources from production. Any disrupted training activity due to illness or leave (such as maternity leave, parental leave or care related leave) potentially impacts crew availability by means of missed due dates in case of recurrents or postponed employment in case of conversion. Constructing a robust cockpit crew training schedule can thus improve on the airlines' operational performance by minimising the impact (i.e. cost) of disruptions.

The current scientific body of knowledge has limited research into cockpit crew training scheduling. Many researchers only consider segregated training scheduling problems for conversion training, recurrent training or instructor assignment. Resource dependencies are all neglected. In the absence of a well-performing integrated training scheduling model, robustness is not considered yet. This thesis addresses the robust cockpit crew training scheduling problem. The research objective is to make recommendations on proactive robustness with respect to a cockpit crew training schedule. To attain this objective, (integrated) models and solution methods are developed and applied and proactive robustness measures are identified and evaluated. All is based on historical crew training data as the application is novel.

The research question is captured in a research framework. It consists of a Training Scheduling & Assignment Model (TS&AM), a Disruption Generator (DG) and a Rule-Based Recovery (RBR) model. Each of these models and methods is inspired on existing literature. The TS&AM integrates scheduling of courses and assignment of trainees, instructors and simulators. The output

roster serves as input for a data-driven disruption generator based on Monte-Carlo Simulation. The disruptions are then solved using a Rule-Based Recovery (RBR) algorithm. Both a Proportional Feedback (PF) and Neural Network (NN) algorithm are tested separately to learn from the output of the recovery model and regenerate a robust variant of the schedule using the same TS&AM. The initial roster serves as a benchmark whilst the robust rosters are disrupted and recovered according to the same process. Based on a comparison of the ability of the schedules to deal with disruptions, recommendations are composed on how to make a cockpit crew training schedule robust.

The TS&AM schedules courses and simultaneously assigns trainees, instructors and simulator slots. The objective is to minimise the sum of schedule cost incurred by each of these resources. A constraint is added to schedule a training course in a portion of the available simulator slot corrected for training demand. Other constraints prevent violation of resource limits and that airline requirements are met on the number of instructors and their respective qualifications. An annual training schedule for a realistically sized problems is generated in five minutes.

Robustness of the output schedules is demonstrated using repeated, stochastic simulation of disruptions. The model mimics the dynamic nature of illness and leave by accounting for trends across the year. The model also caters for dynamic disruption lengths. For realistically sized problems, a disruption scenario is generated in approximately 0.1 second. The RBR model then takes approximately 0.1 second to solve this scenario by searching for the first available recovery action. Recovery options such as to use reserve crew, swapping instructors and trainees and cancellation are included. The application of each recovery action and associated cost are output per disruption and disruption scenario.

The output of the recovery model is used in the feedback loop to update the TS&AM in order to generate a robust training schedule. A set of feature values of each specific disruption and associated roster state is computed. The Proportional Feedback algorithm transforms the feature values into the expected recovery cost of that disruption via a proportionally weighed scheme. On the other hand, the Neural Network tries to learn nonlinear relationships between the features to estimate the recovery cost. The TS&AM is then solved with a combination of the deterministic schedule cost and stochastic recovery cost.

From the conducted experiments, one is highlighted in which the risk of missing due dates is expressed in monetary value. Next to attributing cost of the recovery action to the disruption, the experiment uses a cost structure in which missed due dates are penalised linearly with the number of days until re-qualification of that crew member. Additionally, the sensitivity of the number of available instructors is researched to quantify the effects of operating closer to resource limitations.

The experiment showed, that the larger the cost attributed to the disruption, the larger the projected gain in robustness. The PF algorithm still performed best, followed by the NN algorithm. Both outperformed the benchmark roster. The gain in robustness of the PF and NN algorithms with respect to benchmark schedule amounts 16.75 and 10.93 percent respectively. The gain in robustness negates any (marginal) increase in schedule cost triggered by the TS&AM objective on robustness as opposed to a pure efficiency objective. The total cost of the PF schedule is 1.11 percent lower than that of the benchmark schedule and 0.63 percent lower for the NN roster. The respective gain in stability is 28.50 and 24.97 percent. In more practical terms, this means that robust roster leads to a projected saving of up to 16.75 percent on recovery cost or cost of crew unavailability due to missed due dates with respect to a roster that is generated using the same method and model, but a different objective. It also translate to a 1.11 percent cheaper operations of the training schedule.

The sensitivity analysis on the number of instructors puts the conclusions of the research into perspective. Under the assumption that the added salary cost of an instructor (with respect to a regular crew member) can be attributed to the training schedule, a reduction in total cost per assignment can be achieved. In fact, the total cost per assignment can be reduced by 7.67 percent when reducing the amount of instructors by 20 percent for each qualification. It comes at the cost of a 29.61 percent decrease in robustness. On the other hand, increasing the number of instructors improves robustness by 2.80 percent while increasing the total cost per assignment by 10.74 percent. From the sensitivity analysis can be concluded that the gain in robustness is marginal as long as the number of instructors can be optimised. The importance of robustness only increases when the amount of instructors decreases towards the limits.

Following the experiment and the sensitivity analysis, it is concluded that the PF and NN algorithms are both viable methods of generating a robust cockpit crew training schedule. Despite the larger potential of the NN to learn nonlinear relationships, it is outperformed by the PF algorithm. Further research into the NN could reveal its true potential. Both methods show similar behaviour into obtaining a higher level of robustness, which is translated into recommendations. Assigning overqualified instructors shows a moderate correlation to robustness. The same applies to assigning high(er) levels of unique instructors, each covering less courses. All is aimed at increasing the swap opportunities in the recovery process. Another method to increase swap opportunities is to simplify the schedule by assigning similar courses on the same day. Lastly, a stable schedule is beneficial for robustness, but the TS&AM must remain sufficiently agile to deal with inherent peaks in training demand and resource supply, also on a course specific basis and per instructors qualification. This stability ensures that a balance is obtained between resources used for training and for non-training duties. Crew assigned to the latter category can then be used for swap opportunities or as reserves. Note that further research into optimal rates and levels of assignment is needed to expose additional benefits in robustness.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Irregular airline operations cause schedules rarely to be executed as planned. Disruptions originate from crew absence, upstream delays, mechanical failure of aircraft and many other sources and can have a propagating disruptive effect in the entire network on a daily basis (Belobaba et al., 2015). According to Barnhart et al. (2003) and Kohl et al. (2007) significant savings are possible when handled efficiently. This field of study is known as disruption management and has two established directions: (1) robustness and (2) recovery. Recovery deals with disruptions re-actively by efficiently restoring schedule feasibility. On the flip side, when generating a robust schedule, disruptions are taken into account proactively. Clausen et al. (2010) listed slack, reserve crew, simplified schedules and swap opportunities as examples of accounting for easier and more efficient resolution of future disruptions.

As crew is one of the biggest airline expenses (Belobaba et al., 2015) and crew disruptions due to illness, leave and potentially other causes are unavoidable, researchers devoted energy to crew disruption management. The research aims at minimizing impact of expected crew disruptions on crew availability and thus the operability of the schedule. This has proven to work via all of the mentioned ways of achieving robustness (Kohl et al., 2007), but only for the flight schedule.

Other activities are neglected, of which crew training is an important one. Crew training deals with satisfying legal requirements of maintaining crew members 'current' by training them in a simulator up to a four to six times a year Sohoni et al. (2003). When past-due, the crew member is not allowed to operate until satisfying the requirements again. This puts a strain on crew availability next to the inherent pressure on crew availability through training. The trainees are unavailable to operate revenue-generating flights whenever in training, and on top, training is taken care of by other crew members who have additional instructor qualifications. Furthermore, airlines (need to) cater for crew members to move-up in rank and transition to operate a different aircraft type when possible. According to Sohoni et al. (2004), such transitions requires between five and 13 days of simulator training under supervision of an instructor. Combining research shows that a crew member could be involved in training in up to seven percent of the available productive time per year. The high cost involved with this, the potential impact on crew availability and unexplored effect of robustness on the crew training schedule in academic literature give rise to a research question. This thesis aims to answer this question, which is defined as:

*What robustness measures can be taken to proactively minimize the impact of disruptions related to an airline cockpit crew training schedule considering relevant rules and regulations?*

The research objective is to make recommendations on increasing proactive robustness with respect to crew training schedule by identification and evaluation of proactive robustness measures and solution methods that utilize historical training disruption data.

To answer the main research question, a set of novel algorithms is developed. Starting, a novel model- and solution methodology are applied to quickly generate a cockpit crew training schedule. The schedule is then subjected to a series of disruptions and a recovery algorithm restores schedule feasibility for each disruption scenario individually. The output of the recovery model is used in two algorithms: (1) a proportional feedback algorithm, and (2) a neural network. Both algorithms are used as a learning mechanism to improve the scheduling model on robustness. The novel approaches of achieving robustness via a learning mechanism are compared for their suitability and effectiveness. Afterwards, performance is tested on a real disruption scenario from a major European airline and tested against the performance of the actual airline's cockpit crew training schedule. Based on this, recommendations are made on how to achieve crew training schedule robustness efficiently and effectively.

In chapter 2 existing literature is studied in more detail with the objective of defining and arguing the research gap. A supplementary review of existing literature is provided in chapter 3 with the objective of closing the identified gap using the existing body of knowledge, both from the airline scheduling domain and others. The knowledge and inspiration forms the basis of the research design as presented in chapter 4. This also includes an overview of how various models are linked together. The cockpit crew training scheduling model and solution methodology is presented in chapter 5. chapter 6 describes the primary disruption generator and chapter 7 contains a description on the associated recovery model. The combination of models is tested in a series of experiments in chapter 8. The performance is put into perspective by conducting a sensitivity analysis in chapter 9. Finally, the conclusions following from this project, along with a number of recommendations and a discussion, are treated in chapter 10.

Chapter 2

# Research Gap in Airline Crew Training

As explained in the introduction, this thesis focuses on robustness of the cockpit crew training scheduling problem. The line of argumentation to define this research gap is set out throughout this chapter. In section 2-1, the current understanding, advances and shortcomings in the domain of airline crew training scheduling are analyzed. Focus is shifted to the domain of disruption management in section 2-2, where the sub-domain of robustness is treated as well. Robust schedules aim to outperform deterministic crew schedules in terms of sensitivity when taking into account the stochastic operating environment. The state-of-the-art of both airline crew training and disruption management forms the basis to synthesize on the gap in existing research in section 2-3.

## 2-1 Airline Crew Training Scheduling

To legally allow pilots to fly, each flight crew member must complete numerous training programs throughout their career. These crew training activities have various objectives and different frequencies. Belobaba et al. (2015) describe that these frequencies are generally airline specific. Nonetheless, Yu et al. (2004) defined three overarching categories of airline crew training described below. Note that many terms exist, but the ones below are used throughout this thesis.

- **Conversion training:** Each crew member undergoes initial qualification training each time that person transitions to operate a new aircraft type or operate in another rank (Yu et al., 2004). Typical conversion training programs take 4 weeks (Belobaba et al., 2015). At least 4 to 6 weeks of route instruction follows, in which the trainee flies the aircraft together with a certified instructor (Kohl and Karisch, 2004). At Continental Airlines, as much as 15 to 20 percent of all pilots were awarded a new position every half year and undergo conversion training accordingly (Yu et al., 2004).

- **Recurrent training:** Each pilot must complete annual recurrent training (Sohoni et al., 2004), typically consisting of one- or two full days of ground- and simulator training (Yu et al., 2004). According to Xu et al. (2006), sixty percent of all training resources are devoted to providing recurrent training.

- **Re-qualification training:** Pilots that were unable to fly for a prolonged period of time must complete re-qualification training (Yu et al., 2004). Such a program can take several days or a couple of weeks consisting of a mix of ground, simulator and route instruction (Yu et al., 2004). Both frequency and structure are stochastic.

The three categories of airline crew training are constituted from ground, simulator and route training. Ground training consists of classroom sessions about operating the aircraft, safety, medical checks and more. Simulator training involves getting used to the aircraft and/or practicing abnormal- or emergency situations. Finally, route instruction programs aim at familiarization with the aircraft and demonstrating a sufficient level of proficiency. The latter, also known as Line Flying Under Supervision (LFUS), is operated by a trainee in the role of a regular crew member as indicated in its terminology. Only for the line check at the end of the route instruction program, generally accounting for 15 per cent of the route instruction program, an extra crew member should be scheduled. This route instruction program is least restrictive in resources. Recovery is easier than for simulator training and robustness is thus most interesting for the latter. Simulator training is restricted in both training device capacity and instructor capacity. According to Qi et al. (2004), both the trainees and instructors are drawn from the pool of regular pilots for the entire duration of training. As this makes airline crew simulator training an expensive operation, focus is directed towards airline crew simulator training scheduling. More specifically, the two distinctive problem formulations identified from literature, being segregated models and integrated models, are treated in subsection 2-1-1. Afterwards, in subsection 2-1-2 airline crew training is put into perspective of the bigger picture of the crew scheduling problem.

### 2-1-1 Segregated- and Integrated Airline Crew Training Scheduling

Scheduling simulator training activities is not straightforward considering the complex governing rules described by Kohl and Karisch (2004). Examples of such rules are the various crew compositions allowed for the different types of training, the required qualifications and number of instructors, the geographical location of training resources and many more airline specific requirements (Kohl and Karisch, 2004). They also describe that a common objective for European airlines is to optimize for a combination of cost and crew satisfaction. Robustness is mentioned as a possible objective also, but the authors do not provide further details.

Because conversion training programs decrease the amount of available pilots considerably (i.e. up to 3.5 percent of the airlines' pilots are involved in conversion training at any given time), early research focused on optimizing manpower planning (Yu et al., 1998). They forecast staffing needs and proposed a heuristic approach. The heuristic is used to identify new-hires and pilot transitions plus to schedule the associated conversion training activities. Their model accounted for hard constraints such as capacity and availability of training resources and soft constraints such as seniority rules and optimal crew complementing. Yu et al. (2004) describe a commercialization of this earlier manpower planning research. They applied a deterministic, more integrated approach. Their model is termed integrated, but only limited information about the inter-dependency between training capacity restrictions and planning pilot transitions is taken into account. They claim to optimize instructor schedules and account for recurrent training. However, the former is not explicitly included in the proposed model whilst the latter is only incorporated by setting aside the left-over resources. As a result, the model of Yu et al. (2004) cannot ensure sufficient capacity for recurrent training.

Sohoni et al. (2003) focused on deterministic scheduling of recurrent training activities instead of conversion training as done by Yu et al. (1998, 2004). Sohoni et al. (2003) mention the individual needs for recurrent training, but decided to use a set of standardized training programs instead, known as yardsticks. A yardstick is a standardized training program specifying (day-to-day) training events based on fleet, crew position and composed set of pilots. Every incomplete crew (i.e. a crew composition other than captain, first officer and potentially a second officer) requires a help-out pilot, which adds cost. Sohoni et al. (2003) assume that only instructors could be assigned as help-out, which is preferred for practical reasons but not necessarily covers airline policy. A regular or reserve crew member could also be summoned as help-out (provided that it is in agreement with the regular crew member). The assumptions aside, the model allows for various priority levels trying to prevent pilots from missing their due dates.

Holm (2008) took the model of Yu et al. (2004) and some influences of Sohoni et al. (2004) and combined it to account for both recurrent and conversion training. She made assumptions such as to exclude help-out instructors and allowed for only two kinds of crew compositions. However, later on in her report, she did provide model adaptations such as to allow below-rank flying and instructor training sessions. It remains unclear in what form she implemented and tested the model. Given the lack of valid comparisons, the integrated model can only be hypothesized to improve solution quality.

Holm (2008) constructed sets of trainees, but did not assign, nor proposed to assign, instructors to the training activities despite Sohoni et al. (2004) stressing the importance. A large inter-dependency exists in scheduled training activities and the scarcity in qualified instructors. Xu et al. (2006) are one of few to address the flight instructor scheduling problem and target to cover a set of training activities by a qualified instructor. They assumed that all conversion training events are pre-assigned. In other words, they are already fixed in the schedule and the remaining recurrent training activities need to be scheduled around it. Extending the model to an integrated approach for both recurrent and conversion training is fairly straightforward when combining it with training scheduling models presented in other literature such as Holm (2008).

Summarizing, airline crew training scheduling has received little attention in research despite training being an expensive part of airline operations. Proven segregated and (more) integrated approaches have been researched by, among others, Holm (2008) and Xu et al. (2006). However, all models only proved to perform well under deterministic operating conditions. The impact of disruptions is omitted and so is the inter-dependency with the flight schedule. Crew training requirements bound the availability of crew members for operating regular flights and operating flights limits the amount of available instructors for training purposes.

### 2-1-2 Airline Crew Training Scheduling as Part of Airline Crew Rostering

The link between airline crew training and flights becomes more apparent when zooming out to the larger domain of airline crew rostering. According to Kohl and Karisch (2004), the difference is that training is only one of many inputs of the overarching problem of crew rostering, as indicated in Figure 2-1.

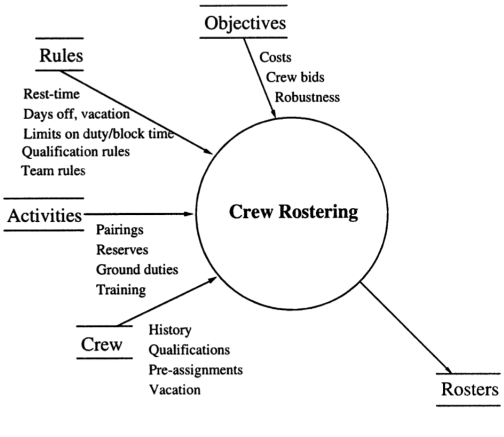

**Figure 2-1:** Schematic of the overarching crew rostering problem (Kohl and Karisch, 2004, p. 228)

As indicated, the commercial Constraint Programming (CP) model of Kohl and Karisch (2004) takes into account crew training activities, crew (and instructor) qualifications and associated rules. They describe in detail some of constraints encountered when also rostering trainees and instructors. Examples are to force an instructor and trainee(s) on the same pairing containing the training activity and the allowance of various crew compositions for simulator training. On the one hand, having to roster training activities increases the amount of scheduling options. On the other hand, it also limits the amount of options as some crew is forced to a specific activity. Kohl and Karisch (2004) explain modelling techniques for airline crew rostering, but do not report on the performance of their commercial software. However, they do stress that airlines often design very complex, multiple-day, multiple-requirement training programs that cannot feasibly be scheduled, even by their commercial software. This rules out the use of nonstandard training programs.

Kohl and Karisch (2004) describe just one implementation of crew rostering software while multiple approaches exist that take crew training into account differently. Airlines in the U.S. generally apply so-called bid-line approaches to the crew scheduling problem. Anonymous lines of work are created and then assigned to individual crew members. Conflicts due to personal training activities cause the trips reopen for scheduling according to Sohoni et al. (2004). In other words, training activities are prioritized over flights. In the European approach, known as preferential bidding, training activities are pre-assigned. Maenhout and Vanhoucke (2010b) describe such a personalized rostering problem. They schedule flights and other activities around pre-assigned (simulator) training activities. The model is claimed to take into account instructor qualifications, although this is not explicitly apparent from their model description. Maenhout and Vanhoucke (2010b) generate a crew roster using a fast hybrid scatter search heuristic. Obtaining a solution for a real-sized problem took approximately two minutes.

Summarizing, some approaches explicitly assign (simulator) training (Kohl and Karisch, 2004) whilst others take it as fixed, pre-assigned activities (Maenhout and Vanhoucke, 2010b). Emphasized in all approaches to overarching crew rostering problem is that assignments to flights are scheduled around prioritized training events. From the transportation crew rostering domain, it can be concluded that pre-assigning training activities is the more popular and faster approach (Kasirzadeh et al. (2017), Maenhout and Vanhoucke (2010a), Salazar-Gonzáles (2015) and Xie and Suhl (2015)). Still, all models focus on deterministic conditions, although accounting for disruptions was identified as future research already by Kohl and Karisch (2004) among others.

## 2-2 Crew Disruption Management in Airline Crew Training

Literature often neglects the stochastic operating environment of airlines because this is a field of research on its own. This section focuses on explaining and listing the current understanding, advances and shortcomings of so-called crew disruption management. A distinction is made in timing of crew disruption management. Early on in the rostering phase, airlines account for disruptions proactively. This so-called robustness (Clausen et al., 2010) is described in subsection 2-2-1. Close(r) to the day of operation airlines react to disruptions by recovering the schedule for the purpose of feasibility, as treated in subsection 2-2-2.

### 2-2-1 Crew Training Schedule Robustness

Clausen et al. (2010) define a robust schedule as one that includes a level of insensitivity to disruptions. They also use the term ’absorbing robustness’ while Dück et al. (2012) and Kohl et al. (2007) name it ’schedule stability’. Dück et al. (2012) further distinguished between robustness indicators and robustness measures as two independent concepts. The former is part of the objective function and represents a particular property of the schedule. The latter is a way of quantifying robustness via the simulation part of the model. The concept of robustness remains unaffected, so both are grouped by the term 'robustness indicator' throughout this thesis. Dück et al. (2012) listed the following robustness indicators applied in literature to make an airlines' (flight) schedule robust:

- **Buffers:** Add extra time (i.e. slack) to aircraft and crew connections to minimize the knock-on effects of delays.
- **Swap opportunities:** Enable swap opportunities for crew and/or aircraft assignments by synchronizing resources. Small changes can be applied to the roster to account for actual operational performance.
- **Simplified schedules:** Match aircraft and crew to minimize knock-on effects and pairings only consist of 'out-and-back' legs to eliminate knock-on effects of any cancellation.
- **Reserves:** Assign aircraft and crew reserves in advance to replace a disrupted resource.

All these indicators bring about increased cost with respect to the optimal schedule generated for a deterministic environment (Ehrgott and Ryan, 2002). However, Dück et al. (2012) stress that the efficiency varies. They define efficiency as the amount of robustness that can be gained per unit of cost. As a result, the objective is often to optimize a weighted combination of cost and robustness. Robustness indicators are frequently quantified by propagated delay minutes (Weide et al., 2010), buffer time (Ehrgott and Ryan, 2002), counted swap opportunities (Dück et al., 2012) or the number of reserve pairings needed to cover a specified amount of disrupted flights (Sohoni et al., 2004). The individual robustness indicators are treated in more detail below.

Starting, Ehrgott and Ryan (2002) added buffers to ground time by using the expected delay based on historical data of Air New Zealand. Generation of multiple bi-daily schedules with different weights for cost and robustness showed that robustness can be achieved efficiently. Schaefer et al. (2005) applied a similar approach, but tried to find a solution that better fits with the stochastic nature operational cost. They penalized pairings attributes that result in poor operational performance. A simulation method was used to evaluate the method's performance with random disruptions occurring throughout a single day. As a critical note, Schaefer et al. (2005) assumed that flight delays cannot be propagated, cancellation is not allowed and resources are completely independent. As a result, the value of their model is purely theoretical. Unlike the model of Schaefer et al. (2005), the Stochastic Integer Programming (SIP) model with recourse presented by Yen and Birge (2006) does account for interaction effects between aircraft and crew schedules. Their model compared weekly pairings in which crew switch aircraft and those in which crew and aircraft are matched based on simulated cost.

Shebalov and Klabjan (2006) introduced a method to optimize a combination of cost and the number of swap opportunities to increase schedule flexibility. The swap opportunities, which are called move-up crews, are computed upfront for each crew base, flight leg and crew member. The two-stage LP model is then tuned using two parameters: (1) to indicate the percentage of cost that is deemed acceptable with respect to the minimum cost solution, and (2) to indicate the maximum amount of swap opportunities to consider per flight. The first parameter is user-specified and the latter turns out to be insensitive. Ionescu and Kliewer (2011) presented a stochastic version of the same model. They considered swap opportunities based on their likelihood of being used, which is derived via scenario-based simulation. This approach only adds cost-effective swap opportunities without having to calibrate weight factors in the objective function (Ionescu and Kliewer, 2011), but it requires more computational effort.

Instead of exploiting swap opportunities, Weide et al. (2010) aim for a simplified schedule by matching aircraft and crew schedules aimed at minimizing delay propagation. They implicitly integrated the aircraft routing and crew pairing problem via the objective function and solved it iteratively. Dunbar et al. (2012) advanced this model by analyzing delay propagation between crew and aircraft more accurately. Historical data was used to create a dynamic penalty system for delay propagation. Dunbar et al. (2012) directly compared both methods on twelve randomly generated scenarios with limited amount of aircraft and crew at a single day of operation. The solution improved by eight percent on average when using dynamic data, but required longer computational time. Dück et al. (2012) applied the decomposition technique of Weide et al. (2010) to a stochastic model integrating crew pairing and aircraft routing. The model only considered delay propagation on a single crew pairing with several aircraft changes and the other way around. Nevertheless, it also converges to a solution in which crew and aircraft are matched as much as possible. Both the deterministic model and stochastic model show similar solution quality.

The last robustness indicator to review is the use of reserve crew. Sohoni et al. (2004) add reserves as part of their integrated manpower planning model. They proposed a two-stage Stochastic Integer Programming (SIP) model minimizing the number of reserve patterns needed to cover a specified percentage of 'open time' trips as possible. Open time is defined as the collection of trips that cannot be operated by the initially assigned crew member due to vacation, training activities or irregular operations. In the first phase of the model, they estimated reserve demand with a simulation based approach as presented by Rosenberger et al. (2000, 2002). The second phase consists of solving a set-covering model to generate a minimum set of reserve duty periods under the assumption that a reserve pattern is between three and five days long. A set-partitioning model, presented in Sohoni et al. (2006), is then applied to generate the actual optimal reserve patterns. The multiple-stage, scenario-based model shows solution times up to one hour for monthly reserve crew schedules at Delta Air Lines. Less computationally intensive reserve crew scheduling models are presented by Bayliss et al. (2012). They compared several solution techniques such as Dynamic Programming (DP) and heuristics to quickly generate a set of reserve pairings based on crew absence probabilities. Other disruptive effects are neglected and so are all recovery actions other than using reserves. Bayliss et al. (2013) presented a simulation evaluation approach to the same problem instead and also accounted for swap opportunities. They see swap opportunities as a way to reduce the need for reserve crew and obtain a more efficient solution. Unfortunately, they do not provide further details to their simulation evaluation method needed to understand underlying assumptions and effectiveness. Finally, Bayliss (2016) integrated the same simulation evaluation model with a Mixed Integer Linear Programming (MILP) model that is used to generate a set of reserve pairings. Thus, the MILP model generated a reserve schedule and simultaneously determined a policy of when to use reserve crews. The probabilistic model outperformed the MILP model in almost every scenario because it more accurately combines the numerous disruptions scenarios.

Summarizing, all literature presented incorporated robustness indicators to deal with flight delays. As far as other disruptions concerned, only Bayliss et al. (2012) considered crew illness as a source of disruption. Sohoni et al. (2004) are one of the few to explicitly mentioned vacation and training activities as cause of disruptions, grouped under 'open trips' in their research. However, it is important to note that their model was applied to the bid-line system that is often used by U.S. airlines. Training is put into an existing roster, meaning that overlapping flights will be dropped from the personalized roster. An extensive pool of reserves is then needed to cover all these open flights. Sohoni et al. (2004) thus see training as a disruptive effect, but they still assume that crew training is deterministic and undisrupted. A stochastic environment for training in particular is thus not accounted for. Other literature only mentions the difficulty of quantifying and proving the value of robustness (Clausen et al., 2010). The added cost of robust schedules must yield a cheaper schedule when considering stochastic effects in the operating environment, which is not trivial.

### 2-2-2 Crew Training Schedule Recovery

Robust cockpit crew training schedules have not been researched before. But what has been studied in the field of training schedule recovery? Airline crew training schedule recovery is actually a sub-field of crew recovery, so this section starts with treating the overarching domain.

Whenever a disruption occurs, aircraft, crew and passengers are re-planned sequentially due to the complex interdependencies (Medard and Sawhney, 2007). The aim of schedule recovery, also termed schedule flexibility (Ionescu and Kliewer, 2011), is often to return to the original schedule within a predefined period of time known as the recovery horizon (Kohl et al., 2007). Clausen et al. (2010) stress that requirements on solution quality are subordinate to restrictions posed on computational time due to the urgent nature of schedule recovery. The objective is to restore schedule feasibility. According to Nissen and Haase (2006), schedule recovery is concerned with deciding on the length of the recovery window. The shorter the recovery horizon, the faster the solution time and the worse the solution quality will be due to the limited amount of options considered. A link with cost is often established in the recovery process too. However, cost minimization is only suitable for U.S. airlines that compensate crew members for actual duty times. On the contrary, Nissen and Haase (2006) described that European airlines use a guaranteed pay structure and, as a result, incur only minor cost changes in the recovery phase. Opting for a minimization of schedule changes is thus more efficient for European airlines.

One of the early approaches to the recovery problem was to eliminate part of the complexity. Lettovský et al. (2000) assumed that the reference schedule is optimal in terms of cost and solved only for disrupted flights. Their model minimized the increase in cost by flight cancellation or swapping crew to newly generated pairings. Yu et al. (2003) proposed a similar approach, but emphasize that recovery solutions need to be generated in real-time. Their model is able to generate partial solutions that cover the urgent disrupted flights with higher priority first. This comes at the cost of implementing suboptimal solutions because the full set of flights affected by the disruption is not considered.

Although implementing suboptimal solutions is often of a lesser concern for airlines, Abdelghany et al. (2004a) try to improve on the method of sequentially generating partial, real-time solutions. They projected disruptions into the future to get insight in the propagation of delays into the network, but use a deterministic scheme to do so, as described in Abdelghany et al. (2004b). Next, a rolling-horizon approach was employed to chronologically solve the expected disruptions. They even automated the aforementioned process by considering swapping crew and using reserves. Flights that cannot be covered (cost-effectively) are reported open. A controller can then try to find solutions for these open flights by employing methods, such as finding volunteers, that the automated optimization model cannot. Abdelghany et al. (2008) integrated aircraft and cabin crew resources into the existing model to minimize discrepancies between decisions regarding a single resource. Medard and Sawhney (2007) similarly focused on proactive, integrated recovery by applying a rolling horizon technique. Where Abdelghany et al. (2004b) did not consider particular reasons behind disruptions, Medard and Sawhney (2007) explicitly included multiple sources of disruptions into the model. These sources are flight delays, flight cancellations, crew illness and aircraft roster changes.

The aforementioned literature implemented and proposed means of improving solution quality, but Clausen et al. (2010) underscored the value of a fast and conflict-free recovery solution. The latter can be achieved via integrated models, of which one is presented by Kohl et al. (2007). They worked on a commercial decision support system that integrated recovery for aircraft, crew and passengers. Still, fully integrated approaches proved too complex to solve in real-time. As an alternative approach, they presented dedicated solvers for each resource and integrate all in a next layer. The model iterates over any conflict between the partial solutions after integration. Unfortunately, no details are provided on the implementation of the solution methodology. On the contrary, Petersen et al. (2012) are one of the first to provide an extensive description of their solution methodology. They presented a similar fully integrate recovery model, but the size and complexity still asked for a problem size reduction using heuristics. They applied benders decomposition in conjunction with column generation to decompose the problem into computationally tractable parts. At the cost of solution quality, their procedure produced good quality solutions within 30 minutes of run-time for large problem instances. While solution methodologies were aimed at problem simplification, Maher (2015, 2016) was one of the first to find the exact solution of a fully integrated recovery model within reasonable time. A column-and-row generation approach solved favourable disruption scenarios with a maximum computational time of five minutes. The run time for less favourable disruptions scenarios rapidly grew to twenty to 45 minutes.

In summary, researchers have made progress in solving integrated recovery models, but five minutes of run time, as presented by Maher (2016), is still not near what is required by airlines. Instead, opting for simplified problems and sequential solution methodologies (Abdelghany et al. (2004a) and Kohl et al. (2007)) seem to be the proven method. However, Abdelghany et al. (2004a) underscored the stochastic nature of disruptions and the associated mismatch with the deterministic models presented. To others, the stochastic environment is reason why recovery cannot be solved without intervention of crew controllers. According to Kohl et al. (2007), crew controllers can suggest recovery strategies, such as mutually agreed voluntary overtime, that simply cannot be obtained with automated recovery models. But still, Abdelghany et al. (2004a) proved the value of automated recovery models in conjunction with human controllers. Such automated recovery decision systems can be of help when explicitly tailored towards any robust scheduling model.

## 2-3 Synthesis of the Research Gap in Airline Crew Training

Belobaba et al. (2015) elaborated on three widely recognized areas of research within the airline scheduling domain: (1) integrated scheduling, (2) robust crew scheduling and (3) crew schedule recovery. All directions will be deliberated on from a crew training point of view in this section to define the research gap, research question and research objective.

First, airline crew training is mentioned as an important interface of crew scheduling and crew rostering by, among others, Maenhout and Vanhoucke (2010b) and Salazar-Gonzáles (2015). Most airlines prioritize crew training over assigning regular flights because resources are more restricted and are deployed from the same, overarching pool of flight crew. Nevertheless, all literature assumes deterministic, pre-assigned training activities. In other words, they do not solve an integrated training and flight scheduling or rostering model. This can be explained by the little amount of previous work dedicated towards airline crew training scheduling. Yu et al. (2004) and Sohoni et al. (2003) presented models to schedule conversion training and recurrent training respectively. Later on, Holm (2008) integrated both types of training into a single, large Linear Programming model. The high run times do require a faster solution method such as the heuristic presented by Qi et al. (2004). Another possibility is to integrate the model with that presented by Xu et al. (2006), who scheduled flight instructors to the training events. Concluding, there is limited research into integrated approaches to airline crew training scheduling.

Second, the amount of research devoted to crew disruption management is indicative of its importance and the high cost associated with it (Kohl et al., 2007). Inter-dependencies between the flight schedule and crew training schedule, in the form of crew availability, make the schedules vulnerable. Adding robustness to training schedules could potentially improve the overall performance. The word potentially is added here because no previous work explicitly researched robust airline crew training scheduling. Numerous examples can be given of adding robustness to an airlines' flight schedule (Bayliss et al. (2012, 2013, 2017), Ehrgott and Ryan (2002) and Shebalov and Klabjan (2006)), but this could only serve as inspiration when considering robustness of the crew training schedule. Sohoni et al. (2004) are one of the few to explicitly mention training as cause of disruptions. However, their model inputs a known training schedule into a personalized flight duty roster. Overlapping flight duties are then removed and covered by reserve crew. Concluding, they do not account for disruptions in the training schedule itself, leaving a research gap.

Robustness is closely linked to schedule recovery as the former eases the latter. When implementing a robust schedule, the recovery process can be simplified to an extent in which a (partly) automated approach can be employed (Abdelghany et al., 2004a). This is an important attribute that can be used to quantify robustness, which is a non-trivial task according to Clausen et al. (2010). Nevertheless, the link with airline crew training is again neglected in literature. At most, Abdelghany et al. (2004a) mentioned the requirements for crew recovery plan, including crew training activities. On the other hand, Medard and Sawhney (2007) outlined how to handle pre-assigned (training) activities in a duty-period network-based recovery plan as part of restricting the problem size. Still, they do not account for disruptions with respect to the pre-assigned training activity or instructor itself. They implicitly assumed that training activities, assigned trainee and assigned instructor are all fixed (in time). These assumptions can be challenged as instructors with similar qualifications could easily be substituted rather than kept fixed. Rescheduling could also be a possibility. In short, including such options upfront (i.e. by means of a robust crew training schedule) would increase recovery flexibility and ease.

In summary, there is a lack of proactive consideration of disruptions other than flight delays. Namely, Sohoni et al. (2004) identified a large interface between airline crew training activities and schedule robustness. Likewise, Abdelghany et al. (2004a) mentioned training as a requirement that should be satisfied in a schedule recovery plan. Although identified as part of the problem, none of the researchers applied their models to solve for robustness of crew training. Concluding, the research gap lies in the combination of crew training scheduling, robustness and recovery, where the latter is needed to quantify the effectiveness of robustness. The research gap is captured in the following research question:

*What robustness measures can be taken to proactively minimize the impact of disruptions related to an airline cockpit crew training schedule considering relevant rules and regulations?*

The words 'related to' can be interpreted in two ways: (1) disruptive effects of training activities, and (2) the impact of disruptions propagating to training activities. This thesis focuses on the former category as the latter category requires modelling the wider crew scheduling domain. The answer to the research question is required to attain the research objective:

*The research objective is to make recommendations on increasing proactive robustness with respect to crew training schedule by identification and evaluation of proactive robustness measures and solution methods that utilize historical training disruption data.*

As seen throughout chapter 2, literature dedicated to proactive, robust crew training scheduling is limited, if not non-existent. Research into other, related fields is used to find interfaces and draw inspiration from to help fill the research gap. This also helps to identify a broader application of the research undertaken. Examples of this are the field of airline cabin crew training (Bijvank et al., 2007) and the transportation domains such as rail (Cacchiani et al., 2014) and bus (Xie and Suhl, 2015). Other domains that explicitly account for crew training, and that are well-researched, are that of nurse scheduling (Maenhout and Vanhoucke, 2010a) and personnel scheduling (Van Den Bergh et al., 2013). These domains will be part of the supplementary literature review in chapter 3.

Chapter 3

# Supplementary Literature Review

The supplementary literature review focuses on fast and proven solution approaches to fill the knowledge gap on robust airline crew training scheduling. Adding robustness to the scheduling methods is an integral part of the solution approach, as described in section 3-1. As a robust schedule eases the process of restoring feasibility of disrupted schedules, recovery and evaluation of schedule performance come into play as part of proving schedule robustness. The state-of-the-art of schedule recovery and schedule evaluation is therefore reviewed in section 3-2. Combining methodologies from both parts thus leads to a complete problem approach, as will be explained in section 3-3. The synthesis of models and methodologies will be addressed with respect to the research gap of robust airline crew training scheduling and the foreseen contribution to the body of knowledge.

## 3-1 State-of-the-Art in Robust Scheduling

Supplementary literature on the first part of the research gap, robust scheduling that is, is treated first. Inspiration is drawn from existing work to comment on the (potential) application of robust scheduling to the field of airline crew training in subsection 3-1-1. This allows to list promising robustness indicators and learn from research into their effectiveness. The issue of quantification of robustness, as identified by Clausen et al. (2010), is addressed in subsection 3-1-2. Here, an overview of the objectives encountered in research will be provided. Not only the aim of such models in wording is included, but also promising mathematical implementations thereof. However, an objective alone is not enough to achieve robustness. Proven models for robust scheduling will be analyzed in subsection 3-1-3. The chapter will be concluded with an overview of- and reflection upon - the identified methodologies in subsection 3-1-4.

### 3-1-1 Robustness Indicators and its (Potential) Application to Airline Crew Training

All robustness indicators treated in subsection 2-2-1 are applied to disrupted flight schedules. The modifications needed for applicability of buffers, swap opportunities and reserves to training schedules are described next respectively. The notion of simplified schedules is merged with swap opportunities as both suggest a similarly structured crew training schedule.

### Buffers

Recurrent training activities generally have a 'grace period' of three months, indicating the period in which the session can be scheduled to take place without impacting the due date for the next year. The due date is defined as the date until which a legal qualification is valid. Not completing a (re-)check on this qualification means that the pilot license validity is lost and the pilot is unable to fly. Thus, timing of recurrent training is actually captured in a trade-off. Notwithstanding the desired spreading across the semester, scheduling training close to the due date allows for a predictable, theoretically cost-effective schedule while lacking robustness. Robustness could be achieved by scheduling training with some due date buffer. However, note that scheduling recurrent training too early also impacts the due date for next year. Namely, when scheduling the recurrent training activity before the grace period, the due date will be set to the last day of that month in the next year. Effectively meaning that the due date will be less than one year ahead. Sohoni et al. (2011) solved a similar trade-off problem of which the approach could be applied to crew training timing as well. They presented a stochastic model to determine optimal departure- and arrival times based on historic (disruption) data. To reduce problem size, they only considered some pre-defined allowable time frame.

### Swap Opportunities

Where buffers affect production, swapping is considered to be a cost-neutral action to limit the impact of disruptions (Ionescu and Kliewer, 2011). Although beneficial from the robustness point of view, it achieves the opposite of the objective of crew schedule recovery as explained in subsection 2-2-2. It increases the amount of crew members with affected schedules if swapping is applied. This could be important for a flight schedule in which crew members have expressed bids and preferences, but to a lesser extent in a crew training schedule. Instructors and trainees are primarily assigned based on their qualifications and need for training respectively. Only timing preferences may apply, but these can be disregarded if absolutely necessary. On top, changing multiple schedules raises less resistance as all (simulator training) duties start and end at the home base at the same day.

Synchronizing schedules of several instructors and trainees thus allows for quick(er) recovery decisions and also to a more effective use of the (flight) schedule reserves. Also, swap opportunities can be used to maximize the accumulated priority level over all non-disrupted training activities, and hence minimize impact of disruptions. Sohoni et al. (2003) already provided an example of varying priority levels based on due dates of recurrent training, but priority levels could also depend on the type of training, length of the training program or expected shortages in available crew per rank or qualification. Swapping as a strategy is expected to be most effective when having limited amount of instructors and training devices.

Ingels and Maenhout (2017) presented a general, cost-minimization Linear Programming (LP) model that distinguished in various substitution possibilities that could be applied to crew training as well. First, they considered swap opportunities on individual basis (e.g. instructor or trainee) and group basis (e.g. instructor qualification). Then, they made a distinction between three types of reassignments: (1) reassignment to the same shift, but with a different skill requirement, (2) reassignment of shifts with the same skill requirement, and (3) an employee on a day-off is reassigned to take a shift. The first type of substitution could be applied to add flexibility by scheduling instructors with multiple qualifications. The second case is trivial whilst the third describes an option to take back 'loss days' incurred by scheduling inefficiencies or assigning crew members based on mutual agreement.

### Reserves

A special case of swap opportunities as explained by Ingels and Maenhout (2017) above is that of reserves. Here the swaps come from resources dedicated to passive duties. Looking into this research, Bayliss et al. (2017) pose no explicit requirements on the qualification or training need of the reserve crew member when optimizing for reserve pairings. Training disruptions are thus not explicitly accounted for, despite using reserves for training purposes in practice as well. Proactively accounting for disruptions in the training schedule and preparing cost-effective training recovery actions could add robustness to the training schedule. Another option is to set up dedicated training reserves, but as instructor reserves are interchangeable to cover flights as well, optimizing the skill mix in existing reserve pattern assignment is likely to be more efficient due to an economy of scale. The following reserves categorization, which may or may not be practically feasible, can be made:

- **Trainee reserves:** A set of trainees that have a similar due date / planning for the same training activity. If a trainee reserve is not available, the sessions need to be rescheduled or continue with an help-out instructor or pilot. Resources are being wasted as the help-out is not allowed to do the training activity for the records.
- **Instructor reserves:** A set of instructors that are qualified to provide the scheduled training activity. Having an instructor on reserve that is over-qualified potentially allows for swapping training activities as well.
- **Training device reserve capacity:** Reserve capacity that could be allocated towards requalification training sessions or used for rescheduling purposes to minimize the impact of disrupted sessions when capacity is limited.

Airline crew scheduling problems can serve as benchmark for a novel application to crew training scheduling. An example is the work of Sohoni et al. (2004), who optimized for reserve patterns based on randomly applied disruption scenarios under the assumption that all input is readily available. The input consists of: (1) a vector specifying the daily minimum number of off-duty reserves, (2) the minimum number of reserve patterns per type defined in on- and off duty days, and (3) a month-long vector specifying reserve numbers needed per day. The solution quality directly depends on the specification of this input, which is non-trivial for a novel domain. Reserve patterns are non-existent for training reserves and the minimum number of reserves can only be estimated through historical data or trial and error. The latter is complicated by any interface with flight schedule reserves.

A starting point to value such a novel application is then to look at general crew scheduling research. Ingels and Maenhout (2015) proposed and compared several strategies for generating reserve duties within constraints of their Integer Programming (IP) shift rostering model. These strategies enforced a minimum reserve staffing requirement, time-related constraints or a combination of the two. For each, the actual positioning and / or timing of reserve duties was based on simple rules such as: (1) levelling reserve duties over the time-window, (2) covering a fixed ratio of the staffing requirements by reserve duties, or (3) distributing reserve duties over time based on a weighting function proportional to staffing requirements. Ingels and Maenhout (2015) showed that such simple methods can be employed as a first means of generated reserve pairings for one of the aforementioned training resources. These methods also require little user-specified input, albeit more complex methods are likely to yield higher solution quality.

## 3-1-2 Objectives of Robust Scheduling

A robust solution is only obtained when being explicitly connected to the model's objective. The objective of airline crew training scheduling and associated robustness indicators need to be aligned. This means that they must be defined in the same units, or weighed in such a way that a desirable solution is obtained. The objective variants and associated fitness per type of model for both crew scheduling and robustness are reviewed next.

### Objectives for Airline Crew Training Scheduling

Kohl and Karisch (2004) state that training rostering problems are aimed at optimizing a combination of cost, crew satisfaction and robustness, but the research gap already contradicted this. One of the crew training scheduling objectives encountered in literature is the implementation of Yu et al. (2004). They minimized the number of training days in their training scheduling problem where they determined start- and end dates of all training activities. Subsequently, they solve the training assignment model in which they optimize for cost. Only in wording they explain to have made the cost dependent on preferred training devices and periods. Sohoni et al. (2003) provided more mathematical details for a similar cost minimization objective as in Equation 3-1.

$$A_{ij} = \rho_1 \cdot \frac{L_i}{T_i} + \rho_2 \cdot F_{ip} + \rho_3 \cdot D_{ij} \tag{3-1}$$

The user can specify weight factors $\rho_1$, $\rho_2$ and $\rho_3$ for each of the parts individually, and thus also obtain multiple solutions. The first part proportionally penalizes schedule $i$ for the length of the training activity $L_i$ with respect to the shortest possible schedule for crew member $j$ indicated by $T_j$. The second part penalizes the need for help-out instructors $F_{ip}$ in schedule $i$ and for position $p$. The last part is used to penalize for interfering activities in schedule $i$ for crew member $j$, implemented via parameter $D_{i,j}$. The latter is a disguised way of prioritizing crew members that have a less obstructed schedule, but this is only effective when using a bidline crew scheduling approach. On top, Sohoni et al. (2003) include generic cost for leaving a trainee and trip unassigned, but this is excluded from Equation 3-1.

Instead of a implementing a cost function that can be explicitly tweaked by the user, Holm (2008) opted for cost levels representing reality. Such an approach requires accurate specification of cost to prevent the user from impacting the solution quality after all. But if done right, the solution is directly comparable to current airline operations. Looking at the implementation of her pilot training- and transitioning model, accurately defining the cost is non-trivial. She has defined dynamic cost of transitioning an individual pilot and dynamic cost of having a shortage of pilots on each position. These cost parameters directly impact the trade-off to transition crew members and thus also impacts the training schedule. Assuming that these transitions are known, the remaining part of her model minimizes the schedule cost for all types of training via Equation 3-2.

$$min \sum_{k \in K} \sum_{t=1}^{N} \sum_{p \in \left\{CP, FO\right\}} (cc_k \cdot a_{kt} + cr_k \cdot x_{kt}^{p}) + \sum_{k \in K} \sum_{t=1}^{N} c\beta_k \cdot \beta_{kt} \tag{3-2}$$

The first part of Equation 3-2 iterates over all training activities $k$ for all time instances $t$ until time $N$ and all positions $p$. Cost are incurred per training activity $cc_k$ for each course $k$ scheduled at time $t$ indicated by $a_{kt}$. Cost are also incurred per per trainee of a certain position assigned to a specific course at a given time $cr_k$ via decision variable $x_{kt}^{p}$. The second part of Equation 3-2 determines for every training activity and time instance if a nonstandard crew is scheduled. A nonstandard crew is one other than a captain and first officer together. This incurs extra cost via the need for additional simulator capacity or additional instructors. Decision variable $\beta_{kt}$ specifies this need and adds an additional cost factor $c\beta_k$ that is dependent on the course.

Although the model could be extended, it does not account for instructor scheduling yet. Xu et al. (2006) did consider instructor scheduling and optimized a weighted combination of cost, workload balance and instructor preference violation. Such an objective function tends towards the social aspects of the problem (Gamache and Soumis, 1998), which are important in reality, but might be too little added value or be too restrictive in terms of run time for research purposes.

### Objectives for Robustness Indicators

The objectives of the training scheduling model should be appended with one that ensures some level of robustness. Dück et al. (2012) stated that a common approach is to optimize for a combination of cost and robustness. Several approaches have been applied in literature to merge two contradicting objectives into a single measure of the same units or order of magnitude. Ehrgott and Ryan (2002) penalized pairings with certain attributes that are considered to be 'non-robust' via weighting factors. However, choosing the weighting factors is difficult and directly impacts solution quality. Another technique is to convert all but one of the objective measures to constraints with a specified upper bound of $\epsilon$ (Bertsimas and Sim, 2004). The $\epsilon$ then represents the additional cost (as a percentage of the optimal solution) that can be incurred to achieve robustness. Still, these approaches depend on the actual robustness indicator that is implemented, its predictability and efficiency. Dück et al. (2012) explained predictability as the correlation between the robustness indicator and its effectiveness in terms of disruption impact minimization. Efficiency is defined as the level of additional cost incurred for a given amount of robustness. These quality measures are used to evaluate robustness objectives for each of the different robustness indicators below.

Starting with robustness through buffers, Ehrgott and Ryan (2002) used historical data on expected delay minutes to define a linearly increasing penalty function. This function is minimized over all pairings. On top, the cost increase with respect to the optimal level is capped via the constraints. Instead of using deterministic cost, Schaefer et al. (2005) minimized expected cost obtained from running stochastic disturbance simulations. This method implicitly included a penalty system as disrupted flights encompass greater cost. Yen and Birge (2006) minimized a combination of deterministic cost and expected cost of reactionary delays obtained from a stochastic recourse model. Dück et al. (2012) did the same, but included weight factors based on the probability of the scenario rather than a chosen penalty factor. Adding buffers is a predictable means of adding robustness, but it is less efficient as it could lead to wastage of resources.

A more efficient, but less predictable, robustness indicator is found in swap opportunities. Shebalov and Klabjan (2006) simply counted swap opportunities for all pairings under consideration. They then optimized for a combination of cost and number of swap opportunities. Ionescu and Kliewer (2011) used the probability of using each available swap opportunity as a model extension. These probabilities, obtained from simulating delay propagation, are used as scaling factors for the bonus cost in their cost minimization problem. Ingels and Maenhout (2017) extended the notion of swap opportunities to multiple mechanisms of crew substitution, both on individual level as on a group perspective. The number of substitution possibilities, combined with a cost factor and cross-training shortage penalty / surplus bonus, is used in the objective function of a deterministic crew schedule. They compared the objective function based on simulation, which outputs expected cost and the expected number of substitutions used.

Reserve crew is less efficient than swap opportunities, but offers higher predictability. Especially for airlines with large size operations. Rosenberger et al. (2002) simulated such operations and proposed a simple form of a robustness indicator for usage of reserve crew. They counted the number of rule violations due to disruptions to estimate the number of used reserve crews. Instead of simulating, Bayliss et al. (2012) introduced an analytical model to minimize the expected level of disruptions given a set of reserve crews. They compared nine objective functions related to the probabilities of crew unavailability, the standard deviation, coefficient of variance thereof or a (weighted) combination. Minimizing the sum of crew unavailability probabilities turned out to yield the best solution in terms of reserve utilization rate and flight cancellation rate. In a later work, Bayliss et al. (2013) introduced a simulation evaluation approach to minimize a weighted total of expected crew delay measured after a set of reserve crew pairings was applied. Next to this, they calculated cancellation rate, reserve utilization rate, average crew delay, total crew delay and probability of having more than half-an-hour of delay. Bayliss et al. (2017) wanted to minimize the combined level of disruption of both delays and cancellations. They introduced a formula to convert delay and cancellation into a common measure. The user is able to specify a scaling factor between absorbing delays and cancelling flights. Note that the problem of reserve crew is actually a balance between reserve crew cost and cost for covering open trips with regular crew based on premium pay (Holm, 2008). A premium pay is added on top of the normal salary for crew that agreed to take on additional duties.

In short, every robustness indicator is implemented differently. Buffers are often measured in time, swap opportunities are counted and reserves are optimized for by (un)availability rate. This alone does not yield a robust solution. It is often directly combined with cost or via a cap on increased cost. These explicit methods work well with deterministic models, but using stochastic models allow for optimization of expected cost. The robustness indicators are then applied more effectively. Another important fact noted by Bayliss (2016) is that each measure (for an indicator) shows different predictability and efficiency discrepancies under different solution realizations. Nevertheless, swap opportunities are considered to be most efficient, whilst buffers and reserves are more predictable.

## 3-1-3 Scheduling Models and Methodologies

Robustness can be captured in model objectives, where in turn the model objectives are part of larger scheduling models. These models include problem specific information and constraints to obtain a realistic solution by employing algorithmic frameworks. The state-of-the-art on scheduling models and methods will be described and analyzed for airline crew training scheduling first. Due to the limited amount of available research, a distinction can only be made between exact solution methodologies (applied to LP models) and approximate solution techniques (heuristics). The models, their assumptions and performance are explained in order. A lot more research is available on robust scheduling. Researchers have applied, tested and compared more solution methodologies, challenged more assumptions and found more applications. Still, many robustness models are solved using exact approaches in the form of LP models. Yet, researchers started questioning the deterministic nature of these models and captured this in stochastic models, as will be explained later. Lastly, robust schedules can also be approximately obtained via heuristics for the sake of quick yet sufficiently accurate solutions. Each of the aforementioned models and approaches, their assumptions and performance are treated in more detail in order.

### Linear Exact Approaches for Training Scheduling

Yu et al. (2004) solved their conversion training scheduling model with a rolling-horizon approach. They assigned a set of training activities to a set of available training device periods while setting aside redundant capacity for recurrent training. It remains unclear if the solution for Continental Airlines was deemed feasible with this assumption. Details on computational time are also not provided and benefits are only established with respect to manual solutions. Conversely, Sohoni et al. (2003, 2006) optimized recurrent training only via a LP model. They constrained the number of assigned trainees based on position- and domicile to ensure sufficient capacity was left for regular flight operations. Sohoni et al. (2003) even used a post-optimization heuristic to increase simulator slot usage more effectively by rescheduling sessions with partial simulator demand. For realistic test cases of assigning 140 pilots to training, their LP model solved within 5 - 10 minutes. They also reported improvements in terms of dropped trips due to overlapping training of more than 15 percent compared to manual training schedules. However, when integrating scheduling of conversion- and recurrent training, computational time will sharply increase. Holm (2008) reported computational time upwards of 18 hours for her model tested with a real-case of SAS Scandinavian Airlines. The test case involved 1,884 recurrent courses and 69 conversion training events during one year. For smaller problem sizes, the run time is expected to still be high due to the large amount of scheduling options throughout a year.

### (Construction Based) Heuristics for Training Scheduling

Solution quality can be traded against a benefit in computational time. This requires other solution techniques such as (construction-based) heuristics. In 1997, Lufthansa Technical Training moved from manual training rostering to an algorithmic approach based on previous research into timetabling (Haase et al., 1999). They employed a construction method considering training courses one-by-one constructing a schedule from scratch. This greedy heuristic does not necessarily lead to the global optimum, but is a simple model that solved within six minutes for a test case assigning over 300 training courses.

Instead of a greedy algorithm, Qi et al. (2004) developed a rolling-horizon, construction-based training rostering model based on a branch-and-bound solution methodology. In each iteration, their method scheduled $h$ out of the $n$ considered courses. Both $h$ and $n$ are user-specified parameters $n$ is restricted to be larger than $h$. With the correct settings, a near-optimal schedule is achieved. Keeping the value of $h$ below five ensures that the solution time remains under one second. With increasing $h$, run time sharply increases to three minutes with $h = 7$ already. Note that Qi et al. (2004) tested the algorithm on a small problem considering 43 classes of varying length with a combined total of 230 trainees during a single year. As the research is relatively old, the computational performance should be put into perspective. Such construction-based heuristics mimic the manual process of constructing and changing schedules, which airlines do on a daily basis. However, it is difficult to capture this scheduling process in rules that could be employed by the heuristic that maintain a certain level of flexibility for future scheduling decisions and yield a feasible solution after all. On the other hand, making straight-forward decisions is what keeps construction-based scheduling heuristics fast as opposed to consideration of large amounts of irrelevant options as done by LP models. Albeit some challenges arise when constructing schedules instead of generating them, there is potential in terms of the fast solution times and good solution-quality.

The construction-based heuristic of Qi et al. (2004) can be applied in conjunction with any LP training scheduling model available with little alteration of the algorithm. The same applies to instructor scheduling. Xu et al. (2006) presented a set-partitioning of the instructor scheduling problem and solved it with a different kind of heuristic: a dynamic neighborhood based tabu search. Their algorithms solved realistic test cases involving a couple dozen instructors and hundreds of training events within five minutes of run time. The only problem is that sufficient instructors cannot be guaranteed. Integrating the instructor scheduling and training scheduling problems eliminates this issue.

### Linear Exact Approaches for Robust Scheduling

Ehrgott and Ryan (2002) formulated a LP model to select a set of pairings yielding minimum delay. They have added a soft constraint on the increase in cost due to added robustness. This constraint is soft in the sense that it can be violated by a margin under the assumption that the user is unable to establish a hard cap. This allowance of variance is stated to improve computational time required, although Ehrgott and Ryan (2002) did not quantify this.

Shebalov and Klabjan (2006) minimized cost of a crew schedule and maximized the number of swap opportunities in a second stage in their LP model. Both stages are linked via a constraint that caps the added cost of pairings containing swap opportunities. The potential swap opportunities to choose from per crew base, flight leg and crew member are pre-processed using a heuristic. The heuristic trades-off duty limitations, duration of remaining duties and more. Solutions with few move-up crews (i.e. one or two) covering many legs proved superior over many move-up crews covering few legs. However, the efficiency of using swap opportunities is not tested independently as Shebalov and Klabjan (2006) applied flight cancellation, use of reserve crew and crew deadheading as recovery actions next to swap opportunities. On the other hand, it does provide a good benchmark for potential benefits for realistic cases seen by airlines. Another option is to simply enforce swap opportunities via constraints in a single stage LP model. This was done by Ingels and Maenhout (2017).

Earlier, Ingels and Maenhout (2015) presented an IP reserve model for a general scheduling problem. The benefit of this model is that it can be used next to any scheduling model without the need for integration as opposed to their later LP model for swap opportunities. Though it does require adjustment to the schedule under consideration and specification of some user-specified parameters. The latter includes static- and / or dynamic reserve staffing requirements, which is a difficult task with a direct influence on the level of robustness. Ingels and Maenhout (2015) showed that posing dynamic requirements on reserve staffing levels outperform simpler approaches such as static requirements or assignment of left-over resources. The performance improved in terms of resource utilization to 61 percent compared to 53 percent of the second best approach. Also, cost predictability improved to a gap of only three percent with respect to planned cost as opposed to 11 percent for the second best reserve scheduling strategy. Unfortunately, Ingels and Maenhout (2015) did not provide details on their set of experiments for comparing reserve staffing strategies. Despite, their research showed that simple methods can be employed as a first, yet already accurate, means of generating reserve pairings for one of the aforementioned training resources.

### Stochastic Approaches for Robust Scheduling

Sohoni et al. (2011) optimized for on-time performance in a stochastic environment that simulates flight disturbances. They proposed a Stochastic Integer Programming (SIP) formulation with non-linear constraints. These constraints are converted to a set of linearized constraints. A cut generation technique is applied to solve the approximated model. With limited allowance in variability in start times, their model found optimal departure times within one hour for realistic problem sizes. An example was given with 1,500 flights, 85 airports and five aircraft types. They plotted multiple service levels versus the objective function value displaying operational profit.

Yen and Birge (2006) also implemented a SIP model, but evaluated interaction effects between aircraft and crew schedules. The cost of crew switching aircraft is estimated in randomly generated disruption scenarios and fed-back to a deterministic crew scheduling LP model for the next iteration. Yen and Birge (2006) do not describe how these updated cost estimates are treated across the iterations, although it can be deduced that they use average cost per pairing. The model is solved for numerous scenarios, but can be stopped at anytime to provide the best solution found up to that point. They comment on the results obtained with respect to a historical case from 1997 but only show the difference between the initial and final solutions, which are meaningless from a non-methodological perspective. They show that the solution rapidly converges to be within one percent after 25 iterations compared to the solution obtained with many more iterations.

### Heuristics for Robust Scheduling

Schaefer et al. (2005) tried to find a solution to the airline crew scheduling problem that fits the stochastic nature of the operational environment. They penalized operational cost, obtained via the disruption simulation environment of Rosenberger et al. (2000), based on pairing attributes that resulted in poor operational performance. They considered pairing attributes such as sit times between flights and elapsed duty time. They found non-continuous expected cost and approximated the scheduling problem solution using a local search heuristic to limit computational time. This insight is provided by earlier research, but they do not report on the improvement in run time. The conclusion is also limited to general statements on well-performing pairing attributes from the perspective of expected cost.

Not only stochastic models can be solved using heuristics, also nonlinear model formulations. Campbell (1999) showed this for a problem of maximizing cross-utilization. The cross-utilization capabilities are governed by a continuous variable ranging between 0 to 1 and enforced in the objective function by a quadratic function. Basically, their model ensured scheduling of workers that have qualifications for multiple duties. Campbell (1999) opted for a heuristic approach instead of a standard approach using Lagrangian relaxation combined with Dynamic Programming (DP). Their method converged to within 0.5 percent of the optimal solution in every test run, although they do not compare results with the DP algorithm.

Bayliss et al. (2012) also employed a DP algorithm, but did so to solve the reserve crew scheduling problem. They also solved the same problem with heuristics such as: (1) greedy algorithm; (2) construction heuristics; (3) local search; (4) tabu-search; (5) simulated annealing; (6) genetic algorithms, and (6) ant-colony optimization. Although their focus is to compare all these algorithms, they do not provide the exact algorithms or pseudo-code. Their comparison however, is extensive. They concluded that construction heuristics were fast and lead to good-quality solutions, while tabu-search came close to optimality in repeated simulation evaluation runs. As criticism, their model uses a single-valued reserve duty length. Selecting only reserve duties of equal length is not necessarily realistic. The DP algorithm solved a small test case of 25 departures, nine reserve crew and a fixed reserve pairings length of 3 departures for 2,000 simulation runs in less than 40 seconds. However, the fastest heuristic, that still provided a high-quality solution, was solved in less than 0.2 seconds. The heuristic solution techniques presented by Bayliss et al. (2012) are promising in terms of efficiency compared to the IP model presented by Ingels and Maenhout (2015). Both can easily be solved for an existing schedule, but the former is expected to be quicker, but will not solve to optimality. However, optimality is difficult to prove for stochastic problems.

Later on, Bayliss et al. (2017) again solved the same model using the same techniques, but for a larger and more realistic problem size on a computer that is more representative for current computational performance. They established a three-day test case comprising 243 flights, 148 crews and 37 aircraft. Rule-based methods, such as equal distribution of reserve crew, all solved within one-tenth of a second. These can be used when computational time is critical. Note that they did show lower reserve utilization rates than the method presented by Bayliss et al. (2012), which still solved in half a second. Bayliss et al. (2017) did present extensive pseudo-codes and model descriptions in this paper, making their model and method easy to recreate and implement. What remains is the drawback that the models presented by Bayliss et al. (2012, 2017) are static in terms of reserve duty length and disruption probabilities. Sohoni et al. (2006) did explicitly take into account reserve patterns of varying length, but solved it with a slower two-stage LP model. Despite their tests, the computational time cannot be valued and compared with the models presented by Bayliss et al. (2017). Namely, Sohoni et al. (2006) used their model to cover open time trips at an U.S. airline which results in larger problem sizes compared to the models tested for an European airline.

### 3-1-4 Synthesis of Robustness Indicators, Objectives and Models

In this subsection, literature on both training scheduling and robust scheduling has been reviewed. An overview of the objectives, models and solution techniques is presented in Table 3-1.

All training scheduling models presented optimize for cost in some form, making it the objective formulation of choice. However, there is no literature that combines this with some robustness indicator. Several methods have been identified in literature as how to extend a cost minimization objective to also produce a robust schedule. Options are to convert the robustness indicator into a common (cost) factor (Bayliss et al., 2017) or to cap the increase of cost with respect to a non-robust, minimum cost solution (Ehrgott and Ryan, 2002). Capping the allowable increase in cost is more natural and suited for obtaining realistic solutions when implementing expensive forms of robustness such as buffers and reserves. It is less important when accounting for more efficient swap opportunities. Shebalov and Klabjan (2006) incorporated a certain number of swap opportunities

**Table 3-1:** Overview of all models and methods on robustness

| Training scheduling objectives |
|---|
| • Training cost minimization<br>• Training cost minimization and workload balance / crew satisfaction<br>• Training cost minimization and robustness |
| Robust scheduling objectives and indicators |
| • Delay penalty<br>• Number of swap opportunities<br>• Reserve utilization versus unavailability |
| Training scheduling and rostering |
| • Sequential exact approach<br>• Integrated exact approach<br>• Integrated construction based heuristic |
| Robust training scheduling and rostering |
| • Linear exact approach<br>• Stochastic model<br>• Heuristics |

per day of the flight schedule via bonus cost. They used a simulation approach to scale the bonus cost based on likelihood of being used in the recovery process. A final distinction is found in the implementation of objectives. When using deterministic cost, the solution will naturally converge to the minimum cost solution in practice. Using a simulation approach to determine the expected cost steers towards a robust solution instead and is therefore more promising from an academic perspective. Especially when numerous options are available, one should only choose the most efficient buffers, swap opportunities or reserves.

The body of knowledge on training mainly consists of separate scheduling models for conversion training, recurrent training and instructor scheduling. These models could be solved sequentially, but this neglects the interdependence in terms of simulator capacity and instructor availability. The model of Holm (2008) already integrated recurrent training and conversion training scheduling and is the most advanced training scheduling model found in literature. When integrated with instructor scheduling it can establish fully feasible schedules, which seems achievable based on the model formulations of Holm (2008) and Xu et al. (2006). However, further integration of models requires a faster heuristic solution methodologies such as presented by Qi et al. (2004). Their algorithm takes minutes to solve a training schedule as opposed to 18 hours. As their heuristic is also build around a LP model solved using a branch-and-bound algorithm, the same construction-based heuristic is applicable to the model of Holm (2008) with little modifications needed. It must be noted that their heuristic selects subsets of training events for each iteration obtained via rules, which could be difficult to tune in such a way that a realistic schedule is constructed. The same algorithm could be employed for solving a robust training schedule. Researchers have solved robust schedules using heuristics before such as Schaefer et al. (2005) did for buffers and Shebalov and Klabjan (2006) did for swaps. Others have formulated LP models, or approximated stochastic variants thereof by LP models, that could be solved using branch-and-price as part of the model as well. Reserves can be solved by heuristics (Bayliss et al., 2012) or using exact approaches (Ingels and Maenhout, 2015) separately from the training scheduling model with the user only specifying minimum reserve levels or available reserve capacity. Still, all these promising methods need to be adjusted to the training scheduling problem and prove themselves.

## 3-2 State-of-the-Art in Evaluation and Recovery of Disrupted Schedules

This section is concerned with the analysis of supplementary literature covering the second part of the research gap: schedule recovery. An important reason for this lies in evaluation of robust schedules. Improved performance of a robust (training) schedule can only be validated by extensively testing the schedule in a stochastic environment. Methods and models are presented in literature will be treated in subsection 3-2-1. Recovery of disrupted schedules is an integral part in evaluation of robust schedules. Therefore, the state-of-the-art in schedule recovery is described in subsection 3-2-2. The section is ended with an enumeration of- and reflection upon - the identified evaluation and recovery methods and models in subsection 3-2-3.

### 3-2-1 Evaluation of Schedule Robustness

The predictability and efficiency can only be compared for airline crew training after evaluating the robustness of the schedule. Van Den Bergh et al. (2013) have identified common evaluation approaches of robust schedules in the personnel scheduling domain. Analytical evaluation models are used in some literature due to their straightforward nature. However, the majority of researchers use more complex discrete-event simulation based approaches as these well-connect to stochastic airline operations. Related to this is a promising approach originating from game theory known as Monte Carlo Tree Search (MCTS). Although this method has not been applied to airline schedules yet, it could be applied for the analysis of outcomes of disruptions and actions. Analytical models, discrete-event simulation and MCTS are described in order below.

#### Analytical Evaluation Models

Bijvank et al. (2007) applied a statistical model to determine the number of cabin crew reserves needed per time period to minimize delay propagation effects. They proposed three simple techniques to determine start time, length and the number of reserve blocks. The performance of these techniques was evaluated using an analytical procedure under the assumption that the disruptions are completely known in advance. They also disregard days-off in reserve blocks. A fixed disruption handling scheme was introduced to update probabilities of having reserves available. Reserve schedules were then evaluated in terms of (1) expected number of unused reserves, (2) expected number of propagated delays, and (3) expected number of unresolved disruptions.

Bayliss et al. (2012, 2013) also used an analytical scheme to evaluate the quality of a given cabin crew reserve schedule in combination with a vector of crew absence probabilities. The scheme iterated over all flights in the duty period of each reserve crew. The probability that a specific reserve crew is still available and the probability of that reserve crew member not being available are both processed and updated. They evaluated the solution quality based on the sum of crew unavailability probabilities, which turned out to be non-linear function due to overlapping reserve crews.

#### Discrete Event Simulation Evaluation Models

The evaluation model of Bayliss et al. (2012, 2013) is analytical in nature, but it is tested in a simulation evaluation approach. Instead of applying their model to a series of historical data, they generate random input disruption probabilities to extensively test their model in a multitude of repetitions. Still, the analytical evaluation models have in common that they are simplistic and involve a lot of assumptions. Simulation evaluation approaches can capture complex scenario's of airline operations, making them suitable for stochastic problems.

Both Rosenberger et al. (2000) and Clarke et al. (2007) described a framework for simulation evaluation methods known as SimAir and MEANS respectively. These simulation models are specifically developed to evaluate robustness of flight schedules and recovery policies in a stochastic operating environment. Yet, airline crew training schedules are not that different. The SimAir model of Rosenberger et al. (2000) consists of three modules: (1) an event generator, (2) a controller module, and (3) a recovery module. The event generator uses randomly sampled numbers from aggregate distributions to introduce delay. In this specific implementation, they add additional flight time, ground time and / or downtime due to unscheduled maintenance. The controller module is notified about every event and determines if and when to invoke the recovery module. Rosenberger et al. (2000) do not provide details about this, the controller module could decide to neglect disruptions if they are small or when all resources remain within duty limitations. If a disruption cannot be neglected, the recovery module is invoked to solve the disruption. Recovery is thus an essential part of the simulation evaluation framework, on which more details will be provided in subsection 3-2-2. This subsection continues with an analysis of literature on event generation and comparable controller module implementations.

Starting with event generation, Rosenberger et al. (2000) mention aggregate distributions as means to generate random disruptions, but lack details about this process. Fortunately, other literature does provide these details for implementations in domains of airline scheduling, transportation and manufacturing environments. Common approaches are to use theoretical probability distributions or synthetic ones. Historical data remains important throughout to tune parameters and validate the model.

Bijvank et al. (2007) discussed the problem of selecting the correct number of cabin crew reserves at KLM to prevent propagation of disruptions. Historical data was used to determine a probability for the primary disruptions. Under the assumption that all historical events are independent, they simply divided the event count by the total number of crew days or flights. These probabilities were then applied to a statistical model to determine the number of cabin crew reserves needed per time period. A binomial distribution, used to model a series of independent true or false experiments (Ross, 2010), was then approximated by a normal distribution and used to ensure 95 percent availability of reserve cabin crew. Next to Bijvank et al. (2007), many other researchers used historical data, but not all are clear on the details. Ehrgott and Ryan (2002) used 18 months of historical data to estimate the average flight delay. Similarly, Abdelghany et al. (2004b) estimated taxi times, flight times and aircraft service times. Due to the lack of detail in their description, the reliability of the data sets cannot be assessed.

Unreliable results are also a risk when little historical data is available. Bayliss (2016) used only one month of operational data to estimate crew unavailability. To extend the applicability of the model beyond one specific data set, they varied control parameters in a random schedule generator. This approach tested model sensitivity, but the variation in control parameter values remained unsubstantiated. Whenever there is a lack of data, parameters could also be assumed. Abdelghany et al. (2008) assumed parameters of a uniform distribution to generate delays and simultaneously stressed that the uniform distribution is invalid due to dynamic effects. In the same way, Campbell (2011) assumed a distribution and associated parameter for demand of workers and Easton (2014) assumed a distribution for the availability of employees. The aforementioned studies are suited for sensitivity analysis and model testing, but do not demonstrate the value of a model for a realistic cases.

A comparable approach is to use historical data to tune parameters of synthetic distributions, but this again requires a sufficiently large data set. Ingels and Maenhout (2017), for example, assumed a Bernoulli distribution for capacity uncertainty by true or false decisions (Ross, 2010). An employee is either available or absent. For the uncertainty in staffing demand, a Poisson distribution was used to model incoming work per unit of time. Ingels and Maenhout (2017) used data of an emergency department of an hospital presented by Ahmed and Alkhamis (2009) to determine the inter-arrival time needed for the Poisson distribution. Instead of using a static synthetic distribution, Barmby (2002) modelled employee unavailability as a function of the number of days that an employee has been absent already. The formula, that captures the notion of endured absenteeism, was tuned based on historical data of an undisclosed firm. Yet another approach was applied by Lapp et al. (2008). They filtered data and generated synthetic distributions related to specific data attributes. However, this requires a vast amount of detailed and categorized data.

Devore (1999) provided more details on possible methods to estimate parameters of synthetic distributions. The mean and standard deviation of a data set are often used and easy to calculate. Other of such parametric estimation methods are Maximum Likelihood Estimation (MLA) or Least Squares Regression (LSR). Parametric estimation assumes an underlying normal distribution, which requires a sufficiently large sample size to be valid. If no information is available from theory about the distribution, its form and function can be obtained via non-parametric estimation (Devore, 1999). These methods are similar to machine learning approaches. Sejdovic et al. (2016) applied unsupervised learning to cluster large historical data sets to analyze and detect patterns on disruption management in a manufacturing environment. In the bus transportation domain, Mendes-Moreira et al. (2015) used a vast data set on vehicle location to validate the coverage of bus schedules using clustering algorithms. The clustering algorithm analyzed time series data to find events in which headway deviations did lead to 'bus bunching' (i.e. buses being too close to each other). This data was then converted to a distribution for usage to reduce bus bunching. Potvin et al. (1992) on the other hand used a neural network for automated vehicle- and crew dispatching. They trained the algorithm on empirical data from expert dispatchers to mimic and eventually outperform the human decision process by evaluating the quality of each candidate vehicle. Lagerholm et al. (2000) applied a similar technique, but implemented the neural network in a feedback loop. By solving the airline crew scheduling multiple times, their neural network trained and updated the solution process based on this. The latter approach could also be applied to robust airline crew training scheduling in conjunction with simulation evaluation models to synthetically create a training data set. Robust training schedules could be rewarded based on performance in a random disruption scenario using machine learning algorithms. This learning effect could benefit convergence and takes away the need to tune parameters from the user.

With a known distribution and (estimated) parameter(s), discrete disruption events can be generated to represent reality. Subsequent handling of these disruptions must also be modelled in a realistic problem environment. Literature on evaluation models of random disruptions in the flight schedule can serve as inspiration for potential application to airline crew training scheduling. Examples of such elaborate disruption handling model implementations are given below.

To start with, Abdelghany et al. (2004b) presented a model to predict future delays under the initial assumption that no recovery actions were taken. They used a network representation to map connectivity of aircraft and crew to the various flights in the schedule. Each resource has an individual ready time due to operational constraints and crew legality rules that are enforced in the network by default. This is then used in a shortest path algorithm to determine the earliest possible start time of each flight or event. Afterwards, random delays were generated for the next twelve hours of operation. The delay propagation algorithm simply propagated these delays via deterministic event duration based on historical data and resource dependencies provided in the network. If the earliest possible start time of an event has changed beyond the departure time of a flight, a delay is propagated. Abdelghany et al. (2004b) solved the model every three minutes with the intention of aiding human controllers in their decision making. This means that the controller receives regular but rough updates on the delay scenario. Rough, in this case, means that stochastic and dynamic effects are neglected.

In contrast of applying the model to real disruptions, Ionescu and Kliewer (2011) and Yen and Birge (2006) apply randomly generated scenarios. They both present different simulation evaluation approaches as part of a stochastic recourse problem. Ionescu and Kliewer (2011) only used primary delay simulations to estimate the usage rates of swap opportunities and did not propagate these disruptions. The usage rates of swaps are then directly translated to scale cost factors of the deterministic part of the model. Instead, Yen and Birge (2006) solved a deterministic crew scheduling problem first, after which the recourse problem evaluates expected cost of one randomly generated disruption scenario. Constraints are then generated for the most expensive pairings in each iteration after which the deterministic model is resolved. If some specified upper bound is violated by the solution, the recourse model is solved again. Both models of Ionescu and Kliewer (2011) and Yen and Birge (2006) must be solved for many scenarios realizations making it computationally expensive (Yen and Birge, 2006).

In the personnel scheduling domain researchers also evaluate the effectiveness of robustness indicators via simulation. Ingels and Maenhout (2015) evaluated different reserve crew strategies based on discrete event simulation in conjunction with a recovery model. A deterministic scheduling and allocation model, minimizing cost and maximizing crew satisfaction, is solved using a branch-and-price algorithm applied to the nurse scheduling problem by Maenhout and Vanhoucke (2010a). The simulation approach then uses stochastic variables for demand uncertainty and capacity uncertainty based on probability distributions and associated parameter tuning. The simulation draws random realizations from these probability distributions to generate random daily perturbations.

Ingels and Maenhout (2017) used the solution methodology presented in Ingels and Maenhout (2015) to optimize for cost in combination with substitution possibilities. They have applied their model to a real-life test case in the nurse scheduling domain in where a portion of the are nurses are cross-trained (i.e. have several qualifications). Randomly generated disruptions were applied to evaluate and compare a baseline (i.e. non-robust) schedule and the robust schedule after recovery. They quantify robustness in terms of number of incorporated substitution possibilities and the fraction of used swap opportunities. The worst-case, best-case and average scenario results are compared. This approach is logical, but the worst and best performance only provide an estimate of lower and upper bounds. Having a stable scheduling model is more valuable than having one that performs good on average. Ingels and Maenhout (2017) reported computational times of approximately six minutes for small test cases of twenty staff members already. This partly has to do with the monthly time span for which the problem is considered. This time span vastly increases the amount of options that need to be checked by the Linear Programming (LP) model. The same applies to when increasing the number of staff members considered to a more realistic level for the airline crew training problem. It does provide an accurate estimate of run time for such optimization models as the research is relatively recent.

### Monte Carlo Tree Search as Simulation Evaluation Model

A different evaluation method that also uses discrete-event simulation is Monte Carlo Tree Search MCTS. MCTS has found its way from game theory to other applications such as scheduling. According to Browne et al. (2012), two fundamental concepts apply to MCTS: (1) outcomes of actions can be approximated using random simulation, and (2) these outcomes can be used to find a best-first strategy. According to them, MCTS could be applied to define optimal sequential decisions, thus also to recovery from disruptions in airline crew training. Another application is that of testing the performance of robustness indicators related to the airline crew training schedule. Building a tree allows for evaluation of a sequence of decision on schedule robustness in a stochastic environment. According to Browne et al. (2012), such sequential decision processes (i.e. Markov Decision Processes (MDP)) are then used to define a reward-maximizing policy over a set of states, set of actions, a transition model and some reward function. The transition function determines the probability of reaching a new state after applying a certain action.

According to Browne et al. (2012), MCTS is an expensive model to solve and hence includes a prespecified budget cap to which the models iteratively builds a search tree. This search tree consists of states with directed links to other derivative states. The links represent actions. Browne et al. (2012) described the pseudo-code for a general MCTS approach, which is visualized in Figure 3-1. The first step is to recursively select a child node that has unvisited children and is non-terminating. Then, the expansion phase adds the child nodes to expand the tree. Via simulation, the outcome is produced for the new node. Finally, the simulation result is updated and back-propagated through the tree to learn from the performance of the sequence of decisions. Because of this continuous updating, the algorithm can be stopped at any time and produce the most promising root decision to take. Browne et al. (2012) defined most promising as the combination reward and number of visits of the root child. Another benefit of MCTS is the tendency to grow asymmetrical trees, which means that focus is naturally shifted towards promising branches.

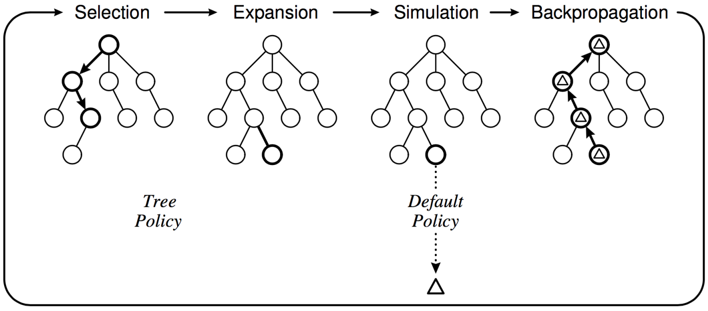

**Figure 3-1:** Iteration of the general Monte Carlo Tree Search Algorithm (Browne et al., 2012)

A widely used variant of the MCTS algorithms is known as Upper Confidence Bound on Trees (UCT). Browne et al. (2012) described that this algorithm is proven to converge and ensures that all children of a node are at least visited once. Walędzik et al. (2015) and Walędzik and Mańdziuk (2018) applied UCT to the resource-constrained project scheduling problem. This problem involves scheduling a set of activities to a set of workers while satisfying resource constraints and constraints related to the sequence of activities. Their aim is to make the solution risk-aware. These risks include employee absence, major project duration underestimation or external risks about on which Walędzik and Mańdziuk (2018) do not provide any details. One of only few other non-game theory applications of UCT is that of the vehicle routing problem presented by Mańdziuk and Świechowski (2017). They first map the vehicle routing problem onto a tree-like structure. Then, a static solution is generated, which is then input into the UCT method. They generated random traffic jams based on pre-defined probability distributions and distributions for length and intensity. Based on these random disruptions, vehicles are rerouted by taking simple actions such as switching the client order, do nothing or to switch clients between vehicles. Mańdziuk and Świechowski (2017) and Walędzik and Mańdziuk (2018) encountered two problems in their application of UCT: (1) the tree branching factor became too high because of the number combinations of risk responses and activities that can start simultaneously, and (2) the many project states that could occur. They both concluded that UCT could only be directly applied when solving for simplified states and actions. Walędzik et al. (2015) did reduce computational time using a heuristic to limit the amount of cases the UCT algorithm was invoked. For their implementation, Walędzik and Mańdziuk (2018) claim that the performance improved over a series of experiments. However, they only mention applying the solution methodologies to 'the most sophisticated project' and report the order of magnitude of computational time only. As proven application of MCTS are limited, it remains to be seen how MCTS performs on airline crew (training) scheduling problems.

## 3-2-2 Schedule Recovery

The more advanced evaluation methods - discrete event generation and Monte Carlo Tree Search - rely on random disruptions and solving them to estimate schedule performance. In practice, airlines use recovery models as decision support system to do so. As such recovery models are part of the robustness evaluation approach and not directly associated with the state-of-the-art in robustness, it is studied less extensively. Nevertheless, models and methods are compared for solution quality and efficiency. Ionescu et al. (2010) distinguished between simple rule-based recovery and more advanced re-optimization approaches. According to them, simple rule-based methods represent reality more closely than re-optimization approaches. Re-optimization approaches tend to propose large schedule changes, something that airlines find undesirable. Both types of approaches are described next. Lastly, research into recovery in the railway industry is studied in to learn from more recent literature.

### Rule-Based Recovery Models

Rosenberger et al. (2002) applied rule-based recovery to solve flight schedule disruptions. They proposed the following policies of varying complexity:

- **Push-back recovery:** A simple routine in which flights are delayed until the scheduled aircraft and crew are available. This procedure works well with sufficient slack and short delays, which is rarely the case in reality.
- **Short cycle cancellation:** Instead of solely propagating delay through the network, flights can be cancelled as well. If some weighted cost function for cancellation becomes lower than the cost of delaying the flight, it will be cancelled.
- **First available reserve crew selection:** When a reserve crew member is needed, select one with the lowest excess time after covering the open duty. More formally, select a reserve crew for which the weighted sum proposed in Equation 3-3 is minimized.

$$excess(\lambda) = \beta_1 \cdot max\big\{rt(\lambda) - ct, 0\big\} + \beta_2 \cdot dt(cb(\lambda), ds(f_1)) + \beta_3 \cdot dt(as(f_n), cb(\lambda)) \quad (3\text{-}3)$$

$$\Lambda_1 = min\big\{excess(\lambda) \mid excess(\lambda) < \epsilon, \lambda \in \Lambda(ds(f_1)) \cup \Lambda(as(f_n))\big\} \quad (3\text{-}4)$$

$$\Lambda_2 = min\big\{excess(\lambda) \mid excess(\lambda) < \epsilon, \lambda \in \Lambda(s), s \in B\big\} \quad (3\text{-}5)$$

$\beta_1$ is the weight factor for the waiting time for a reserve crew member to become available, which is comprised of the reserve time that the crew is available ($rt$) minus the current time ($ct$). $\beta_2$ weighs the deadhead time required between the crew base ($cb$) and the departure station ($ds$) of the first flight of the (partial) pairing. $\beta_3$ weighs any deadhead time required between the arrival station ($as$) and the crew base. Both deadhead factors are zero when the disrupted pairing concerns one starting and ending at the same crew base. If not, the usage of deadheading should be minimized, which is achieved via Equation 3-4. Here, $\Lambda_1$ represents the reserve crews that have an excess time below some limit $\epsilon$ and have their domicile equal to the departure station of the first flight or are stationed at the arrival station of the last flight. If this set is nonzero, a reserve is called when needed. If such a reserve is not available, then the algorithm searches for a reserve crew ($s$) at all other crew bases ($B$), as represented by Equation 3-5. If the set of available reserve crews $\Lambda_2$ is empty as well, then no reserve crew is available within the posed limits ($\epsilon$) and another recovery solution should be opted for. This straightforward method of determining which reserve to use could be adapted and used for training reserves as well. Even deadheading cost might be applicable if training activities are scheduled at an external facility and then disrupted.

The above rule-based recovery policies are simplistic and tailored to solve disrupted flights only, but could incite to similar implementation to the field of cockpit crew training scheduling research. The push-back recovery policy is not applicable to cockpit crew training as simulator training is scheduled on fixed time slots Sohoni et al. (2003). When disrupted, the simulator training session is cancelled almost all times. However, an opportunity lies in an approach similar to short-cycle cancellation policy. It could be attempted to cancel the session with least priority in terms of due date restrictions. The first available reserve crew selection policy can also be adjusted to be applicable to cockpit crew training. Rules can be used to determine which reserve crew member satisfies training qualification requirements and / or timing constraints.

Other researchers have further developed this area by proposing and testing more complex rule-based recovery policies that might- or might not be suited for training disruption recovery. Ionescu et al. (2010) simply cancelled flights whenever a pre-defined delay threshold was exceeded. When delay was below this threshold, they tried to swap crew and aircraft to reduce delay impact. Unfortunately, they did not present details on swap selection criteria. Neither did Shebalov and Klabjan (2006), which served as inspiration for this research.

Bayliss (2016) presented a similar rule-based recovery policy with a specific aim at using reserves. One of the several differences with the recovery policy of Ionescu et al. (2010) is the implementation of a variable cancellation threshold. Bayliss (2016) also presented a model to allow for a series of swaps for the purpose of delay-reduction instead of delay-prevention. They put as a condition that the alternative resource cannot be delayed for the next activity in their original schedule. This approach yield more realistic recovery plans as swapping is rarely limited to a single instance. Complex rules on instructor qualifications and requirements make that swapping multiple resources might lead to less disrupted resources after all. Even if it only is to minimize- or eliminate crews missing their due date.

Other additions of the research by Bayliss (2016) are the so-called absence only recovery policy and the look-up table approach. In the absence only policy, they only allowed for reserves to be used when crew is absent and not for delay-elimination. Although not explicitly mentioned, they also assigned the first available reserve crew. The look-up table approach is based on learning. The rules that are used to review certain reserve options are updated based on previous performance. Both approached showed similar performance in terms of reserve usage rates and cancellation rates. Bayliss (2016) did not report on computational time, although it is expected that these rule-based methods are sufficiently fast in conjunction with simulation evaluation approaches. This is deduced from their note to not test more computationally intensive approaches on a multitude of simulated scenarios.

### Re-Optimization Models

Abdelghany et al. (2004a) developed a decision support tool to automatically recover from irregular flight operations. Their model, aimed at generating a solution both efficiently and with minimum changes, re-optimizes assignments in combination with a rule-based pre-processing methods. They consider a multitude of options such as delaying, using stranded crew, swapping crew, deadheading, using stand-by crew and using reserve crew. According to their definition, stand-by crew are placed at the airport while reserve crew are positioned at home. Abdelghany et al. (2004a) come up with a general framework of prioritizing all recovery actions. Any delay of less than 15 minutes is untouched. Then they look at using stranded crew as this is a resource that is wasted otherwise. However, it is not a reliable source of recovery possibilities. Swapping is the next most cost-efficient solution, which will only negatively reflect in terms of crew satisfaction. Standby crew and reserve crew are generally used for more severe delays. As a last resort, they delayed or cancel flights.

Abdelghany et al. (2004a) represent the above priority scheme implicitly via a simple cost-minimization Mixed Integer Linear Programming (MILP) model. A linear delay penalty factor is included in the objective to prevent solutions with high delay from being generated. The recovery actions and limitations are enforced through the constraints of the model. As an example, they specify the number of available reserve crews per airport, limit the number of swaps and restrict the amount of deadheads. They solve this model for a specific time window in the rolling horizon, after which the recovery action limits are updated for the next iteration. Under the assumption that all resources and recovery action limits are known, Abdelghany et al. (2004a) solved for a real-life historical disruption scenario in one minute and 51 seconds. This run time is acceptable when solving real-time disruptions, but it is too long when solving for numerous scenarios as part of a robustness evaluation procedure.

Abdelghany et al. (2008) extended their earlier cost-minimization MILP model to include aircraft resources and an explicit cost penalty for cancellation cancellation. They also took two additional pre-processing steps. First, they generated a set resource-independent flights in time and space using a simple heuristic. Second, they generated a set of undisrupted flights that could potentially provide resource swaps. A simple rule-based method was used to find only a limited amount of eligible flights that depart within $x$ minutes from the departure time of the disrupted flight at the same airport. The integrated approach was tested on the same sample problem as that of Abdelghany et al. (2004a). A faster computer, using limited swap opportunities and disregarding reserves brought down the solution time to 36 seconds. This is still long when proactively solving numerous random scenarios.

Although the model presented by Abdelghany et al. (2004a, 2008) is promising, the real performance of the full model cannot be assessed. The MILP model is linear whilst in reality, cost of disruptions could be highly nonlinear. Moreover, the user needs to tune the cost parameters which directly influence the output of the model. Still, many parallels can be drawn with robust training scheduling. They presented an extensive list of recovery actions that, for the majority, also apply to training schedule recovery. The interaction effects between these actions might guide towards selecting the most important ones for cockpit crew training recovery. Lastly, the rolling-horizon approach could be employed to assess the impact of sequential disruptions. Analyzing sequential disruptions is more realistic than assuming that all disruptions are known upfront and are fixed, which is one of the standard assumptions in the recovery research domain.

Medard and Sawhney (2007) not only focused on disruptions occurring at the day of operation, but also implicitly included the effect on the roster of the weeks after. They considered disruptions such as flight delay, flight cancellation, crew illness and changes in fleet assignment. All but the first also apply to training scheduling albeit not being related to flights. Their model objective is to minimize the number of crew affected by any of the disruptions. As a simplifying measure, Medard and Sawhney (2007) let crew stay together as much as possible when re-planning. Also, only subsets of crew members are considered for rescheduling to reduce run-time, but did not provide details on the subset selection method employed.

In the model, Medard and Sawhney (2007) define a recovery time window in which the original roster of the subset of crew members must be restored. Any flights, activity, day off and training activities are pre-assigned and cannot be changed. Only activities in between are unlocked for replanning. The fixed activities, in combination with duty rules and the fly-together assumption are used to prune the legal duty network. However, by pre-assigning training, Medard and Sawhney (2007) implicitly assume that disruptions cannot occur to any of these activities and hence they simplify reality. Although beneficial for computational time, pruning also reduces the flexibility and robustness (Ionescu and Kliewer, 2011), meaning that careful judgment is needed for selecting a subset of crew to re-schedule for.

Medard and Sawhney (2007) use a tree-based greedy search algorithm to generate multiple rosters per crew member, which serve as input for the IP model. They check roster legality using the rule evaluator presented by Kohl and Karisch (2004). Medard and Sawhney (2007) also implemented a reduced cost column generation approach based on the k-shortest paths algorithm applied to the Crew Dependent Network (CDN). They then compared both approaches on several disruption scenarios to test for the number of crew affected, number of illegal crew, number of open positions and computational time. The column generation method consistently showed better quality solutions at the cost of higher computational time. The way in which Medard and Sawhney (2007) present these results make it difficult to quantify the exact difference in solution quality. It also remains unclear if the used scenarios are representative. Although they tried to make the solution methodology more efficient, the computational time still lies around the one minute mark for realistic size test cases. Compared to the run time of Abdelghany et al. (2008), the greedy search and IP model are not fast enough to be used in a simulation based evaluation approach.

Other researchers focused on more specific parts of a re-optimization recovery model. Lettovský et al. (2000) presented a cost-minimization Integer Programming (IP) model with an explicit focus on deadheading. Deadheading is considered to be an expensive yet necessary action to restore roster feasibility that is required when: (1) crew is out of flying time, (2) a crew is deadheaded to cover part of a disrupted pairing, or (3) stranded crew is re-positioned to cover (part of) another pairing. Deadheading is applicable to training when having multiple domiciles or when providing training at third parties. Next, Nissen and Haase (2006) used a duty-period-based network model instead of a pairing-based approach, which allowed for shorter re-scheduling time windows and thus lower computational time. A duty-period-based network formulation is ideal when employing a column generation approach. Such an approach iteratively considers the next best option until the optimal solution is found. Also note that Nissen and Haase (2006) are one of the few to present a recovery model applied to a European airline where crew salaries are fixed. This impacts the recovery action selection procedure whenever it is based on cost.

#### Recovery in the Railway Industry

Cacchiani et al. (2014) reviewed railway crew recovery, which is similar to the airline crew recovery problem. The biggest difference is that single day schedules are used in the railway industry, where crew pairings at airlines could cover multiple days. Nevertheless, Cacchiani et al. (2014) identified other solution methods applied to railway crew recovery that are applicable to the airline crew recovery problem as well.

Abbink et al. (2009) used an agent-based model for considering swap opportunities in real-time. In case of any disruption, autonomous train driver agents negotiated about swapping duties or tasks. However, this approach turned out to be too computationally expensive for larger problem instances. Both Potthoff et al. (2010) and Rezanova and Ryan (2010) proposed a simple set covering model for the railway crew recovery problem explicitly considering duty replacements (i.e. swaps), deadheading and cancellation. Due to the large problem size and computational time, Rezanova and Ryan (2010) proposed a dynamic column generation approach. They considered a subset of train drivers and tasks and iterated over an increasingly large subset whenever the solution turned out to be unsatisfactory. Potthoff et al. (2010) used dynamic constraint aggregation instead. This method, similar to those presented by Lettovský et al. (2000) and Nissen and Haase (2006), cluster a number of tasks and generate only one constraint for this. Although this method is faster, it becomes difficult to solve on large scale recovery problems. Veelenturf et al. (2012) extended the model of Potthoff et al. (2010) to also consider re-timing of train rides. They clustered a number of copies of tasks all with a (slightly) different start- and end time. Via an overarching constraint, only one of the timing options is forced to be covered. Veelenturf et al. (2012) reported less cancellations compared to Potthoff et al. (2010), both with and without consideration of stand-by crew.

The method of Veelenturf et al. (2012) integrally considers recovery and re-timing. Such an approach could improve on solution quality of recovered cockpit crew training schedules. An important issue is that of rescheduling simulator training activities due to limited capacity and due date restrictions. If spare capacity is available, such a model could already consider re-timing. By explicitly combining it with delay-reduction swaps (Bayliss, 2016), the impact of missing due dates might be mitigated.

### 3-2-3 Synthesis of Schedule Evaluation and Recovery

In this subsection, literature on schedule evaluation and schedule recovery has been studied. An overview of all reviewed methods and models is presented in Table 3-2.

Although analytical evaluation models have been successfully applied by Bayliss et al. (2012) and Bijvank et al. (2007), they do require extensive model simplifications. According to Van Den Bergh

**Table 3-2:** Overview of all models and methods on evaluation and recovery

| Evaluation of robustness |
|---|
| • Analytical evaluation |
| • Discrete event simulation |
| • Discrete event simulation with feedback / learning |
| • Monte Carlo Tree Search |

| Recovery of disrupted schedules |
|---|
| • Rule-based recovery |
| • Re-optimization recovery |

et al. (2013) this is also why discrete event simulation is one of the most seen approaches in literature. Such simulation models allow for evaluation of more complex schedules in a stochastic environment, which is needed to prove robustness. Historical data on disruptions can be converted to synthetic distributions, which can then be used to explore the full disruption domain without losing the realistic nature. Ingels and Maenhout (2015) and Ingels and Maenhout (2017) have applied this approach. Both generated a crew schedule first and then adapted it for robustness with the same model. Afterwards, they perturbed- and recovered both schedules which allowed them to make a valid comparison between a robust and non-robust schedule. After that, their approach becomes less convincing. They compared a worst-case scenario, best-case scenario and the average of all simulated scenarios. In a stochastic operating environment, the worst and best case scenario's are not representative. Also, the average scenario does not exist. Instead, a stable schedule is needed that performs well under most if not all circumstances. The latter could be accomplished by incorporating a machine learning or reinforcement learning algorithm in the evaluation approach. The basic idea is to feedback the measured performance to the robust scheduling model and adjust it for the next evaluation iteration. A drawback of such a method is that trace-ability is lost once the algorithm starts learning. As a consequence, the robust scheduling model needs to be trained on a data-set before it can be compared with the non-robust base schedule in a valid way. Then, the schedule is ought to converge to one with the most effective implementation of the robustness indicators only. When sufficient historical data is available, using a discrete event simulation model combined with learning is promising in terms of performance and rate of convergence.

A special kind of reinforcement learning algorithm is found in Monte-Carlo Tree Search. Combining recovery methods with MCTS would allow to randomly sample the disruption space and evaluate the (sequential) effect of recovery decisions- and distill an optimal recovery policy by using the algorithms' back-propagation mechanism (Browne et al., 2012). However, both Mańdziuk and Świechowski (2017) and Walędzik and Mańdziuk (2018) stress that choosing a simple recovery model is key in keeping the problem computationally tractable. They also simplify their models in terms of states and actions for this purpose. MCTS can thus only be efficiently applied to robust training scheduling if the problem turns out to be sufficiently simple or can be simplified without negating its realistic nature. This decision can only be taken after the problem size is made insightful with data analysis. Second, Veelenturf et al. (2012) proposed a method that integrally considers recovery and re-timing. Such an approach could be applied to simulator training where the re-timing options are other simulator slots at potentially other days. By explicitly combining it with delay-reduction swaps (Bayliss, 2016), the disadvantageous aspects of missing due dates could be minimized.

When using discrete event simulation combined with learning or MCTS, a well performing recovery model is needed. Almost none of the researchers explicitly presented and tested performance of the recovery models. Ionescu et al. (2010) compared simple rule-based recovery approaches and re-optimization approaches for a case dealing with On-Time Performance (OTP). Their comparison showed that the rule-based recovery method underestimated OTP compared to the re-optimization approach, but did so with a constant discrepancy of roughly one percent. The average run-time for rule-based recovery over various scenarios is just three seconds, while the re-optimization approach requires several minutes (Ionescu et al., 2010). The rule-based recovery method thus comes with a significant improvement of run-time at the cost of a slightly lower, but constant underestimated performance measure. The only re-optimization recovery approaches that come close to rule-based methods in terms of computational time are those presented by Ingels and Maenhout (2015, 2017). However, these are only applied to small test cases and are expected to show (exponential) increase in run-time for more realistic, larger test cases. If many random simulations are used to estimate training schedule robustness, a rule-based method is preferred. Although the performance largely depends on the specification of the governing rules, schedule evaluation using rule-based recovery methods is indicative of the possible improvements in the field of cockpit crew training scheduling research. One other promising aspect is presented by Veelenturf et al. (2012). They implemented a re-optimization approach with dynamic constraint aggregation that considered re-timing options before solving for the disruption. This could be beneficial for simulator training as this is often highly constraint in terms of resource capacity (Quintiq, 2017). However, this would have to be converted to a rule-based approach to be computationally tractable across random scenarios.

## 3-3 Synthesis of Robust Scheduling Methods and Recovery

As concluded from section 2-3, the current state-of-the-art in airline crew training scheduling is limited and concerned with basic scheduling models and algorithms. When combined with the research area of robustness, as identified by Belobaba et al. (2015), the following research question was formulated:

*What robustness measures can be taken to proactively minimize the impact of disruptions related to an airline cockpit crew training schedule considering relevant rules and regulations?*

This chapter provided a view on the current state-of-the-art on robust scheduling and recovery to identify potential problem approaches to attain the research objective and answer the main research question. This section provides an overview of potential, complete problem approaches and reflects on applicability, pros and cons. This is not only based on the state-of-the-art in airline crew scheduling, but also from research into personnel scheduling, nurse scheduling and other modes of transport.

Researchers such as Ingels and Maenhout (2017), present a general framework that could be employed to proactively achieve and validate robustness in the scheduling domain. This framework also corresponds to the structure of the literature review. First, select an objective, associated model and solution technique for a baseline cockpit crew training schedule (i.e. a non-robust schedule) as described in subsection 3-3-1. Then do the same for a robust cockpit crew training schedule as done in subsection 3-3-2. Lastly, select a robustness evaluation method and corresponding disruption recovery model, as described in subsection 3-3-3. This section provides a combined synthesis of all available methods to compile such a complete problem approach. Choosing one element in each of the cells displayed in Table 3-3 forms such a complete solution approach. However, some combinations might be ill-suited or dependent on each other. Some of these choices for models are clear whilst others must be determined on based on actual training disruption data or based on actual experience trying to model the problem. A reflection upon the models is provided below.

### 3-3-1 Cockpit Crew Training Scheduling

According to Kohl and Karisch (2004), generating an optimized cockpit crew schedule including flights and training is difficult and involves various assumptions. As the quality of commercially

**Table 3-3:** Overview of all models and methods presented in this literature study

| Training scheduling objectives |
|---|
| • Training cost minimization<br>• Training cost minimization and workload balance / crew satisfaction<br>• Training cost minimization and robustness |

| Robust scheduling objectives and indicators |
|---|
| • Delay penalty<br>• Number of swap opportunities<br>• Reserve utilization versus unavailability |

| Training scheduling and rostering |
|---|
| • Sequential exact approach<br>• Integrated exact approach<br>• Integrated construction based heuristic |

| Robust training scheduling and rostering |
|---|
| • Linear exact approach<br>• Stochastic model<br>• Heuristics |

| Evaluation of robustness |
|---|
| • Analytical evaluation<br>• Discrete event simulation<br>• Discrete event simulation with feedback / learning<br>• Monte Carlo Tree Search |

| Recovery of disrupted schedules |
|---|
| • Rule-based recovery<br>• Re-optimization recovery |

generated schedules is hard to match, it is necessary to generate a baseline cockpit crew training schedule with a known method instead. It turns out that all training scheduling models presented optimize for cost. Some also optimize for workload balance or crew satisfaction of instructors. As this only adds practical value it is omitted in this research. However, the model must used to generate a robust roster as well, but more on this in subsection 3-3-2.

The cockpit crew training scheduling problem is difficult mainly because of interdependencies and resource-constraints. Although important, the crew availability interface with the flight schedule is neglected under the assumption that sufficient crew members are available and a training roster is generated before flights are assigned. Other researchers also neglected simulator availability and / or instructor availability. The models for conversion training, recurrent training and instructor scheduling could be solved sequentially to obtain a full model and solution. An iterative approach is then needed to actually solve the models to feasibility. Instead, Holm (2008) integrated training scheduling for conversion- and recurrent training already. Her scheduling model can be extended to also assign trainees as it works with a set of trainees in need of a particular type of training. With inspiration from Xu et al. (2006), the model could also be extended to assign qualified instructor as well. This novel approach would take into account all training related interdependencies and is solved to feasibility at once. The remaining problem lies in the computational time of 18 hours of the basic variant of the model. Given the extensions, a faster heuristic solution methodology must be used. Qi et al. (2004) presented such a construction-based heuristic that can be applied to the extended LP model based on the model of Holm (2008). As only minor adaptations are needed in the selection of subsets for each iteration, the implementation of the algorithm is promising. The rolling approach provides a natural way of dealing with the carry-in and carry-out of activities in the scheduling time window. All in all, the algorithm is expected to bring down run time to under one minute, which is sufficiently fast. As such an algorithm mimics the manual scheduling process, the solution is also expected to be of sufficient quality.

### 3-3-2 Robust Cockpit Crew Training Scheduling

The cockpit crew training scheduling model of Holm (2008) and the construction heuristic presented by Qi et al. (2004) is ideally extended to also produce a robust training schedule. According to Shebalov and Klabjan (2006), isolated optimization of robustness indicators has shown not to always yield acceptable performance from a cost-minimization perspective. Proven methods to combine the objectives are to convert the robustness indicator into a common (cost) factor and reward its implementation (Bayliss et al., 2017) or to cap the increase of cost (Ehrgott and Ryan, 2002). A cost-efficient form of robustness is to add swap opportunities to a schedule. Swap opportunities allow for continuation of prioritized training events. However, swap opportunities are not necessarily predictable as there are timing constraints, qualification constraints but also constraints on the assignment history of swappable crew members. Adding buffers is more predictable, but it is expensive. Buffers could be added to minimize the chance of crew members missing due dates due to disrupted training events. However, this is only beneficial when instructor capacity or simulator capacity is restricted. Recovering from such disruptions before- or after the due date is straightforward. Lastly, reserves could be optimized for training requirement as well (Bayliss, 2016). Although expensive, it minimizes wastage of simulator capacity and maximizes training throughput. All these robustness indicators could be added to the training schedule, but all are effective for different reasons and under other circumstances. Further analysis of practical limitations, disruptions and impact must show the potential of the various forms of robustness.

Many researchers apply simple, but sufficiently effective deterministic Linear Programming (LP) formulations with a robustness term and solve it using heuristics. However, stochastic models are more promising in terms of performance when applied to a stochastic operating environment (Tam et al., 2011). Stochastic models are combined with simulation-based approaches and converge to the most promising parts of the schedule that contribute to robustness. Shebalov and Klabjan (2006) reported run times of half an hour for adding buffers and swaps solved to optimality, while Tam et al. (2011) reported computational times of two hours to approximate the stochastic variant of the model. For reserve pairing and scheduling, the computational time ranges from a tenth of a second for rule-based methods to half a second for the probabilistic model and three seconds for an efficient solution methodology to an exact problem formulation (Bayliss et al., 2017). As this research is recent the reported run times are considered to be representative. However, the methods are deterministic. Applying a feedback mechanism based on random disruption scenarios would make it stochastic. The stochastic variants of the robustness indicators are preferred due to the added accuracy and academic value. However, it comes at the cost of increased run times.

### 3-3-3 Schedule Evaluation and Recovery

Stochastic optimization best suits the variable operating environment of airline crew training, which in turn requires a simulation evaluation technique. Such simulations provide inside in expected performance and provide a way of comparing a robust- and non-robust schedule under the application of the same disruptions (Ingels and Maenhout, 2017). Extensively repeating the process also allows for quantification of robustness. It can then also be compared to the robustness an existing airline schedule to quantify potential benefits with respect to current airline operations. However, as the model is solved for numerous random scenarios, all methods used must show a good trade-off between solution quality, computational cost and ease of implementation. Another requirement is the availability of sufficient historical disruption data. This data is used to quantify the problem in terms of disruptions and probabilities, which has never been done before in literature. An interesting extension is to test the model's behaviour under dynamic distributions provided that the underlying data set is large enough to accurately do so. The disruption probability itself could, for example, be made dependent on the month of the year. Due to the absence of literature on robust airline cockpit crew training scheduling, using historical data is also the only possibility to accurately quantify the problem size and define promising research directions.

Evaluating the average disruption scenario like Ingels and Maenhout (2015) did is not valuable. On the contrary, schedule stability is desired, which means that a learning approach becomes suitable. These algorithms use a feedback mechanism to update a reward function as a robustness attribute is used more often for the next iteration in the disruption recovery process. This is not only a novel application, it is expected to make the robust cockpit crew training scheduling model converge to a good quality solution without the need for tuning parameters or bonus cost factors. However, the working principles and learning effects of the model become less clear. Nevertheless, using a discrete event simulation model combined with learning is promising in terms of performance and rate of convergence.

As opposed to these state-of-the-art artificial intelligence inspired algorithms, recovery models have been studied more extensively. Still, Ionescu et al. (2010) are one of few to explicitly analyze and compare performance of the rule-based recovery and re-optimization approaches. They showed that rule-based recovery models underestimate solution quality by a constant margin of one percent at the benefit of much faster solution times. Their rule-based method solved in three seconds in contrast to the several minutes needed by the re-optimization approach. More recent research by Bayliss et al. (2017) showed that run times below one tenth of a second per iteration is possible for rule-based methods as opposed to three seconds for a full re-optimization. Assuming that similar performance can be achieved in for the robust airline crew training scheduling problem, rule-based recovery models are more promising in combination with a simulation-based evaluation approach. For this, run time is prioritized over solution quality. Such a model does however require the set-up of a novel recovery policy for training scheduling inspired on practice.

# Chapter 4

# Research Design

The research question defined in the research gap is addressed in the subsequent chapters using methodologies and models inspired by the literature reviewed. But first, the scope of the research is defined in section 4-1. Then, the research objective is captured into a research framework and methodology in section 4-2. The main assumptions related to both topics are listed in section 4-3. Assumptions specific to an element of the research framework are treated in subsequent chapters. Under these assumptions, the cost differentiation in the training scheduling & assignment problem is expected to be small. For that reason, the high cost associated with disruptions and the impact on crew availability justifies the research into robustness of cockpit crew training schedules. Throughout this research, robustness is defined as the capability to deal with- or absorb negative effects of unexpected events. This capability is expressed as a combination of scheduling cost and expected recovery cost. This definition is based on that of others such as Ingels and Maenhout (2015) and Shebalov and Klabjan (2006).

## 4-1 Scope

This section defined the wider context in which the cockpit crew training problem is placed. Figure 4-1 displays the crew scheduling process. Each of the blocks is treated from left to right.

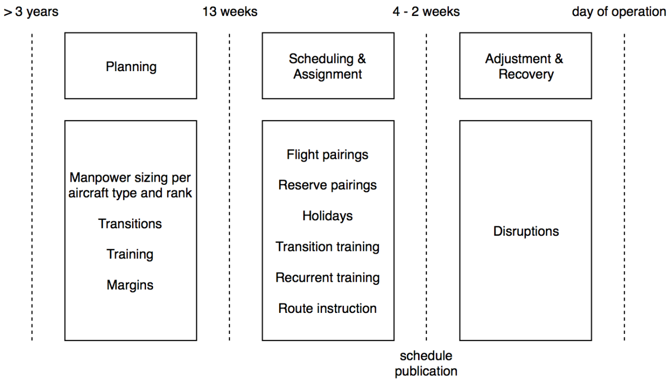

**Figure 4-1:** The airline crew scheduling process with crew training

Years in advance of the day of operation, airlines start solving the manpower planning problem. According to Holm (2008), the objective is to have the correct amount of crew members of the correct qualification at the right moment in time. Crew is needed to operate the flight schedule, but the airlines also account transitions, training and margins. Things become more complex when an airline operates multiple types of aircraft as cockpit crew members are qualified to operate a single, or at most two, aircraft types. The hierarchical nature of the airlines is captured in a fixed career path specifying the progression of crew members in both rank and aircraft type. These transitions are accompanied by transition training programs. The time invested in these transition training programs need to be accounted for when defining the desired number of crew members. A similar factor is applied for recurrent training to keep crew continuously qualified to operate the aircraft. On top, the manpower planning problem accounts for other margins such as crew illness and operational disruptions.

Weeks before the day of operation, the airline starts the scheduling and assignment process. Airlines optimize for flight pairings, reserve pairings and holidays. However, airlines also need to mind crew preferences and flight requests. On top, the legal requirement to satisfy the training demand is enforced. Next to all flights, reserve duties and holidays, crew must be paired to undergo recurrent training or transitions training. The cockpit crew training scheduling problem deals with the timing issues of training courses. The cockpit crew training assignment problem, which is interdependent with the training scheduling problem, assigns individuals to each of these sessions in various roles. Every training session requires a pair of trainees, a sufficiently qualified instructor and potentially a helpout instructors. The latter is required whenever the trainees are not of complementary rank or whenever a single trainee is assigned. Working rules are in place in order to ensure that instructors meet aircraft exposure requirements themselves, but in general that the balance between flying and instructing remains feasible. The scheduling and assignment problem is thus concerned with balancing the demand and supply of crew to satisfy all legal requirements and network requirements, while maintaining efficiency. Provided that efficiency is not suffered, this problem becomes ever more complex when more crew members are employed by the airline.

Schedules are then adjusted and recovered for any disruption that occurs after the schedule has been published. The goal of these adjustments or schedule recovery is to re-establish schedule feasibility. Disruptions in the airlines' network can propagate to the training schedule by means of, for example, minimum rest requirements. Disruptions in the training schedule can propagate to the flight schedule by means of reduced crew availability caused by missed due dates. From the perspective of training, any disrupted training in the former category is secondary, whilst the latter category is primary. This thesis focuses on the primary disruptions to eliminate the need of extensive modelling of a broader crew schedule. Primary training disruptions can be caused by illness or crew leave. Both are, in general, short term disruptions with a notification date close to the day of operation. The impact could be a missed due date resulting in the crew member being unable to fly, or wastage of resources.

With the basic knowledge on the crew scheduling process, the scope on airline cockpit crew training scheduling is narrowed down to scheduling and assignment plus the recovery phase. In other words, a secluded training schedule is generated and then disrupted and recovered to test its robustness. The output of the planning phase with respect to training is assumed to be known and fixed. As the airline processes are simulated, there is no critical operational requirements for run time. Nevertheless, the model(s) should be able to solve for problem sizes experienced by airlines. This could encompass up to 5,000 annual training events and up to 1,000 crew members of which up to 200 are instructor. The realistic nature must be preserved when making assumptions to limit complexity to allow the problem to be solved during a nine-month master thesis project. Finally, integration of the model(s) into a more elaborate scheduling and assignment model should be possible. This means keeping inputs and output generic is essential. This also allows the developed model(s) to be used in conjunction with current airline crew (training) scheduling and assignment solvers to analyse robustness or even to use the model(s) to solve the cockpit crew training scheduling problem. Finally, the scope of the thesis is limited to a single fleet type operated on an European network. This is advantageous for the problem size and does not impact the validity of the research nor the attainment of the research objective. Both ranks, captain (CP) and First Officer (FO) in this case, are integrated in the problem of crew training scheduling because of variable crew pairing.

## 4-2 Research Model Framework and Methodology

To attain the research objective of making recommendations on increasing robustness of the cockpit crew training schedule, a research model framework is drafted. The schematic overview is provided in Figure 4-2 and all elements combined constitute a total problem approach. Each layer is explained below in order of input, process, output and evaluation. The process layer is treated in more detail.

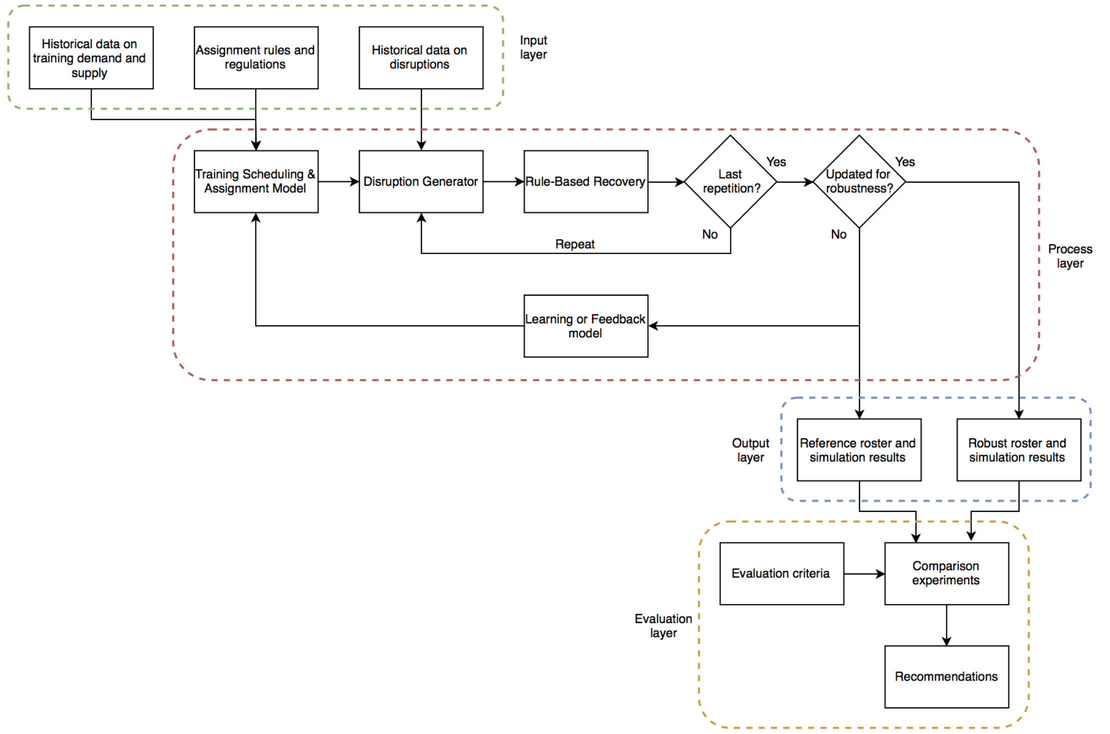
Historical data on training demand and supply
Assignment rules and regulations
Historical data on disruptions
Input layer
Training Scheduling & Assignment Model
Disruption Generator
Rule-Based Recovery
Last repetition?
Yes
No
Updated for robustness?
Yes
No
Repeat
Process layer
Learning or Feedback model
Output layer
Reference roster and simulation results
Robust roster and simulation results
Evaluation criteria
Comparison experiments
Evaluation layer
Recommendations

**Figure 4-2:** Framework and research model for the robust airline cockpit crew training problem

### Input layer

The input is adapted to the requirement of segregating training scheduling from the overarching crew scheduling problem. This means that historical data on training demand and supply of crew is used to ensure the schedule feasibility with a limited interdependence with the flight scheduling and assignment problem. In this context, feasible means that the isolated training schedule does not lead to crew availability issues later on when scheduling flights. The words later on indicate that training scheduling is prioritized over scheduling of other activities, as mentioned by Kohl and Karisch (2004). The minimal interdependence is also captured by taking into account the need to cover flights by leaving gaps in the schedule large enough to assign a flight duty. Implementation of rules from a Collective Labour Agreement (CLA) and using historical training demand data of an European legacy carrier also ensure such a balance. Full specification of these rules is treated in chapter 5.

The input block on the right displays that historical data on disruptions is input into the disruptions generator to simulate the disruptions occurring in the adjustment and recovery phase. The disruptions are scoped to short(er) term disruptions only. Any decision on disruptions known long in advance are postponed until the last 72 hours before the start of that duty. Moreover, the scope is limited to primary disruptions. This means that illness and leave are modelled and solved for, but any subsequent disruption or the impact thereof is neglected to limit complexity. This means that historical probabilities on illness and crew leave are sufficient to simulate disruption scenarios.

### Process layer

The process layer consists of multiple models and algorithms that are jointly called the research model. The idea is that a training schedule is generated with a calibrated model and algorithm. It is then disrupted and recovered in a simulation environment to extensively test the schedule's performance in a stochastic environment. This allows to quantify the robustness, which is defined as the trade-off between stochastic recovery cost and deterministic schedule cost. The objective is to bring down the expected recovery cost while keeping the deterministic schedule cost in check. The simulation results on the reference roster are then used in a learning / feedback loop to update the training scheduling & assignment model to generate a robust roster using the same calibrated model and algorithm. The disruption and recovery process is then repeated for the robust schedule in order to make a comparison afterwards. All models are briefly described below in order of occurring in the framework.

1. **Training Scheduling & Assignment Model (TS&AM):** A cost minimization LP model is formulated to integrally schedule the training courses and assign trainees, instructors and helpouts when needed. The model is inspired on research by Holm (2008), but is adjusted to correctly implement the allowance of various crew compositions. A benefit is that the model is (relatively) easy and proven techniques can be used to solve it. However, as the majority of the training events have same cost levels, the LP model actually reviews (almost) all options. This, in combination with integrated scheduling & assignment, requires a faster solution method to reduce the 18 hour run time of the original model of Holm (2008) involving 2,000 training events annually. Qi et al. (2004) developed a fast construction heuristic that schedules training events per simulator slot by reviewing only subsets of training demand and available instructors. The construction heuristic showed run times in the order of magnitude of minutes rather than hours for similarly sized problems. However, using a subset selection heuristic requires specification of selection rules, which could be difficult to draft. Especially as the local decisions could impact robustness of the greater whole. Still, the similar cost levels between training events make the approach valid. The purpose of the TS&AM is to generate a training schedule with a calibrated model and algorithm to allow for a valid comparison between reference and robust variants. The robust training schedule is generated by updating the subset selection algorithm based on learned features of disruptions and recovery actions applied to the reference roster.

2. **Disruption Generator:** A Monte-Carlo Simulation (MCS) approach is then applied to randomly generate disruption scenarios given the roster. Historical disruption probabilities on crew illness and crew leave are applied to the full set of crew involved in the schedule. As explained by Rosenberger et al. (2000), MCS allows to extensively test robustness on the full spectrum of the empirical distributions and combinations of disruptions, which is needed to prove robustness of a schedule. The disadvantage is that a large number of repetitions is needed to gain statistical significance in the results. Associated to this is the requirement that the research model must be able to solve each repetition in limited time.

3. **Rule-Based Recovery:** The run-time requirement is also effective for the recovery model. Ionescu et al. (2010) showed that rule-based recovery methods outperform re-optimization

approached in terms of run time (a couple of seconds as opposed to a couple of minutes) with only a limited price in solution quality. Because a rule-based recovery algorithm is non-existent for recovery of disrupted training events, a novel algorithm is developed. This algorithm uses a fixed order based on efficiency in the various possible recovery actions. Once all conditions are met for one of the recovery actions, it is applied. Next to its speed, a rule-based method is suited to mimic the decision making process of an airline, increasing the validity of the research. The drawbacks are however, that the working mechanism of the algorithm (i.e. the rules) are static and that these rules can be difficult to draft.

4. **Learning / Feedback:** The results of the disruption generator and rule-based recovery model are used by two separate, novel algorithms to achieve crew training schedule robustness inspired on the approach presented by Lagerholm et al. (2000). The first is to apply Proportional Feedback (PF). This algorithm feeds back the expected recovery cost based on features on the disrupted training activity and roster state. The feature values on the training activity and roster state can also be determined prior to scheduling, and based on this, are matched to the simulation results. This matching determines the correct level of expected recovery cost to add to the course scheduling decision variables of the TS&AM. The second algorithm is a Neural Network (NN), which uses the same set of features. The difference is that the NN learns the optimal weights for each of the features and allows for nonlinear behaviour. This requires an additional learning step that requires a vast amount of training data as opposed to PF at the benefit of increased solution quality and exploited relationships between features.

### Output layer

The first time the process layer is entered, the research model will output a reference roster based on deterministic training schedule cost alone and the associated simulation results. The size of this data set of simulation results is determined by the number of repetitions. By default, the model will enter the research model a second time in which the cost based TS&AM is appended with stochastic recovery cost. After finishing the disruption simulation and recovery, the robust roster and its simulation results are also output. If both proportional feedback and the neural network are tested, a the research model is applied a third time. The output layer is then comprised of two (possibly three) different rosters and associated simulation results containing information on the recovery cost and what recovery actions is applied for each disruption.

### Evaluation layer

The evaluation layer is key in attaining the research objective of making recommendations on increasing proactive robustness with respect to the cockpit crew training schedule. Based on evaluation criteria and detailed analysis of results from experiments, ways of introducing robustness can be identified, but more importantly, quantified. The combination might lead to new insights into the effects of robustness in the cockpit crew training schedule. These insights can be obtained from conducting experiments that test robustness under conditions of interest to the airline.

## 4-3 Assumptions

The main assumptions made to implement the described methodology are listed and explained below. The assumptions are needed to (partly) isolate the cockpit crew training scheduling & assignment problem from the overall crew scheduling process. This is to reduce complexity and keep the problem size tractable for research purposes, but all is aimed at keeping the problem sufficiently realistic to maintain validity. The validity of the research is also preserved by making the problem generic. Assumptions specific to any of the models are treated in subsequent chapters where each of the models are treated in more detail.

- A single fleet type of an European airline on a short-haul network with a single hub is considered with all available ranks. All ranks are required because of variable pairing of training activities.
- All training activities are scheduled and assigned to crew in Full-Time Equivalent (FTE). Part-time employment is neglected to reduce complexity of assignment rules and crew recovery.
- The cockpit crew training scheduling & assignment problem is isolated from the crew scheduling problem by scheduling and assigning training events prior to flights and other activities. Balance in crew demand is covered in the weekly training demand derived from historical data. Likewise, other activities are only taken into account by allocating sufficiently large gaps in the rosters of each crew member. Advanced chaining of such activities is neglected. As a result, training demand can propagate into subsequent weeks, but future demand cannot be brought forward to schedule & assign.
- Only primary disruptions of training events due to crew illness and crew leave are generated and solved for. Secondary disruptions, such as illness of an used reserve crew member or a disrupted flight propagating to the training schedule by means of minimum rest time violation, are neglected. This assumption is made because of a lack of unambiguous historical data and the previous assumption on isolating the problem from the flight schedule.
- Re-qualification training is taken into account by historical events only. Simulated disruptions leading to the need of re-qualification is assumed to be of similar magnitude to the historical set of re-qualification activities.

Chapter 5

# Airline Crew Training Scheduling & Assignment

A generic, calibrated Training Scheduling & Assignment Model (TS&AM) is based on research by Holm (2008). It takes input on historical training demand and supply and assignment rules and converts it into a feasible schedule. So far, this is all known and explained. section 5-1 provides more details into the input and further details the model related assumptions. The adapted, integrated TS&AM is then formally defined and explained in section 5-2 followed by a detailed description of the solution method in section 5-3. Next, the learning and feedback algorithms are elaborated on in section 5-4. Combining the model and solution method, possibly with a learning or feedback algorithm will, first and foremost, result in an output schedule. This and the other output(s) of the TS&AM are described in section 5-5. The chapter is ended with a validation and brief conclusion on the suitability and performance of the model and method in section 5-6.

## 5-1 Model Input

Under the assumptions made in section 4-3, the crew training scheduling & assignment model is isolated from all other activities to reduce complexity and prevent excessive run times as reported by Holm (2008). The associated balance in crew demand for flights and other activities such as training is covered by the input data. Historical training demand is input into the model on a weekly basis, meaning that it is highly likely that sufficient crew will remain 'unassigned' to cover flight duties. The differences in the amount of crew assigned is expected to be minimal. It is also considered to be a necessary assumption to ensure scheduling flexibility to properly test robustness in an environment that is unconstrained by external influences. Note that airlines do allow for some degree of flexibility when scheduling and assigning the other activities.

The crew training scheduling & assignment problem is also isolated per fleet. The only interdependencies that exist are crew transitioning between these fleets, leading to variable amount of crew and instructors. By updating the crew and its qualifications on a weekly basis, the assumption on isolation is considered to be sufficiently accurate.

Kohl and Karisch (2004) described the input of the general crew scheduling problem as well and visualised it in Figure 2-1. For the isolated crew training rostering problem, the model input is:

- **Crew:** Identification number, qualifications and assignment history. The set of crew determines who is available at what time and who is allowed to take on certain duties.
- **Activities:** Training demand for conversion training, recurrent training for regular crew and instructors, re-qualification training and training programs known as 'yardsticks'. These sets of training demand specify what type of training is needed by whom on a weekly basis. The yardsticks govern simulator demand, instructor demand and required qualifications for each type of training and combination of assigned crew ranks. The fixed yardsticks are formed by the rules detailed next.
- **Rules:** Rest-time, days off, limits on duty time, qualification rules and crew complement allowance. These rules govern the legality of assignment problem.
- **Resources:** Simulator capacity indicating the amount of available slots per day.

An example of crew data is given in Table 5-1. The list consists of both instructors and regular crew members. Instructors also have their highest qualification defined, which can be one of three categories: (1) Type Rating Instructor (TRI), (2) Type Rating Examiner or (3) Senior Type Rating Examiner (SRE). A TRI is the most basic qualification, which is needed to instruct routine type recurrents. A TRE qualification is needed for other type recurrents, but also exam sessions on conversion training programs. Finally, a SRE qualification is required for recurrent checks on instructor qualifications. All instructors also have an annual instruction counter defined to keep track of the balance in instruction and flying. Both instructors and regular crew members have a defined last instruction or last training date. This date is used to check assignment legality rules and to cater for other duties in between training activities.

**Table 5-1:** Example of input crew data

| Crew ID | Qualification | Last Instruction / Training | Annual Instruction Count |
|---|---|---|---|
| 771485 | TRE | 2018-02-12 | 12 |
| 645872 | none | 2018-01-17 | none |

As can be seen in Table 5-2, the example instructor defined above also has a training requirement. For this specific training activity, that instructor must be assigned in the role of trainee, limiting the amount of options on what instructors to assign to this session and other daily sessions. Instructor recurrent training is identified by comparing the set of training activities and the set of crew members and their qualifications. The required training entails information on what recurrent session is required (R plus a number identifier specifying the type of recurrent) or specifies that conversion training is needed (C plus a number identifier to indicate the day of the program). The original date indicates the original schedule, which serves as a selection requirement on what historical training demand to consider when.

**Table 5-2:** Example of input activity data

| Crew ID | Required Training | Original Date |
|---|---|---|
| 771485 | R3 | 2018-02-24 |
| 645872 | C | 2018-02-24 |
| 547892 | R1 | 2018-02-25 |

The simulator demand, instructor demand and instructor order, lowest qualification required and helpout demand is captured in training curricula known as yardsticks. Each type of training course has its own yardstick. Helpout demand is only specified for nonstandard crews. A nonstandard crew is defined as an assignment of a set of two crew members of equal rank, or a single crew member. On the contrary, a standard crew is defined as an assignment of crew with supplementary ranks (i.e. a CP and FO together). The demand per day of the training program is related to session identifiers, which have been explained previously. As can be seen in Table 5-3, the example yardstick of a conversion course includes simulator demand and instructor demand per day of the training program. For multiple-day long yardsticks for which multiple instructors are needed to cover the entire program, the instructor order specifies the duties per instructor. The assignment problem is concerned with a decision on instructor order, which could limited by the dynamic minimum qualification required. As said, the helpout demand only applies to nonstandard crews. Any mandatory days-off at the end of the program and in between if specified by working rules are also included by means of a zero resource demand.

**Table 5-3:** Example of a yardstick

| Session Identifier | C1 | C2 | C3 | C4 | exam | off | off |
|---|---|---|---|---|---|---|---|
| Simulator Demand | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Instructor Demand | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Instructor Order | 1 | 1 | 1 | 1 | 2 | na | na |
| Minimum Qualification Required | TRI | TRI | TRI | TRI | TRE | na | na |
| Helpout Demand | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

A training event can only be scheduled if sufficient simulator capacity is available during the entire length of the yardstick. An example of simulator capacity data is given in Table 5-4. Information is provided in terms of integers indicating the number of available slots at a certain moment in time defined by a date and time (indicated by a letter). The simulator availability is constantly updated for scheduled training events.

**Table 5-4:** Example of input simulator capacity

| | 2018-01-01 | 2018-01-0 | 2018-01-03 | 2018-01-04 |
|---|---|---|---|---|
| A (06:15 - 09:45) | 1 | 1 | 1 | 1 |
| B (09:45 - 13:15) | 1 | 2 | 1 | 1 |
| C (13:15 - 16:45) | 1 | 1 | 0 | 1 |
| D (16:45 - 20:15) | 1 | 2 | 1 | 1 |
| E (20:15 - 23:45) | 2 | 0 | 0 | 1 |

As many variants of labour agreements exist, and there is no distinction in right or wrong, the working rules applied in this thesis are elaborated more thoroughly. Note that only common rules are applied for the sake of generality. The implementation of the assignment rules is treated in section 5-3, but a brief explanation is added below.

- At most one simulator training activity can be assigned to an individual on a daily basis. This rule is trivial input for the model.
- A maximum of five consecutive simulator training sessions can be assigned to an individual without off-time.
- Each consecutive period of instructing is, if possible, followed by a regular flight duty to keep roster diversity.
- A maximum of x simulator training activities are given per individual over the course of a year. For this thesis, the maximum amount is set at 60. Exceeding this limit is penalized, not restricted. It could acts as a means to balance the instruction load between crew, but this is not considered an objective of this thesis.

Other assignment rules are neglected for the purpose of reducing complexity or to limit the scope of the research. The assumptions on assignment rules are accompanied by other general training scheduling assumptions. All are listed below.

- The assignment rules mentioned above capture the crew training scheduling problem with sufficient accuracy. It is assumed that splitting yardsticks in time and not allowing for so-called step-backs in trainee assignment can be neglected. A step-back allows for a change in the start time of simulator sessions on long yardsticks. Each jump to another subsequent starting slot is considered a step-back, which could be to a later time as well.
- Rules concerning the dependency with the flight schedule, such as flight- and holiday requests can be neglected. The added value is practical in nature, but neglecting it comes at the cost of increased scheduling flexibility.
- Conversion courses can only be scheduled in pairs for the purpose of efficiency. These long courses would otherwise require a vast amount of helpout instructors.
- No external training capacity is available. Both for simulator capacity and instructor capacity.
- Simulator capacity is assumed to be constant throughout time. The total capacity is statically adjusted for (national) holidays, (unscheduled) maintenance and capacity sold to / used by third parties. Dynamic simulator capacity can readily be implemented in the model, but will increase complexity by means of additional assignment rules.
- The length of all training activities is captured in fixed 'yardsticks'. To eliminate some complexity, the instructor demand and simulator demand only vary per crew composition assigned to the specific training event and not by other variables such as time, rank and position.
- Instructors are allowed to operate all training events with a lower requirement than their highest qualification. This assumptions removes individual cases that do not frequently occur.
- The set of qualified instructors and associated qualifications is static for the week under consideration to match the weekly training demand.

## 5-2 Training Scheduling and Assignment Model

An integrated Training Scheduling & Assignment Model (TS&AM) is drafted from previous research by Holm (2008), Sohoni et al. (2003), Xu et al. (2006) and Yu et al. (2004). The model is drafted as it involves novel adjustments. First of all, scheduling and assignment are integrated as these decisions are interdependent in all resource facets. Next, instructor assignment is explicitly modelled, requiring a model adaptation in terms instructor assignment and role assignment. Furthermore the model is adjusted to schedule standard and nonstandard sets of training events exhaustively as opposed to only allowing for assignment sets of two captains as done by Holm (2008). All changes are captured in the sets, parameters, decision variables, objective function and constraints of the model. Each of these elements is treated in order below.

### Sets

The model input is given in the form of sets as described in section 5-1. All sets are described below, where parenthesis indicate subsets based on certain variables indicated by lower case letters associated with the set. For some, multiple variants exist.

$t$: set of iterable time instances defined by date and time.

$K_{(i)}$: set of courses (for crew member $i$).

$D_k$: set of training days of course $k$.

$S$: set of simulator resources, in this case S1 and S2 designating the available simulators.

$I_{((q)pk)}$: set of crew members (with (qualification $q$,) position $p$ requiring course $k$).

$J_{((p)qk)}$: set of instructors (with (position $p$,) qualification $q$ requiring course $k$).

$P$: set of positions, in this case CP and FO.

$Q$: set of instructor qualifications, in this case TRI, TRE, SRE.

### Parameters

The parameters used in the model describe the problem and process of the European legacy carrier to keep things realistic. For confidentiality reasons, the parameter values used in this thesis are kept undisclosed. All parameters used in the model are listed below:

$c_{sk}$: cost of simulator $s$ for course $k$.

$c_{ik}$: cost of assigning trainee $i$ to course $k$.

$c_{jk}$: cost of assigning instructor $j$ to course $k$.

$c_{j_h k}$: cost of assigning instructor $j$ as helpout to course $k$.

$c_{recovery_{kt}}$: cost of recovering from a disruption of course $k$ at time $t$

$n_s$: number of simulators $s$.

$ns_{st(d)}$: number of available slots at simulator $s$ at time $t$ (at day $d$ of the training program).

$cap_s$: trainee capacity of simulator $s$ for a general, standard crew.

$dem_{sdk}$: demand for simulator $s$ for day $d$ of course $k$.

$l_k$: length of course $k$ in days.

$p_j$: priority factor of trainee $i$.

$p_j$: priority factor of instructor $j$.

$P_{kt}$: disruption probability of course $k$ at time $t$.

$ni_{k_{(p)}}$: maximum number of trainees (of position $p$) that can be assigned to course $k$.

$nj_{k_{(q)}}$: maximum number of instructors (of qualification $q$) that can be assigned to course $k$.

$nh_k$: number of helpouts that need to be assigned to course $k$ when a nonstandard crew is assigned.

### Decision Variables

Per simulator slot at each time instance, the model decides what course to schedule in what crew composition and who to assign to it. The assignment includes trainees, instructors and possibly helpout instructors. The scheduling and assignment models are integrated because of their dependency. More formally, the integer decision variables are:

$a_{kt}$: 1 if courses $k$ is starting at time $t$, 0 otherwise.

$b_{kt}$: 1 if course $k$ is starting at time $t$ is scheduled with a nonstandard crew, 0 otherwise.

$x^{p}_{kt}$: number of trainees of position $p$ assigned to course $k$ at time $t$.

$y_{ikt}$: 1 if trainee $i$ is assigned to course $k$ at time $t$, 0 otherwise.

$z_{jkt}$: 1 if instructor $j$ is assigned to course $k$ at time $t$, 0 otherwise.

$h_{jkt}$: 1 if instructor $j$ is assigned as helpout to course $k$ at time $t$, 0 otherwise.

### Objective Function

The goal of the airline crew training rostering model is twofold: (1) schedule a set of training activities to a series of simulator slots, and (2) assign trainees and instructors to these while minimising cost. For the reference schedule these cost are deterministic. However, to obtain a robust roster, stochastic recovery cost are added to the course scheduling decision variables as well. The objective function part on expected recovery cost is added between square brackets in the first term of Equation 5-1.

$$\begin{aligned} minimize \qquad & \sum_{k\in K}\sum_{s\in S}\sum_{t=1}^{N}\{(c_{sk} + c_{jk} + [P_{kt}\cdot \mathbb{E}[c_{recovery_{kt}}]])\cdot a_{kt} + c_{j_h k}\cdot b_{kt} \\ & +c_{ik}\cdot(ni_k + (ni_k - \sum_{p\in P} x^{p}_{kt})) - (\sum_{i\in I} p_i\cdot y_{ikt} + \sum_{j\in J} p_j\cdot(z_{jkt} + h_{jkt}))\} + C \end{aligned} \tag{5-1}$$

The first term represents the course cost, consisting of simulator cost, mandatory instructor cost and the stochastic recovery cost of that course. The second term adds helpout instructor cost to nonstandard courses. The last part of the equation adds a constant offset cost factor to account for trainee cost of a course assigned to a standard crew. This in combination with the third part of equation, adds zero cost when a standard crew is assigned. If a nonstandard crew is scheduled, the third term of the equation represents a remaining 'penalty' cost that is added on top of the constant offset trainee cost. In this way, the model prefers standard courses - which are more cost efficient, but have a higher absolute value in cost - over nonstandard courses without restricting the latter category. The fourth part of the objective function represents a priority factor that is added to make the model converge to a more efficient solution. A cost differentiation is needed to reward consecutive instructor assignment, which is more efficient in combination with assignment rules on mandatory rest periods. Also, trainees might be preferred because of upcoming due-dates or because they have missed it. The latter is inspired on Sohoni et al. (2003).

### Constraints

Without any constraints, the cost minimization model opts to not schedule any training at all, which would violate the airline's legal training obligations. Yet, the set-up of the solution methodology proved to be a difficult way to hard-constrain all courses to be scheduled. Instead, the solution methodology will force each available slot to be utilized for a training event with at least one trainee and leave open simulator capacity when demand is expected to be lower than supply. This explains the first of the constraints listed below. A short explanation on all constraints is added below the specification.

$$\sum_{t=1}^{N} y_{ikt} \geq 1 \qquad \forall t \in \{1,2,...,N\} \tag{5-2}$$

$$\sum_{k \in K} \sum_{d=0}^{min(t,l_k)} dem_{sdk} \cdot a_{ks(t+d)} \leq min(ns_{std}) \qquad \forall t \in \{1,2,...,N\} \tag{5-3}$$

$$\sum_{p \in P} x_{kt}^{p} \leq cap_s \cdot a_{kt} \qquad \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-4}$$

$$a_{kt} - b_{kt} \leq x_{kt}^{p} \qquad \forall p \in P, \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-5}$$

$$\sum_{i \in I_{pk}} y_{ikt} + b_{kt} \geq cap_s \qquad \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-6}$$

$$\sum_{i \in I_{pk}} y_{ikt} = x_{kt}^{p} \qquad \forall p \in P, \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-7}$$

$$\sum_{j \in J_k} z_{jkt} = nj_k \cdot a_{kt} \qquad \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-8}$$

$$\sum_{j \in J_k} z_{jkt} \geq nj_{k_q} \cdot a_{kt} \qquad \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-9}$$

$$\sum_{j \in J} h_{jkt} = nh_k \cdot b_{kt} \qquad \forall k \in K, \forall t \in \{1,2,...,N\} \tag{5-10}$$

$$\sum_{k \in K} z_{jkt} + h_{jkt} + y_{jkt} \leq 1 \qquad \forall j \in J, \forall t \in \{1,2,...,N\} \tag{5-11}$$

$$\begin{aligned} a_{kt} \in \{0,1\} &\qquad \forall k \in K, \forall t \in \{1,2,...,N\}, \\ b_{kt} \in \{0,1\} &\qquad \forall k \in K, \forall t \in \{1,2,...,N\}, \\ x_{kt}^{p} \in \{0,1,2\} &\qquad \forall k \in K, \forall p \in P, \forall t \in \{1,2,...,N\}, \\ y_{ikt} \in \{0,1\} &\qquad \forall k \in K, \forall i \in I, \forall t \in \{1,2,...,N\} \\ z_{jkt} \in \{0,1\} &\qquad \forall k \in K, \forall j \in J, \forall t \in \{1,2,...,N\}, \\ h_{jkt} \in \{0,1\} &\qquad \forall k \in K, \forall j \in J, \forall t \in \{1,2,...,N\} \end{aligned} \tag{5-12}$$

Equation 5-2 forces at least one trainee to be scheduled at every simulator slot to initiate the scheduling part of the model. Then, Equation 5-3 ensures that simulator capacity is not exceeded. For courses stretching beyond a single session, future simulator capacity is also restricted. Equation 5-4 limits the training course capacity in the number of trainees. Whenever the number of trainees between ranks is not the same, a nonstandard course is scheduled. The model identifies this via the nonstandard course constraints of Equation 5-5, which are added for each rank. With the above-mentioned constraints, the model schedules courses based on interdependencies with demand, but it does not yet assign individuals. Equation 5-6, Equation 5-8, Equation 5-10 forces trainees, instructors and helpouts to be assigned respectively in the correct number based on the scheduled courses. The correct rank is ensured via Equation 5-7. Furthermore, Equation 5-9 ensures that the assigned instructor is sufficiently qualified and Equation 5-11 allows each individual to be assigned in one role only (i.e. trainee, instructor or helpout). There is no qualification requirement on the helpout assignment, other than the crew member having to be an instructor. Lastly, Equation 5-12 display the integer bounds of the decision variables.

## 5-3 Solution Methodology

As Holm (2008) reported run times of 18 hours for a one-year training schedule at SAS Scandinavian Airlines. The extension to integrated scheduling & assignment, and the fact that the problem size is larger than the 1,884 recurrents and 69 conversion training sessions scheduled by Holm (2008), a faster solution methodology is required. This is also beneficial for research, as this means aspects of the model can be analysed more easily. The faster solution method is found in an adaptation of the Construction Heuristic (CH) presented by Qi et al. (2004). The heuristic optimizes locally instead of globally by means of rolling along small pieces of the scheduling horizon and solving these sequentially.

The construction heuristic of Qi et al. (2004) uses a user-specified parameter $h$ as the amount of courses to schedule at each iteration. They schedule $h$ classes per iteration, but only fix a partial solution and re-consider the other courses in next iterations. This means that there is no dependency on future time instances other than for conversion courses. Because conversion courses are longer and require more resources at pre-defined times in the most efficient way possible, the RHH will try to schedule such a course first. If there is no demand or scheduling a conversion course is unfeasible, then the model schedules $h$ short(er) recurrent course. According to Qi et al. (2004), a higher value of $h$ results in a higher number of options considered per iteration, and thus a better solution in terms of quality, although expected to be minimal. Nevertheless, Qi et al. (2004) explained that run time increases exponentially with $h$. Keeping computational time in the range of minutes requires the value of $h$ to be lower than seven. Still, the lower bound of this value is given by the amount of trainees and / or instructors to assign to standard- and nonstandard sessions to not artificially limit the output options of the model. This means the parameter $h$ should be chosen to be four, five or six. Qi et al. (2004) proved in their approach that even higher values of $h$ do not yield added performance. On the contrary, they experienced that the model considered more unfeasible options.

The implemented approach in this thesis differs as it considers $h$ classes per iteration of a single simulator slot with the objective of scheduling one. Still, based on extrapolation of run times encountered by Qi et al. (2004), the adapted algorithm is expected to solve problem sizes involving 5,000 annual training events and 1,000 crew members within five minutes. The faster run time is expected to come at little expense of solution quality. In view of relatively fixed training demand determined historically by due dates or based on constant supply, many timing options drop out. Also, not taking into account personalized instructor- and trainee cost, by assumption to keep assignment balanced, eliminates cost differences. The only difference in cost arises from scheduling standard and nonstandard courses because of the impacted efficiency. A nonstandard course requires more instructors and consists of less trainees. But the effect of this is expected to be minimal and insignificant for the purpose of research.

The full algorithmic framework of the CH is visualised in Figure 5-1. It consist of several other heuristics that together convert the input training demand and supply into a feasible training schedule. As mentioned earlier, the model will update demand and supply on a weekly basis by means of the Input Selector (IS). These sets of input are then used in the algorithm to construct a schedule per simulator slot by selecting smaller subsets of training demand and instructor supply by means of the Selection Heuristic (SH). Based on projections of simulator demand and supply made by the IS, the model randomly decides when to account for spare simulator capacity. The decision of spare capacity is just to keep the schedule realistic without putting much effort into deciding about optimal spare capacity allocation. In case of spare simulator capacity or a zero simulator demand, the algorithm moves on to the next iteration. If a course is scheduled, the model enters the Priority Heuristic (PH) to compute cost and priority factors for the subset on training demand based on assignment rules. Then the TS&AM is solved and resources are updated afterwards. The model will move to the next iteration by default, or is terminated when all time instances of all user-specified weeks are considered.

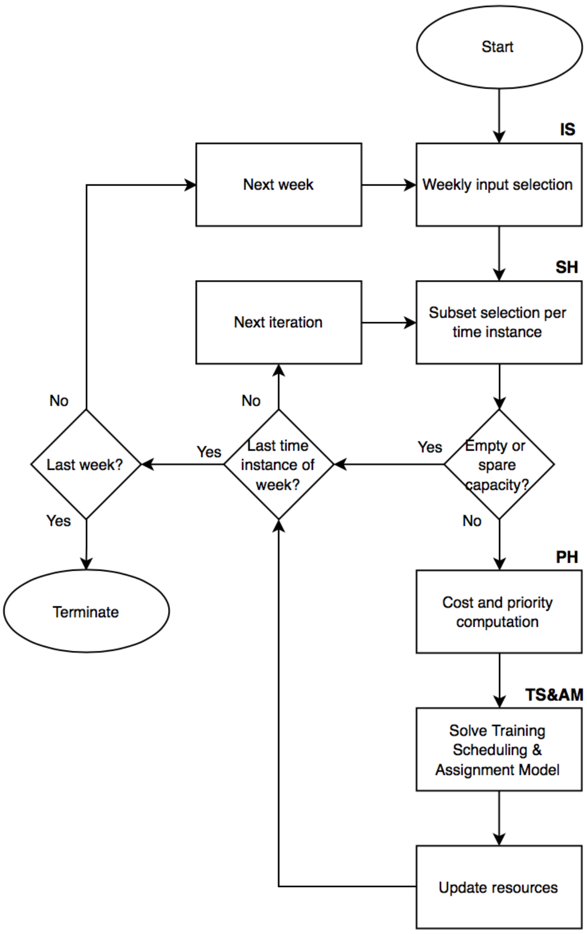

**Figure 5-1:** Algorithmic framework of the Construction Heuristic

Although displayed as two separate heuristics, the SH and PH are linked. Information on prioritized trainees and instructors is used by the Selection Heuristic. The PH then post-processes this into the TS&AM. This combination of heuristic is also a determining factor of solution quality. The combination of the SH and PH is therefore treated in more detail in the remainder of this section. The section is divided into selection and priority schemes are tailored for course selection, trainee selection and instructor selection.

### Course Subset Selection

The course selection categorises types of training activities based on program length. It then starts with the category of the longest course(s) and selects all optional courses are considered per iteration, except when demand has run dry. If demand is non-existent or scheduling of such a course proved infeasible, the course selector moves on to the next category in line and repeats the process of selecting all eligible, unique courses. In specific terms of the the implementation of the cockpit crew training problem in this thesis, a distinction is made between conversion and recurrent training events. All recurrents consist of a single simulation session whilst the conversion courses are longer. After each iteration, the course selector is re-initiated at the longest courses. Any leftover demand (of lower categories) after the week is fully considered is propagated into the new week.

### Trainee Subset Selection

Parameter $h$ is then used for selecting trainees for each of these courses. This parameter indicates the maximum number of options considered per course whilst the minimum of one trainee is covered by the course selection. The selection heuristics is the means to satisfy crew availability rules and obtain a subset in which efficiency is maximized, while allowing for other solutions as well. It uses the following logic:

1. First, a list of non-eligible trainees is composed or updated. This list contains crew members that are not available as trainee because they violate at least one of the assignment rules. Examples are minimum rest requirements, but also the assumption that step-back rules are disregarded.

2. Any crew member that is within a user-specified due date delta ($\Delta$) prior to the due date, is prioritized to be considered for assignment. This also applies to those that have missed their due date. The value for $\Delta$ used in this model is set at five days. This value determines the effectiveness of the priority rules. When few trainees get prioritized, they will be scheduled before their due date with high likelihood, but in case of too many prioritized trainees, the model is not be able to cope with all prioritize crew members. A dynamic priority factor can be implemented in future versions of the model, but this is not done now because it implicitly fixes who to assign when. A discretised priority strategy, such as presented by Sohoni et al. (2003), could also be an option for future versions of the model.

3. If the amount of selected trainees for a specific course is below the parameter $h$, a semi-random supplementary selection is used based on following rules:

   (a) If not yet in the prioritized crew list and a crew member is available with nonzero demand, then randomly select a single captain.

   (b) Similar to that for captains, randomly select a first officer based on the requirement of not yet being considered and being available with nonzero demand.

   (c) Lastly, if the amount of trainees considered is still below the value of $h$, then supplement based on random trainee selection as long as trainee demand is nonzero. Note that this random selection has no significant impact on performance.

Trainee selection thus inputs data to maximize the opportunity of the most efficient solution of scheduling standard courses. To prevent the model from scheduling standard courses at the beginning of the week and only having nonstandard courses, which require more resources, at the end of the week, a priority heuristic is added. The priority heuristic uses the fourth term of the objective function (Equation 5-1) to add a cost reward for the prioritized trainees based on due date information. Else, the most efficient courses are naturally rewarded. Also, a model setting could be selected by the user to prioritize overdue recurrent training over conversion courses. When applied, conversion courses do not necessarily start at the beginning of the week. The same is true when simulator capacity or instructor availability propagate into next week(s).

### Instructor Subset Selection

A similar approach of selection and prioritizing is applied to instructors. Again, the working rules, described in section 5-1, limit some instructors from being assigned.

1. Likewise, a list of non-eligible instructors is composed or updated based on availability of individuals. Those that are already assigned to a duty on that day, or non-training duty in the days prior to the time instance considered are deemed non eligible. The implementation uses availability dates, based on working rules and random blockage of schedules to account for non-training activities, of all crew members to draft this list. As an additional requirement when considering conversion courses, these availability rules are more restrictive. In that case, an instructor should have a consecutive instruction count of zero, which is applicable to those with a non-training duty in their previous schedules.
2. If a list of prioritized instructors of correct qualification is available, then select the amount up to $h$ instructors from this list in a random sense. As this list is based on instructors having a nonzero, non-maximum consecutive count of instruction in their schedule plus timing constraints specifying this only applies to the same simulator slot on the next day, it only rarely occurs that not all prioritized- and available instructors can be selected. Else, the random selection is used.
3. Then for each unique qualification required by any of the considered courses, update the delta between the amount of qualified instructors already under consideration and the parameter $h$. If a nonzero delta applies, then select delta amount of qualified, available instructors based on random selection.

The instructor selection uses both a priority and non-priority scheme. As mentioned, instructors on a consecutive instruction schedule are prioritized. This priority applies whenever the crew member is considered on within one 'step' of the time of day of the previously assigned simulator slot. Such a rule allows a step in slot either up or down on the next day and is common to limit rest time wastage. A more flexible approach would be to allow for days of other duties in between, but these are neglected in this thesis. Prioritizing consecutive instruction assignment comes forth from minimum rest time requirements consisting of a fixed and variable part. As mentioned in section 5-1, the annual amount of instruction is limited per instructor. However, as the legal training requirement is more important, the annual limit is incorporated using a penalty system. This means that as long as this limit is not exceeded, the model will assign instructors with any distribution. As soon as an instructor reaches this limit while other have not, the instructor is not actively considered until the balance is restored. This process mimics reality, in which large differences in the amount of instruction are not uncommon due to personal preferences.

At the end of each iteration, the availability of simulator capacity, instructors, trainees and the trainee demand is updated. For instructors not yet on a maximum-length streak, their availability is updated to the next day. For others, including those instructors that were prioritized, but unused, their availability is updated to a date and time $x$ days away. The $x$ represents a sample from an historical distribution that captures the length of non-training duties including mandatory time off. After the sampled length, the crew member is available for training again and also satisfied the requirement of having a balanced schedule in terms of flying and instructing.

## 5-4 Robust Crew Training Scheduling & Assignment using Feedback

The model and solution methodology described above are tailored to generating a cockpit crew training roster. This reference roster serves as a benchmark throughout the thesis. As an additional benefit, an adaptation of the same model and method can be used to generate a robust cockpit crew training schedule. The optimization revolves around the combination of deterministic roster cost and stochastic recovery cost, as already explained in the objective function in Equation 5-1. The course dependent, expected recovery cost are determined via a simulation framework - explained in chapter 6 and chapter 7 - and a feedback loop. The loop feeds back features on the disruption and roster state and links it with the associated recovery cost. As the features are key in what quality can be achieved, the selection thereof is comprehensively treated in subsection 5-4-1. This sections also explains the first of the two feedback loops: Neural Network (nonlinear) feedback. The NN regressor learns from training data to estimate expected recovery cost for a given input set of features. The model takes part of the input data for training purposes and used the remainder to test its accuracy. The measured accuracy is then used to select a good performing set of features. In order to test the value of nonlinear optimization of weights for the regressor, it is compared to the second type of feedback: proportional (linear) feedback in subsection 5-4-2. The former uses a predefined linear combination of weights, whilst the latter optimizes for the weights in a nonlinear fashion and learns from the input data.

## 5-4-1 Neural Network Feedback

In order to compare the added value of nonlinear weight optimization, both types of feedback loop use the same set of features. The selection of an appropriate set of features should thus work for both. As a consequence, the set of features that needs to be related to a single decision variable. The choice is made to add expected recovery cost based on course scheduling as this is the primary means of achieving robustness. Optimizing for assigned crew has no significant effect as disruption probabilities are generalized across groups rather than individuals. The course determines what qualification is required and also the balance of the schedule. Also note that an important test, as specified in chapter 4, is to test the model's ability to deal with dynamic effects. For this reason, the problem is varied per month. However this is not a feature, but as an independent variable instead. The neural network requires this to adjust the weights dynamically, but more on this in subsection 5-4-1. The remainder of this section described the feature selection process and the model settings that are used to obtain the results.

### Feature Selection

A comprehensive list of features was initially established. It consisted of time related features such as time of day, day of the week and month of the year. Other features are related to the qualification required for the disrupted course. This includes the minimum qualification required, a choice to schedule an overqualified instructor- or helpout and roster dependent features that indicate the amount of (un)assigned instructors (of a certain qualification) per day or week. Lastly, features are drafted that relate to the balance in the roster. The balance of types of courses scheduled per day and / or week, the balance of standard and nonstandard courses and the balance of recurrent and conversion courses. However, the full set of features would require too much training data to be accurately implemented. Even then, the value of such an extensive set could be questioned. Instead, the list is reduced in size by implementing only those features that are related to the dependent variables influencing recovery actions such as the usage of reserves. These variables become apparent in chapter 6 and chapter 7. The idea is that if these features can capture dependent variables of some recovery actions, the neural network might be able to learn how to benefit from the variance therein. As explained, the month of the year is added to capture the dynamic nature of the cockpit crew training scheduling problem. Rather than a feature, this is a dependent variable and determines the set of optimized synaptic weights to use for predictions. The short list of features used is:

- Time of day

- Day of the week
- Minimum instructor qualification required for the training activity
- Percentage of sufficiently qualified instructors that are not assigned to a training duty at the time of the disrupted training event
- Balance between standard and nonstandard courses on the day of the disrupted training event.
- Average training demand from the previous, current and next week normalized by the mean training demand over the course of a year.

Multiple combinations and forms of features have been tested. First of all, a set of eight features was input. However, the categorical variables (obtained via ordinal coding) proved difficult for the neural network to handle. As a next step, the amount of features was reduced to limit the amount of combinations and thus ease the training process. Afterwards, converting (some) categorical features to continuous ones should proved to increase performance. All changes made are listed and explained below:

- The features on unassigned instructors per qualification are linked to the feature on minimum qualification required and then combined into a single feature. The number of instructors not assigned to a training duty in the week around the disrupted training event, but qualified to recover the course are summed. This number is then divided by the total number of sufficiently qualified crew members to make it a percentage. A constant of one is added to this to obtain a value between 1 and 2. The logarithmic value is taken to exaggerate the lower region of the feature value. If very few crew are available to recover a disruption (i.e. do not have a fixed training schedule already), the recovery becomes very limited and thus likely to get expensive and vice versa. The logarithmic transformation should penalize the assignment of large numbers of unique instructors in a single week and ensure a good balance in the assignment of instructors.
- Secondly, the number of daily standard and nonstandard courses are combined into a single feature as well. They are subtracted from each other to indicate the balance in the daily assignment. As standard and nonstandard courses have different recovery processes, as will be explained in chapter 7, an optimal balance could be sought, also in combination with the other feature values.

### Neural Network Settings

Next to features, the neural network itself provides numerous options to improve on the results. The NN used a Rectified Linear Unit (ReLU) activation function and the maximum number of iterations of updating synaptic weights is set to 10,000. Third, the number of hidden layers is increased from one to two to capture any nonlinear relation between the features. A single layer proved insufficient to capture non-linearity. Also, the number of hidden neurons in each layer is increased from a value equal to the number of input features to 50. This should improve on the combinations of features and their respective synaptic weights.

The performance achieved with these settings and changes is displayed in Figure 5-2. The red line and red axis display the number of lines of training data per month of the year. As said, each month is input into a neural network separately to optimize for the synaptic weights. Each month is simulated for a fixed amount of disruption scenario's, but because of monthly varying disruption probabilities and training demand, the number of lines of training data varies also. Furthermore, two performance metrics are used: (1) Mean Absolute Percentage Error (MAPE) in orange, and (2) Root Mean Square Error Percentage (RMSEP) in blue. The MAPE indicates

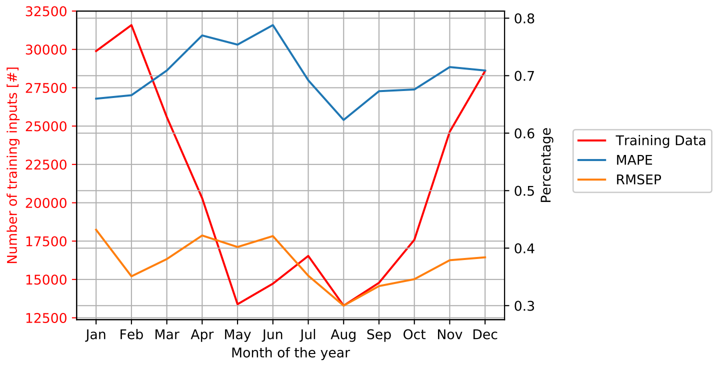

**Figure 5-2:** Neural Network performance metrics per month of the year

the average error of the prediction with respect to the actual output of the training data. The values are high because of the nature of the problem as the recovery cost can only have a few discrete cost levels and many combinations of features could lead to this same output value. Also, the same combination of features could yield multiple different output values. The neural network theoretically trains itself to estimate the mean expected recovery cost for every combination of feature values in a nonlinear fashion. The large values for the MAPE are thus unavoidable. The same holds true for the RMSEP indicating the residuals in percentage.

Instead of focusing on optimization of the neural network performance and results, the trained neural network is used in its current form. The set of weights and biases are extracted from the model and used to predict recovery cost in the TS&AM model in each iteration. The expected recovery cost are also added to the course scheduling decision variables based on the calculated feature values. In each iteration, the values of the features are re-computed. The time of day and day of the week are extracted from the time considered. The minimum qualification required is course dependent and the number of unassigned instructors continuously updated. The same holds for the daily number of standard and nonstandard courses. Depending on the month, a different set of synaptic weights and biases is used to make the predictions.

Lastly, areas of improvement are identified for future research to improve on the neural network performance. First and foremost, feature optimization could yield the largest gain in performance. Also, the amount of training data could be increased. The neural network trained well on 250,000 lines of input data, but splitting it over 12 months might be too restrictive. Although the performance displayed in Figure 5-2 does not directly indicate an inverse relationship between the number of lines of training data and the performance. The lack of this relationship could also be due to the unfitness of the performance measures in combination with the discrete nature of the training data. Furthermore, changing settings on the number of hidden layers and hidden neurons could be optimized. Due to knowledge- and time restrictions on this side, the area is neglected after a few runs based on trial and error. Lastly, introducing a leaky ReLu activation function might reduce the chance of synaptic weights to become zero. This could extract the last few bits of added performance. All in all, it is likely that performance of the neural network could be improved.

## 5-4-2 Proportional Feedback

The same training data, setup and features are used in the Proportional Feedback (PF) algorithm. Rather than going through the intensive process of learning from the data and optimising for synaptic weights and biases to regress in a nonlinear fashion, a straight forward linear, proportional weighing of independent cost levels is applied here. This allows to test if the more complex approach of a neural network learns patterns to fully exploit robustness measures. The lower complexity and arbitrary weighing are contrasted with the benefits of proportional feedback being simple and scalable. The approach will be explained in detail below.

Proportional feedback uses the assumption of independent features, which is a rough assumption for the features, especially the features on minimum qualification required and percentage of unassigned instructors. The same set of 250,000 lines of training data is used to extract, per possible value of the feature, an expected recovery cost. As an example, there are five simulator slots per day, meaning the feature can take on one of five discrete values. For each of these values, all training cases matching the value are filtered and used to compute the mean simulated recovery cost. The same is done for the other features. Note that some are continuous. These features are discretised into a set of intervals to ensure sufficient data is available for a reliable computation of expected recovery cost. As an example, the percentage of unassigned, sufficiently qualified instructors, is discretised into ten intervals of ten percent each.

Then by assumption, the 'independent' expected recovery cost are averaged and scaled by the mean training demand over the three week period consisting of the current, previous and next week. This is visualised in Equation 5-13. Again, the computed expected recovery cost is pre-multiplied with the course- and time dependent disruption probability in the TS&AM's objective function (Equation 5-1) to obtain accurate cost levels per training event. Here, $c_t$ represents the recovery cost for the time of day and $c_d$ distinguished in day of the week. Parameter $c_q$ indicates the expected recovery cost depending on the qualification required for course $k$ and $c_{ui}$ represents the expected recovery cost associated with the percentage of unassigned instructors. Lastly, $c_{sn}$ represents the expected recovery cost for the balance between standard and nonstandard daily courses. The demand is captured in the $d_c$ for the current week (i.e. the week of which the schedule is being constructed), $d_p$ for the previous week and $d_n$ for the next week's expected demand. This demand is averaged to scale the different cost levels introduced by the various features. Taking the average is again an arbitrary choice that proved to work by trial and error. A low average three week demand with respect to the annual mean, yields a value between zero and one in the denominator of Equation 5-13. The differences in expected recovery cost levels are then exaggerated. This provides an incentive to opt for the most robust solution. Vice versa, if demand is high, the differences in cost levels are reduced. in combination with the Priority Heuristic (PH), this might lead to a more efficient solution in terms of scheduling alone. The part in the objective function that focuses on robustness is then compressed. Note that this scaling function has no guarantee of being optimal.

$$\mathbb{E}[c_{recovery_{kt}}] = \frac{(c_t + c_d + c_q + c_{ui} + c_{sn})}{5} \cdot \frac{3}{(d_c + d_p + d_n)} \tag{5-13}$$

## 5-5 Model Output

The TS&AM model primarily outputs a reference or robust cockpit crew training roster. The roster is kept in a data format to ease the process of inputting it into the Primary Disruption Generator (PDG) and Rule-Based Recovery (RBR) model. Table 5-5 shows an example of the roster. Each line represents a trainee assigned to a certain course at a specified date and time. Associated to this is the session ID, which connects the rosters of trainees assigned to the same session. Multiple day-long courses have multiple entries for the same trainee and same session ID, but with different dates (and times). The combination of session ID and ranks gives away the nature of the schedule course. When two trainees of complementary rank are assigned, the session is 'standard' and no helpout is scheduled. When two entries with the same session ID, date, time and rank exists, the session is 'nonstandard' and a helpout is assigned. The same holds true for cases in which only a single trainee is assigned. Each session has an instructor assigned to it by default, and the last column specifies the due date of the recurrent training event.

**Table 5-5:** Example of an output roster

| Activity | Crew ID | Rank | Date time | Session ID | Instructor | Helpout | Due Date |
|---|---|---|---|---|---|---|---|
| C1 | 645872 | CP | 2018-01-01 09:45 | 1 | 771485 | - | - |
| C1 | 134921 | FO | 2018-01-01 09:45 | 1 | 771485 | - | - |
| R1 | 547892 | CP | 2018-01-02 06:15 | 2 | 621845 | 249792 | 2018-01-31 |
| C1 | 645872 | CP | 2018-01-03 09:45 | 1 | 771485 | - | - |
| C1 | 134921 | FO | 2018-01-03 09:45 | 1 | 771485 | - | - |

The roster is the primary output of the TS&AM, but associated to this is data that can also be extracted for the sole purpose of analysing the roster and its performance. Examples of this are: (1) a list of propagated training demand per week, and (2) a list of instructor utility rates and balance between instructing and flying. These are examples of underlying metrics that could asses different aspects of the roster other than pure cost.

## 5-6 Validation and Discussion

The requirement on the TS&AM for this thesis is limited to generating a feasible cockpit crew training schedule with a method that can be applied to both non-robust and robust scheduling and solves within limited computational time. The validation process consists of two parts:

1. Qualitative comparison of disruption data against that of the European legacy carrier.
2. Judgement of an operational expert solving disruptions on a daily basis based on provided information.

From the validation process can be concluded that all the output schedule of the TS&AM is sufficiently accurate. For a problem involving 600+ crew members, 160+ conversion courses, 2,900 recurrent courses spread across 2 simulators each having 5 slots per day, solves in approximately 5 minutes for an annual schedule. This means an average of below 6 seconds per week. The run time requirement is thus satisfied. As a down side, the TS&AM model does not schedule all courses. This means that the generated roster is infeasible in reality. There primary reason for this is the airline uses much shorter yardsticks for conversion training. The yardstick lengths used in this thesis are obtained from an European airline, yet these are on average 75 percent longer than what is actually scheduled by the same airline. This difference is beyond the variation caused by yardstick length differences due to crew composition and previous experience. Rerunning the TS&AM with the shorter yardstick length, 97 percent of all courses of the airline schedule are assigned. The remaining difference is probably explained by the restrictive assignment rules and the assumptions made in this thesis. Nonetheless, the objective is to generate a reference roster and robust roster using the same method, which is done with model using the original yardsticks.

Due to the setup of the Construction Heuristic, legal requirements on course scheduling are implicitly enforced by means of having to schedule a course whenever possible. The downside is that an unfeasible roster is obtained from a reality perspective. The upside is that a roster is still output. The mismatch only exposes an area of improvement in the algorithm. Despite the vast reduction in run time, drafting the CH proved difficult for the complex training scheduling problem. The assignment rules are often time dependent, both backward looking and forward looking. Especially the latter proved difficult to enforce in a construction heuristic. Also, the rules applied in the Selection Heuristic directly impact the solution, both in terms of the actual schedule and also the efficiency. Arbitrary rewards are also enforced in the Priority Heuristic, which has a similar impact. In all, the model is static and it proved difficult to capture all working rules (in detail) in such a model.

As mentioned earlier, the feedback algorithms could be improved by doing further research. This involves research into a better function for the Proportional Feedback algorithm. Similar, the NN features could be improved to gain performance. Feature selection is an area of research in itself and is thus explored in a limited sense in this thesis. The same applies to optimizing settings and the amount of training data of the neural network to gain performance. This is limited because of time constraints of the thesis. Finally, the model currently adds course dependent expected recovery cost only. Doing the same on the other decision variables such as nonstandard course scheduling, instructor and trainee assignment variables might further optimize the totality of the robust schedule. The difficulty of this lies in the dependency of all decision variables and features.

A last area of improvement is that of combining the Construction Heuristic with the feedback algorithm. The value of the CH lies in speed and its compromise on solution quality proved little for the non-robust roster (disregarding mismatches in rules). The little cost differentiation between scheduling certain courses, trainees and instructors proved to significantly boost solution speed without impacting solution quality as much. Despite an LP being an efficient method in cutting the search space, it still considered too many options with the same cost levels. Also, the many combinations of timing, course, trainee(s) and instructor(s) proved too slow as concluded by Holm (2008). However, when combined with the feedback algorithm, the model does have different cost levels. Now, the Selection Heuristic has a(n) (higher) impact on the cost of the schedule. Applying a LP model to globally optimize the robust training schedule is more beneficial than locally optimizing the schedule for each individual simulator slot. Future research could be commenced to quantify the added benefit of global optimization.

# Chapter 6

# Disruption Generator

Robustness of the generated training schedules is tested using a simulation framework. The first part of this consists of a discrete event generator. According to Van Den Bergh et al. (2013) discrete event simulation allows for a complex, realistic set-up of the model. A data-driven approach should maximize the similarity between the model and reality. A second benefit is that robustness is tested in a stochastic environment with a high number of repetitions leading to a higher accuracy. This is important for practical aspects as robustness is difficult to prove. In section 6-1 is explained how data of the European legacy carrier is obtained and section 6-2 describes how this is input into the disruption generator. The output of the model, elaborated on in section 6-3 is used in a validation with the European legacy carrier in section 6-4. This section is closed with a discussion.

## 6-1 Model Input

The primary disruption generator relies on several assumptions. First of all, only primary disruptions are considered. It proved impossible from historical data to distinguish between primary and secondary disruptions for any category other than illness and leave. This is important as modelling secondary disruptions as primary ones could blur the realistic nature of the model. Also, it would require modelling dependencies such as disruptions in the flight schedule that would impact the availability of resources for training, making the model too large and complex to solve in reasonable time. Thus only illness and leave are considered as these are primary disruptions by definition. Secondly, the model uses independent and identically distributed samples to generate illnesses and leave. Both illness and leave are also assumed to be independent with respect to each other. As a result, illness during leave or leave during illness are precluded. Another important restriction is to focus on 'short lasting' illness alone. Leave, such as parental, marriage, funeral and care related leave, itself is a short lasting disruption by default. In general, these are marked by (multiple periods of) at most two days. This falls within the definition of short lasting. In this case defined as a maximum period of 14 calendar days, which corresponds to the publication period of the schedule. If prolonged illness is known prior to publication, then that crew member is excluded from the scheduling process.

Under the assumptions, a data-driven approach to modelling disruptions is established. For confidentiality reasons, all probabilities and density functions are normalised with respect to the overall mean of all data displayed in the concerning plot. In this way, the variance and difference in order of magnitude is maintained. From analysis of four years of historical, fleet wide data, and as argued by Barmby (2002), illness proved to be dynamic. Dynamic probabilities are added to test the model's capability to deal with changes. Figure 6-1 shows the trend of normalised illness and leave disruption probabilities per crew member per day discretised per month. The implementation is arranged via a look-up table. A look-up table is tractable because the amount of data obstructs a higher resolution, but it is sufficient to prove that significant differences in illness disruption probabilities exist: (1) between non-training and training ($P < 0.0005$) and (2) between instructors and trainees ($P < 0.0005$). These results are obtained via a paired, two-tailed t-test. Paired because of the pool of unique crew is the same for all samples, only the role during the illness differs. The difference between non-training and training is explained by stricter health requirements for flight duties than for ground duties. The difference between instructors and trainees is harder to explain, but could be related to workload or pressure while being trained.

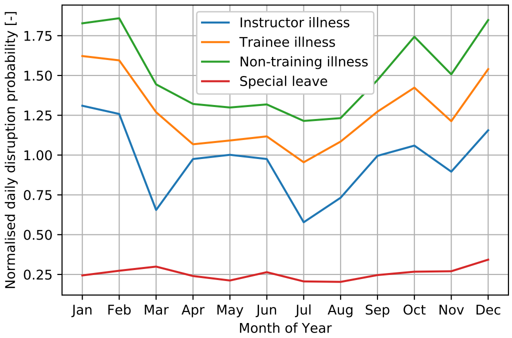

**Figure 6-1:** Normalised, empirical, dynamic disruption probability due to illness for non-training, instructors and trainees and leave

Illness is also dynamic in terms of the length of the illness. This effect is captured in the model input data by means of discretised 'blocks'. These blocks arise from a practical point of view: in case of illness, the airline clears and recovers the roster for a couple of days (i.e. a block) and wait for confirmation or refution on extended illness before undertaking action again. If the crew member reports fit for duty before the end of the expected illness period, the crew member is used as a standby to cover disruptions. Again, four years of historical, fleet wide data is analysed to find the empirical distributions of the conditional illness blocks up to two extensions. Additional extensions rarely occur and are mostly related to long-term illness. These can thus be neglected. Figure 6-2 shows the initial empirical probability density function of illness length and leave. As the distribution is used to sample lengths of events, a zero-day block is excluded from the options. As can be seen, the initial illness block is mostly three or four days long. Also, the length of leave is displayed, but these disruptions cannot be extended by assumption.

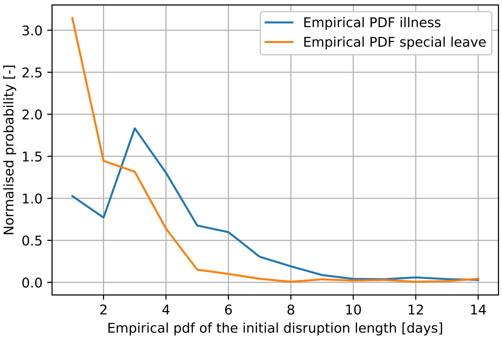

**Figure 6-2:** Normalised, empirical probability density function of the initial length of illness and leave

Figure 6-3 and Figure 6-4 show the conditional, empirical illness length distributions of the first- and second stage extensions respectively. The first-stage extension distribution shows on the x-axis the possible illness lengths of the initial block. The bar indicates the extension probability conditioned to the initial illness block length. If extended, the colours in the bar make up the pdf of the extension length. Darker colours show a longer extension block length. As can be seen, a dependency exist between block lengths of the initial and first-extension. A short initial block is more likely to be followed by another short block, than it is to be followed by a longer block. The opposite applies to long initial illness blocks. Figure 6-4 shows a similar distribution, but has only nine options on the x-axis. This is because historical data has shown that the options above nine days in a second stage extension hardly ever occur.

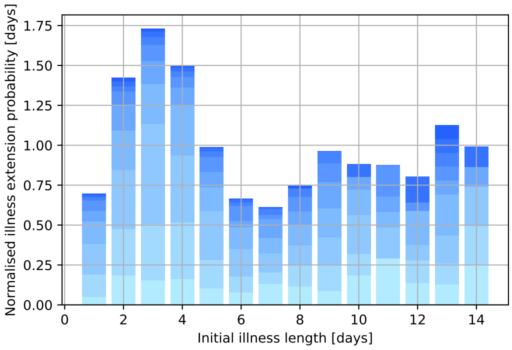

**Figure 6-3:** Conditional, empirical distribution of the extended illness block length

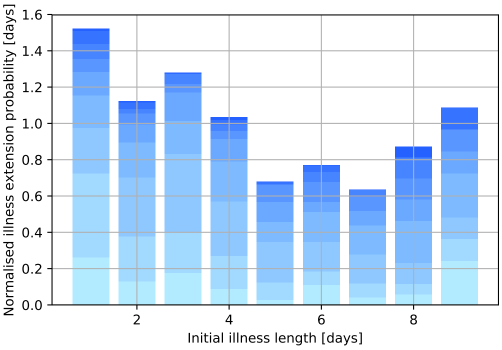

**Figure 6-4:** Normalised, conditional, empirical distribution of the second-stage extended illness block length

In each non-extended illness- or leave block, an empirical distribution applies to determine if and when a crew member reports for duty again prior to the estimated block length. As indicated in Figure 6-5, more than 80 percent of the probability lies beyond the zero-day mark, indicating a block length estimation error. In each of these cases, the roster of the crew member was unnecessarily emptied. However, these resources can be used again for recovery as will be explained in chapter 7. Finally, Figure 6-5 also displays notification timing of the disruptions itself. Around 75 percent of illnesses and 50 percent of leaves will get known at the day of operation, which is logical. However, some illnesses start during mandatory rest, holiday or other duties. Such illnesses are propagated to the training events, but are known earlier. A similar process applies for leave. This notification timing will also come to play in the recovery process described in chapter 7.

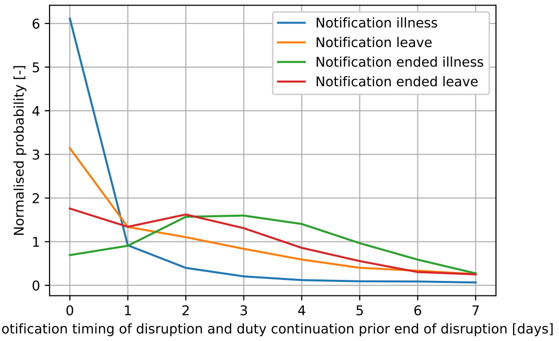

**Figure 6-5:** Empirical pdf of the notification timing of disruptions and duty continuation

## 6-2 Primary Disruption Generator Model

The probability density functions derived from historical data are input into the Monte-Carlo Simulation of primary disruptions. The algorithmic framework displayed in Figure 6-6 converts the data to a set of primary disruptions. The algorithm will generate disruptions for each crew member involved in the middle two weeks of a monthly schedule, as explained in section 4-2. The set of crew members to simulate for is reduced for the purpose of speed as, in general, only ten percent of all crew members are involved with training in these weeks. This neglects other illnesses and leaves, but the only impact is on the list of standby crew originating from disruptions. Because standby crew could also be used to cover other flights again, the impact of this decision is estimated not to outweigh the (90 percent) run time benefit across the large amount of repetitions. Note that for each crew member in the subset, the full four weeks are covered in the simulation to account for carry-in and carry-out of disruptions, as introduced by Nissen and Haase (2006). The carry-in and carry-out is essential to keep the balance in number of disruptions considered for each session.

Thus, after selecting a single crew member, the algorithm iterates over the full 28 days involved. For each day, the daily disruption probability is captured in a moving average based on the personal roster of the next x days. In this case, x is set to seven to correspond to the average illness length. Figure 6-1 is used to account for the significant differences per category. The first distinction is based on the non-training and training duties within the period, and dates are specified to determine which monthly average to take into account. The second distinction applied is that based on role of the training duty. The updated probability of illness and leave are then summed to generate a disruption decision for each day with minimal overlap. Days on which a disruption already exists are blocked for the purpose of efficiency of the algorithm.

Then for each disrupted day, a decision is made on the cause based on relative probabilities between illness and leave. For leave, the predefined density functions are used to generate length, notification timing on disruption and a duty continuation date are generated. When the disruption is caused by illness, a three-step process is started because of potential extension of the illness. At first, an initial length and notification timing are generated. Afterwards, a random yes-no decision is generated conditioned on the initial illness length to extend illness as explained in section 6-1. If extended, then a second length is generated and the notification timing is set one day prior to the end of the previous block length to match airline policy. If the illness is not extended, the duty continuation date is determined, which could be prior to the estimated end. The notification date is again set to one day prior to the duty continuation date by default. If the illness is extended, the entire process is repeated for a second-stage illness extension. For both illness and leave, the probability density function of disruption notification and duty continuation are proportionally scaled based on previous notification timing. such that they are chronologically sound and ordered.

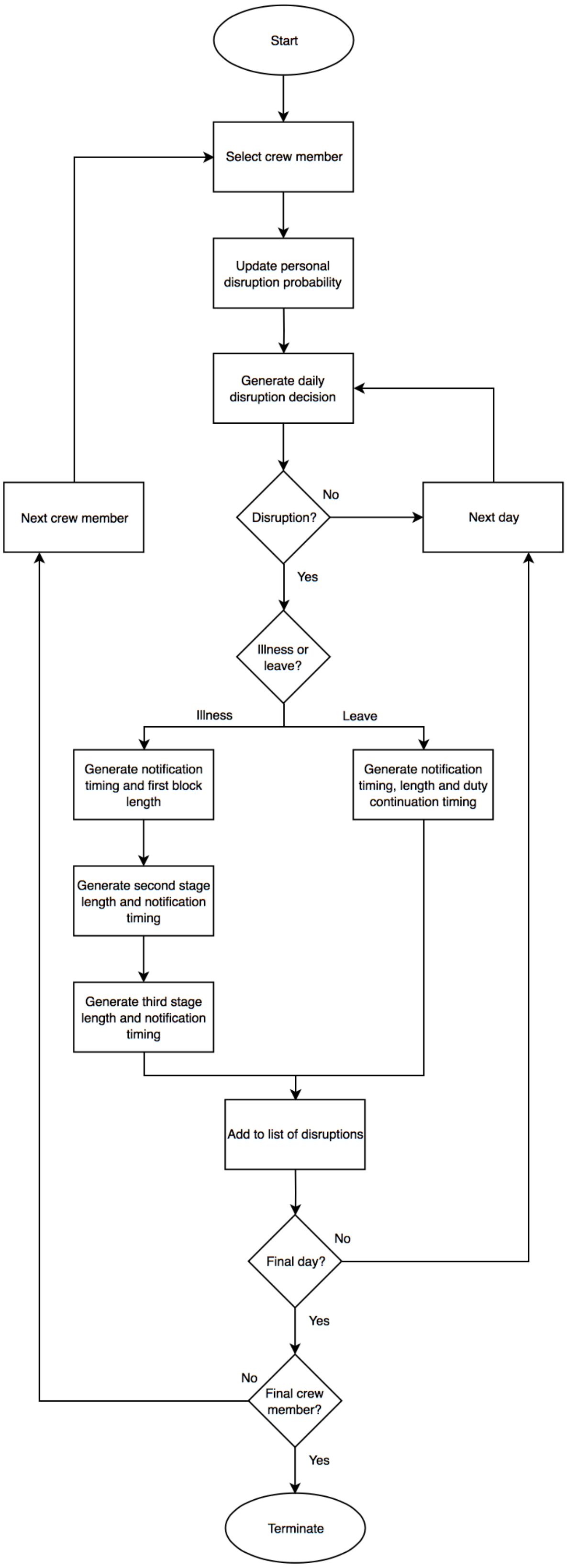
Start
Select crew member
Update personal disruption probability
Generate daily disruption decision
Next crew member
Disruption?
No
Next day
Yes
Illness or leave?
Illness
Leave
Generate notification timing and first block length
Generate notification timing, length and duty continuation timing
Generate second stage length and notification timing
Generate third stage length and notification timing
Add to list of disruptions
Final day?
No
Yes
No
Final crew member?
Yes
Terminate

**Figure 6-6:** Algorithmic framework of the Primary Disruption Generator

## 6-3 Model Output

The algorithm builds a list of disruptions before it terminates when all days in the roster of all crew members are considered. The list, of which an example is provided in Table 6-1, contains all information on crew identification, disruption category, start date, end date, block lengths (displayed in three separate columns), disruption notification timing (abbreviated as noted) and duty continuation date (abbreviated as continued).

**Table 6-1:** Example of primary disruption data

| Crew ID | Type | Start | Estimated End | BL 1 | BL 2 | BL 3 | Noted | Continued |
|---|---|---|---|---|---|---|---|---|
| 771485 | illness | 2018-01-11 | 2018-01-16 | 2 | 3 | 0 | 2018-01-08 | 2018-01-14 |
| 645872 | leave | 2018-01-13 | 2018-01-19 | 5 | 0 | 0 | 2018-01-13 | 2018-01-19 |

This list is then converted to two things: (1) a list of disrupted training events, and (2) a list of disruptions outside any training duty. For each training event within the estimated disruption period, data on crew identification, activity code, role, date- and time, session identification and known since date are listed, as can be seen in Table 6-2. The known since date is determined based on the available data. If the training event falls within the first block, the notification date is already known. Else, the notification date is set at one day prior to the start of the block in which the training event is scheduled.

**Table 6-2:** Example of training disruption data

| Crew ID | Activity Code | Role | Date and Time | Session ID | Known Since |
|---|---|---|---|---|---|
| 771485 | R3 | instructor | 2018-01-13 | 12 | 2018-01-12 |
| 645872 | C | trainee | 2018-01-16 | 67 | 2018-01-13 |

Any day within the time window between duty continuation and estimated end of the disruption is listed as standby duty. By default, these days are listed one day prior to the duty continuation date. As it concerns a short period if all well, these days cannot be used for anything other than standby duties. This list is updated for use in the recovery model. The standby list, of which an example is provided in Table 6-3, contains information on crew identification, rank, highest qualification, date and known since date. Each day appears as a separate column.

**Table 6-3:** Example of standby list data

| Crew ID | Rank | Qualification | Date | Known Since |
|---|---|---|---|---|
| 771485 | CP | TRE | 2018-01-14 | 2018-01-13 |
| 771485 | CP | TRE | 2018-01-15 | 2018-01-13 |

## 6-4 Validation and Discussion

Only information on training disruptions and recent standby crew is useful for testing and proving robustness in the current model set-up. All other information is neglected. The training disruptions are validated against the requirements. The validation consists of two parts:

1. Qualitative comparison of disruption data against that of the European legacy carrier.
2. Judgement of an operational expert solving disruptions on a daily basis based on provided information.

The conclusion of the validation process are that the results are sufficiently accurate for the objective of proving robustness of the cockpit crew training schedule. Especially when applying the model to the reference schedule as well and including disruption probability as part of the sensitivity analysis in chapter 9. Furthermore, the available information is modelled extensively and accurately matches the results of the European legacy carrier.

Still, performance could be improved by adding more sources of disruptions such as propagation of disruptions between flights and training and vice versa. Another category could be a change in crew demand, requiring training events to be rescheduled. To accurately model these disruption categories, a more detailed interdependency with the flight schedule should be created. Such a link would also increase the accuracy of the sampling of notification dates. Now that is random whilst in reality it would be dependent on previous schedules.

Another area of improvement is to model secondary disruptions as well. An example would be to model the possibility of someone reporting better, but becoming ill again a day after. Although the model allows for this, it hardly occurs due to the low disruption probabilities. in reality, there might be a dependence between several illnesses that is not modelled in the current implementation. The same holds true for outbreaks of illness. This is somewhat captured in the seasonal trend, but crew coming into contact with each other could amplify the spreading of health complaints.

The DG takes approximately 0.5 seconds to obtain a list of disruptions for a week. As proving robustness is a process of repeated simulation, the run time is considered large, but reasonable. A thousand repetitions, yielding between five and 10 thousand disruptions takes about 10 minutes of run time. This is reasonable because the main focus of this research is to prove robustness, not to prove it in the fastest manner. On the plus side, the DG models the stochastic nature of all aspects of illness and leave accurately. Stochastic length, discretised blocks, notification timing and duty continuation are all taken into account. On top, any mismatch in clearing the roster and a crew member reporting for duty is taken into account for the purpose of a realistic recovery model.

# Chapter 7

# Training Schedule Recovery

The second part of the simulation framework to test schedule robustness is the recovery model. The repetitive nature of the simulation framework requires a fast recovery model, which is found in rule-based methods according to Ionescu et al. (2010) and Bayliss et al. (2017). They showed that, for simple scenarios, a run time of 0.1 seconds is not uncommon. Another argument is that a rule-based method works well in a data-driven environment. In turn, this increases accuracy of the model with respect to reality. Airline processes account for condition checks before moving to a next potential recovery action. However, airlines do not necessarily stick to the fixed order. A last benefit is that rule-based methods allow to recover with minimum amount of changes rather as opposed to re-optimization models. The latter, however, would solve to optimality while also enforcing constraints on the number of changes applied. As optimality of recovery is not a hard requirement to meet the research objective, a faster rule-based method is opted for. The chapter starts with a description on the input and assumptions of this model in section 7-1. The rule-based algorithm itself is explained in section 7-2. The output and validation plus discussion are treated in section 7-3 and section 7-4 respectively.

## 7-1 Model Input

The input of the recovery model consists of the (generated) roster along with the input data of the training scheduling & assignment problem itself. This includes data on crew members and their qualifications and the training yardsticks. The output generated by the Disruption Generator (DG) is also used as input by the recovery model. This is comprised of the disrupted training events and the standby list generated by other disruptions, as described in section 6-3.

As the recovery model is data-driven, additional input is needed. The input is obtained from analysing four years of historical data on recovery actions. The availability of recovery options is dependent on time variables and roster status. This is what makes recovery a stochastic process. The model captures the randomness via time dependent probability density functions on loss days, Premium Days (PD), standby crew, reserve crew, swappable pairings and counter values. Loss is defined as a roster day in which no activity is assigned. Such days can be reclaimed if a duty opens up and fits within the existing crew schedule. A premium day is a compensation for additional- or higher valued duties. Redemption of premium days make for a more efficient schedule. Both PD redemption and loss days have a probability that is dependent on the month of year and are close to zero. Nevertheless, both are added for the sake of completeness. This is because the

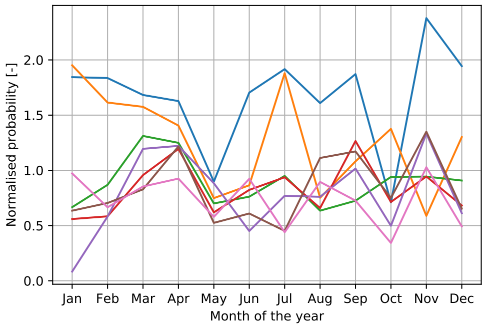

**Figure 7-1:** Empirical distributions on reserve availability per month and day of the week

European fleet is an efficient one in terms of crew scheduling and flexibility with almost zero loss. However, the intercontinental network is less flexible and has thus more loss. When applying the model to the intercontinental fleet instead, this recovery options will gain ground. Next, reserve availability depends on the simulator slot. A large difference exit because of starting times of duties and the coverage of these by reserve crew. This discrepancy is also something to exploit in the robustness scheduling model. Next, a swappable pairing is defined as one that starts- and ends on the crew base and on the same day. This definition is similar to that of Shebalov and Klabjan (2006) and make for swappable duties without impacting the remainder of the crew schedules. The probability is rather constant over the course of a year, but for consistency reasons and testing the dynamic behavior of the model, a monthly dependency is added. Lastly, counters are applied to log compensation for additional or higher values duties. Specific rules apply to reduction of these counters. The counters allow the airline to shift supply of crew in time to cover seasonal trends. This means that counter values depend on the month of the year.

The availability of reserves not only depends on time of day, but also on day of the week and month of the year as displayed in Figure 7-1. These normalised empirical distributions show the probability of having a reserve available for training. It is corrected for reserves used to cover disrupted flights and other disruptions that impact the demand of reserve crew. Noticeable differences can be observed in the availability of reserves per day of the week, which are represented by the lines. Although inexplicable, it can be exploited for the sake of robustness.

Lastly, an empirical distribution is input to specify counters. Similarly to reserves, the redemption of premium days requires the tally to be nonzero, which is varied by month of the year as well. Only probabilities for nonzero counters are displayed in Figure 7-2. Each bar displays the probability per month of the year and the darkness of the colour indicates the probability per height of the counter, which is undisclosed for confidentiality reasons. Higher values have such low probability of occurrence that they are neglected. It can be seen that counters are generally higher valued during the summer period because of higher demand for crew. The counters are used to increase production in the summer period, and compensate for this in the winter period. However, the probability of having a nonzero counter in the winter period is not much lower than for the summer period, indicating that full reduction of counters is difficult.

## 7-2 Rule-Based Recovery Model

Recovery options can already be derived from the input data. Each of these is displayed in the algorithmic framework in Figure 7-3. For clarity purposes, only a single branch is displayed. A branch exists per role (i.e. trainee, instructor or helpout) and assigned crew composition (standard or nonstandard) because different requirements or conditions apply to each recovery option. The requirements will be explained in detail in this section.

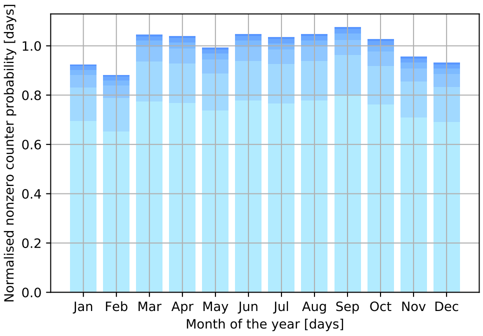

**Figure 7-2:** Empirical distributions on nonzero counters and their respective values per month

But first, the key principle of the algorithm is explained. A single disruption is selected to recover at each iteration. Under the assumption that no recovery action can be hold back, as explained in section 4-3, the possible recovery actions are ordered based on cost. Whenever a recovery option is selected, it is the most cost efficient by default and the model moves on to the next disruption. When all disruptions are solved, the algorithm is terminated. The timing in which the disruptions are solved is determined based on rules and makes that disruptions can be solved sequentially without having to simulate a time dependence. The algorithm acts whenever the disruption is noted or, when noted earlier, if the three day mark before the day of operation is passed. The disruptions are then solved chronologically.

**Loss day reclamation (RA1):** Loss days arising from roster inefficiencies can be reclaimed by the airline without cost when done at least a certain amount of time in advance. For the test case, this limit is set at a full day. Whenever a crew member with a 'scheduled' loss day is available, further analysis is conducted. A disrupted instructor needs to be replacement by another instructor, meaning that qualification requirements apply. The implementation is arranged via a random sample being compared to the probability consisting of the loss day probability per month being multiplied by the chance of having a nonzero counter times the potential crew members. The latter means a correction is done based on the roster status. The total pool of instructors is corrected for assigned duties. Note that due to probability levels introduced in section 7-1, reclamation of loss days have a very low probability and could righteously be neglected. Also note that reclamation of loss days for trainees is less accepted due to mandatory preparation time needed before any training session.

**Standby crew (RA2):** If a loss day cannot be reclaimed, the airline looks for a standby crew member. Crew members impacted by other disruptions are listed as standby to cover open duties on the day of their original duty that is disrupted. Using standby crew eliminates waste as the salary cost of that (otherwise wasted) crew member is already covered. Moreover, because standby duties often do not cover full flight pairing lengths, standby crew are favoured over reserve crew to cover shorter disrupted duties such as training. As a requirement, the crew member must be sufficiently qualified. In case of a disrupted instructor on a standard session, the standby crew member must satisfy qualification requirements as posed in the yardsticks. When an instructor on a nonstandard session gets disrupted, the qualification requirements depend on the qualifications of the helpout instructor as well. A helpout requires a minimal instructor qualification, which might introduce more options in terms of feasible standby crew.

**Reserve crew (RA3):** The same requirements and conditions are enforced for reserve crew, which are considered whenever a qualified standby crew member is unavailable. The difference is that reserve crew must be accounted for, adding salary cost to the operation of crew. As an approximation of reserve crew cost for training, each usage of reserve crew incurs a fixed daily salary cost. Additionally, the start- and end time of the duty to cover should fit within the reserve duty of the crew member. Historical data is used to compute probabilities of having reserve crew

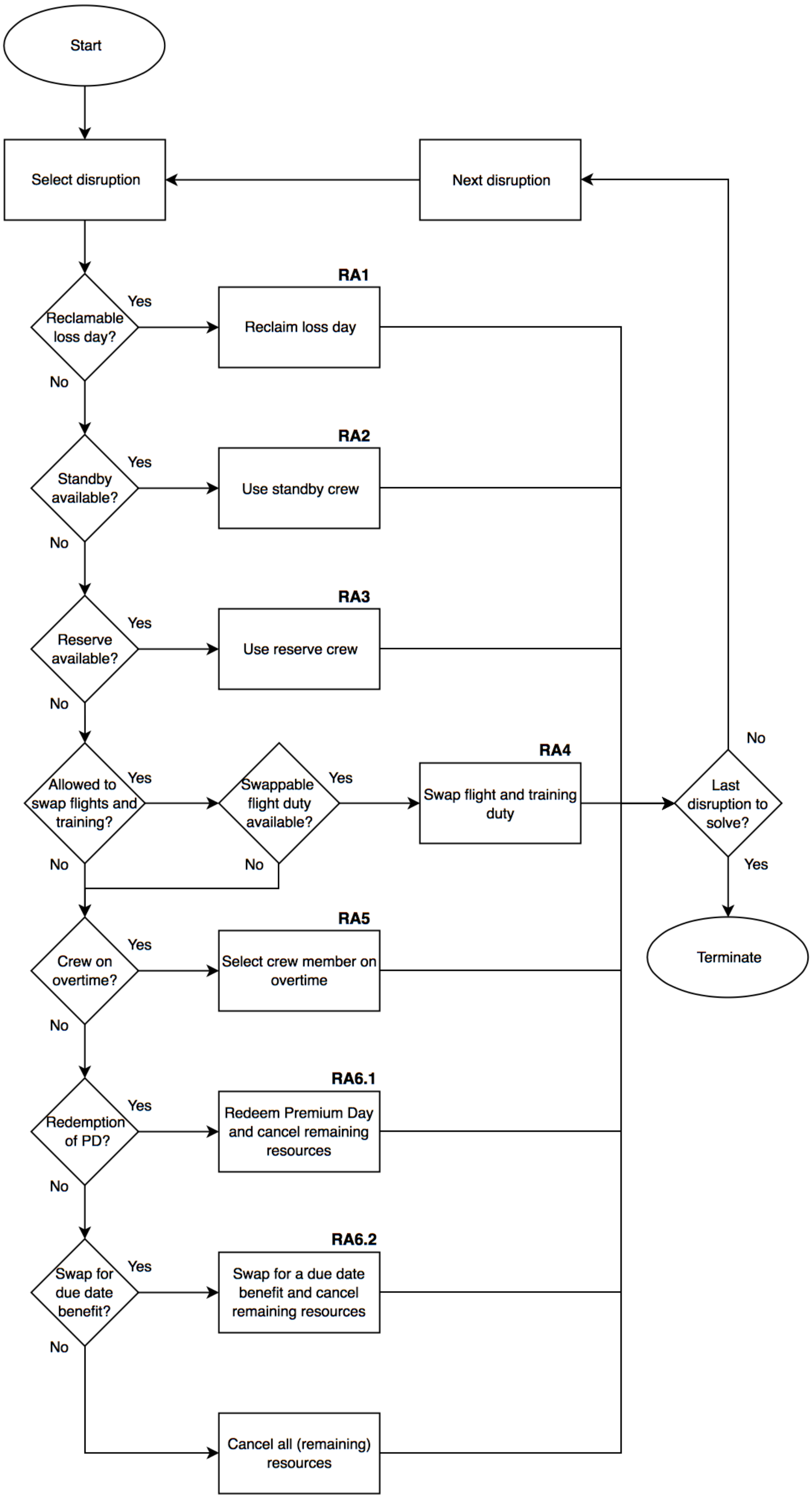
Start
Select disruption
Next disruption
RA1
Reclamable loss day?
Yes
No
Reclaim loss day
RA2
Standby available?
Yes
No
Use standby crew
RA3
Reserve available?
Yes
No
Use reserve crew
Allowed to swap flights and training?
Yes
No
Swappable flight duty available?
Yes
No
RA4
Swap flight and training duty
Last disruption to solve?
No
Yes
Terminate
RA5
Crew on overtime?
Yes
No
Select crew member on overtime
RA6.1
Redemption of PD?
Yes
No
Redeem Premium Day and cancel remaining resources
RA6.2
Swap for due date benefit?
Yes
No
Swap for a due date benefit and cancel remaining resources
Cancel all (remaining) resources

**Figure 7-3:** Algorithmic framework of the Rule-Based Recovery model

available to cover a disrupted simulator training event starting at that specific time of day. This is multiplied with the reserve availability probability, indicating the probability that a reserve is available at that day of the week in that month, as indicated by Figure 7-1. Note that it is corrected for the number of instructors already assigned to a training duty in the days around the disruption and that the computation relies on the assumption of independence between the dependent variables. In turn, the probability is multiplied by a probability that the available reserve crew member is sufficiently qualified, which is dependent on roster state. Again, random sampling is used to determine if a reserve crew member is available or not. This in contrast to standby crew, which is purely based on previous disruptions. Due to the random nature of the DG, this is also a stochastic variable.

**Swapping (splitted) pairings and training (RA4):** Another way of introducing robustness to a training schedule is via swaps. Swaps can acts a special case of reserve crew: an instructor is assigned to a 'swappable pairing' to contribute to the instruction-flying balance, but when needed can be swapped with a non-instructor reserve crew member. As a result, swapping crew between flights and training also incurs a fixed daily salary cost. A trade-off can thus be made on the balance of instructing and flying based on the pairing features. These features are dependent on the type of network. A short-haul network consists of pairings covering multiple flight legs with a varying number of operations to- and from the base airport. A single day training program can be covered by an instructor assigned to a pairing, that on that specific day, starts- and ends at the crew base without impacting the remainder of the schedule. However, it requires a non-instructor stand-by or reserve to be available with the same rank as the swappable instructor to satisfy complementary rank requirements for flights. The same can be done for multiple long training programs, which requires a larger part, if not all, of the pairing the be swappable. The latter could also be applied to the intercontinental network. However, effectiveness is expected to be low as all pairings need to be swapped in totality, which rarely is considered efficient.

Swapping flights and training is implemented via random sampling based on roster features. The probability of having a swappable pairing on the day of interest is multiplied with the probability of having a sufficiently qualified instructor assigned to it. The latter is again adjusted for instructors assigned to other duties in the time window covering the disruption. It is then checked against the availability of a non-instructor reserve crew member by the method introduced above. This process applies to standard training sessions in which the instructor is disrupted. Whenever the trainee is disrupted, a helpout instructor is sought to let the session continue as nonstandard. This requires (one of) the swappable pairing(s) to be assigned to any instructor. A disrupted instructor on a nonstandard sessions requires a qualification based on the capabilities of the remaining instructor as well.

**Overtime (RA5):** If none of the recovery options treated above are successful, crew members in their rest period are contacted for overtime. Next to the salary for that day, the crew member is compensated with an additional day off, known as a Premium Day (PD). Therefore, overtime is assumed to incur twice the daily salary of a crew member. Overtime is short-term recovery method that covers the disruption now at high expense. If there is leeway in the schedule in upcoming weeks, the negative effects of additional compensation due to overtime can be dealt with without creating additional disruptions. The model does not check schedule flexibility in future weeks as this is not part of the standard process. Instead, a historical probability on overtime is used to simulate the process of using crew on overtime. Again, there is a dependency on roster status, but due to inadequate data this cannot be implemented.

**Cancellation (RA6):** When unrecoverable, the undisrupted resources are cancelled. This means that resources such as the simulator, other trainee(s) and / or other instructor(s) are wasted. This incurred cost is added as a new simulator sessions needs to be scheduled that again consumes resources. However, there are ways to minimize the impact as displayed within the dashed box in Figure 7-3. If a simulator training session is cancelled, the original instructor is put on standby for that day. The standby often remains unused as can be explained by a simple calculation. Assuming independence of disruptions, an overall training disruption probability of 0.10 translates into an occurrence of a double daily training disruption on a single day of every 10 to 20 days as calculated via binomial distribution (Devore, 1999). The former applies to having six simulator training events per day and the latter to having only four. Converting cancellations into standby duties is thus a waste of resources in 90 to 95 percent of the cases. A more consistent focus on this aspect might yield more effective recovery options in the following ways:

1. **Rostering of premium days (RA6.1):** An instructor or trainee that has become available can be assigned a day off for the purpose of reducing their accumulated amount of premium days or deferred rest period. This reduces the net expense of the cancellation. As a requirement, the disruptions must be known at least one day in advance. A list is drafted of the other daily simulator sessions of the same activity for trainee swaps. The same qualification requirements apply for instructor swaps. A swap of trainees, for the purpose of premium day redemption, is only applied if the other trainee has a due date that is beyond the last day of the published roster. This allows the airline to reschedule the training event without propagating disruptions onto other activities. This matches the desire to minimise the amount of changes in the recovery phase. As the recurrents have four major peaks in due dates (at the last day of every trimester), swapping for premium day redemption generally does not work well in these busy months. For standard crews, an additional rank requirement is put in place as an additional helpout instructor is required otherwise. The due date- and rank requirements do not hold for swapping instructors for the purpose of premium day redemption, but qualification requirements are enforced instead. If an instructor from a nonstandard session is disrupted, a swap instructor is sought based on qualification requirements that depend on the capabilities of the remaining instructor. The redemption of premium days is enforced via random sampling referenced to a historical probability.
2. **Swapping for due date benefits (RA6.2):** Related to redemption of premium days is that of swapping for a benefit in due dates. This has no direct impact on cost, but is aimed at minimising the risk of crew missing their due dates. This only applies to the undisrupted trainees of a disrupted simulator session. When no premium day can be redeemed, but there is another training event of the same type on the exact same date, trainees can be swapped for a due date benefit. This requires the due date of the trainee removed from the schedule to be outside the current published roster to limit disruption propagation. In case of a swapping trainees with the same rank or swapping to a nonstandard session, this requirements suffices. If rank is complementary, a helpout needs to be added to be a viable swap. The remaining resources that are impacted are listed as standby.

Finally, a special case exists in which no action has to be taken. If a nonstandard session has two trainees of which one gets disrupted, the session can continue as is. Although these sessions are not the most efficient, it limits the impact on the crew training schedule and does not add additional cost. In combination with other disruptions on the same day, a more efficient solution might be available. However, this is only reviewed if such a disruption is preceded by another one on the same day because of differences in the notification timing. Then a more efficient schedule might be drafted for trainees, freeing up additional instructors that can cover the other disruption. As explained, the probability of having two disruptions on the same day is rather low. The impact of this is thus expected to be minimal due to additional timing constraints.

The rule-based recovery model is extended to be applicable to both a European network and intercontinental network. As explained, swapping instructors between flights and training is mostly applicable to a short-haul network. For that reason, the recovery action can be allowed and disallowed by the user. Furthermore, the recovery algorithm is tailored such that the user can specify if some recovery options are desired or not. This is done by means of two additional user-specified choices on:

- **Reclamation of loss days for trainees:** Although trainees have mandatory preparation times for training sessions, it can be challenged if this is actually needed. Without taking a stand, the rule can be challenged analyse its impact is. For that reason, the user can turn this option on and off. The process for reclamation of loss days for instructors also apply to trainees, but with different requirements. A replacement trainee should be available that requires the training event within the current grace period of the activity. Also, dependent on the crew composition, a rank requirement exists. For a standard crew, the rank must be complementary. For a nonstandard crew, no rank requirement exists because helpouts are cross-seat qualified by default. The implementation is also arranged via random sampling references to a probability indicating the availability of a loss day to be reclaimed.

- **Using a non-instructor as helpout:** For similar reasons, all helpouts require to be instructors. Assigning a non-instructor as helpout is often undesired because of contractual agreements, yet it is not uncommon. In case of great urgency, a trainee is assigned as helpout. The user can determine if this option is allowed or not. Again, analysis of the impact of such decision can be explored without taking a stand. Using trainees as helpout is only allowed if no other duties fall open because of it. In case of a nonstandard session with a single trainee, the trainee helpout is ideally of complementary rank. Whenever the mandatory preparation time is not met, the trainee helpout cannot sign-off on its own training requirements. In case of a nonstandard crew with two trainees, this is also impossible. Still, complementary rank requirements are enforced because trainees are not cross-seat qualified.

## 7-3 Model Output

As the optimization of the cockpit crew training schedule is cost-based, the output of the recovery model is also cost based. Next to the cost associated with each recovered disruption, the recovery action used is also output. Per disruption scenario, the cost and usage of recovery options are summed for usage later on in the experiments, as described in chapter 8. The output recovery cost will be fed back to the Training Scheduling & Assignment Model as expected recovery cost.

As opposed to scenario wide feedback, each disruption is converted to a set of features separately and coupled with its simulated recovery cost. The features are both disruption dependent and roster state dependent, as is described in section 5-4, but are calculated per disruption to also capture the impact of different disruption probabilities. As the scheduling model is tested on its ability to adapt, the features are fed back in combination with the month of the year in which the disruption occurred. This is the independent variable rather than a feature. The same applies to the training demand of the current, previous and next week, which are related to the month of the year. The demand only varies (slightly) due to propagated courses and is used to scale the Training Scheduling & Assignment Model accordingly. The feature values that are fed back, of which an example is shown in Table 7-1, is a combination of categorical and continuous features. The time of day, day of the week and minimum qualification required are categorical features converted to integers using ordinal encoding. Ordinal coding is used instead of the more accurate One Hot Encoding method to reduce the amount of input features. Else, each simulator slot, day of the week and qualification would require a separate input node. The percentage of sufficiently qualified instructors, on a continuous scale of 0 to 1, that are unassigned in a certain week of the current roster is added to a constant of one and converted by taking the logarithmic value. The number of standard and nonstandard courses scheduled on the day of the disruption are subtracted to yield a continuous variable.

**Table 7-1:** Example of output data of the rule-based recovery model

| Time | Day | Qualification | Unassigned Instructors | Standard vs Nonstandard |
|---|---|---|---|---|
| 6.25 | 2 | 0 | 0.041 | -1 |
| 13.15 | 6 | 2 | 0.301 | 5 |

## 7-4 Validation and Discussion

The validation of the rule-based recovery model again consists of two parts:

1. Qualitative comparison of recovery data against that of the European legacy carrier.
2. Judgement of an operational expert solving disruptions on a daily basis based on provided information.

Both validation processes support the conclusion that the results are sufficiently accurate to test robustness of a cockpit crew training schedule and satisfy the research objective in making recommendations on how to achieve this. This means that for each of the recovery actions modelled, the order of magnitude matches reality.

Nonetheless, discrepancies exist because of the more static nature of the model and the way it is modelled. Based on 130,000+ disruptions it can be concluded that the recovery model uses 20 percent more reserves and as a result uses a third less overtime. Connected to that is a 15 percent lower cancellation rate. The connectivity between errors has to do with the fact that the rule-based recovery model uses a static scheme and continues to the next recovery option only if the previous action was unsuccessful. As the reserve model is based on historical probabilities, the difference comes from the implementation of the rule-based recovery model. The only differences comes from the amount of qualified crew that is unassigned at each moment. As the model present in this thesis only accounts for other activities in the schedule by means of some gap lengths, it overestimates the probability of sufficiently qualified crew being assigned as a reserve. As a result, the reserve usage is higher. Despite the discrepancy, the order of magnitudes are all the same and thus deemed sufficiently accurate for attaining the research objective.

Also contributing to the differences is that the recovery decision tree is static. As Abdelghany et al. (2004a) already stressed: human controllers can find much more complex options, for which a rule-based methods is unsuitable. Especially swapping multiple resources along several days becomes very difficult, whilst in reality such options could be easier to spot or implement. However, the tree-based method allows for very fast recovery, which is needed because of many disruption scenario's that need to be solved. The model solves a single disruption scenario, with an average of 11 disruptions, in 0.6 seconds approximately. The run time is already quite high, but that can be explained by checking all kinds of conditions. The tree-structure aids in keeping run time to a minimum as the model immediately cuts out of the tree when a solution is found. This run time comes close to what is indicated by Sohoni et al. (2011) and Bayliss et al. (2017), but is higher because of the relative complexity.

Another area of improvement is to make the model more adaptive. The model solves disruptions sequentially and could neglect better options whenever multiple disruptions occur at the same day. However, for such options to exist, the disruptions timing needs to be in line and also the disruption itself should be such that resources share connectivity. Even in the case of sequential solving connected disruptions, options as to reschedule trainees or instructors to a more efficient solution is considered, but the timing could be off. The mismatch of sequentially solving disruptions is thus expected to be small. Furthermore, the model should ideally be extended to better distinguish between recovery options for recurrent and conversion courses. Due to a lack of data, this is currently not implemented.

Concluding, the rule-based recovery model is fast within reason to deal with a high number of disruption scenarios. Also, despite some discrepancies, it is sufficiently accurate to test and prove robustness of a schedule. A last benefit is that the recovery model and its complex requirements and dependencies give away starting points to achieve robustness. Exploiting the cheaper recovery options maximally should yield lower recovery cost and thus a robust(er) solution. When able to combine this with low(er) schedule cost, a more efficient solution is obtained as well. However, the existence of such a combination is analysed and tested by conducting repeated simulations in the next chapter.

Chapter 8

# Experiments

The research framework and methodology as explained in chapter 4 has been treated in subsequent chapters. A brief recap on the complete model set-up is provided in section 8-1. The complete model is then used for a couple of experiments and results are compared to the reference roster. In the remainder of the section, the words reference roster and benchmark roster are used both. The experiments and associated results are discussed in section 8-2. To put the results into perspective, a validation scenario is ran to compare performance to that of reference airline's training schedule. The results of the validation are discussed in section 8-3.

## 8-1 Complete Model Set-Up

The various models presented in chapter 5, chapter 6 and chapter 7 are combined as displayed in Figure 4-2. The Training Scheduling & Assignment Model (TS&AM) uses data on training demand, resources such as crew- and simulator and assignment rules and converts it to a feasible training schedule. This roster is then input into the Primary Disruption Generator (PDG) per four weeks, along with historical data on illness and leave as primary courses of disruptions. A disruption scenario is randomly generated based on the historical disruption probabilities for the middle two weeks of the four week schedule, as indicated by the bars in Figure 8-1. The first and last week account for carry-in and carry-out of disruptions respectively. All disruptions occurring in these middle two weeks and involving training events are input into the Rule-Based Recovery (RBR) model. A static, cost oriented recovery tree is passed to find the most efficient recovery option. The model takes into account rules and regulations and uses random selection to mimic options to be available or not based on historical data as well. The process of generating a disruption scenario and recovering it is repeated $x$ times, where parameter $x$ is chosen by the user. The parameter controls the accuracy of the results, but also the run time of the model. After $x$ repetitions, the model moves on one week of schedule. As seen in Figure 8-1, the original first week is disregarded and a new week of the schedule is added. Again the middle two weeks are disrupted, meaning that an overlap of one week exists between the runs. A single week is tested on two runs of each $x$ repetitions with the exception of the second and second to last week of the full schedule. These weeks are tested $x$ times only. The first and last week of the full schedule are never tested as they account for carry-in and carry-out of the training activities rather than carry-in and carry-out of disruptions.

After all weeks have been simulated, the results are fed-back to the scheduling model. Either via Proportional Feedback (PF), in which a static transformation is applied based on the expected

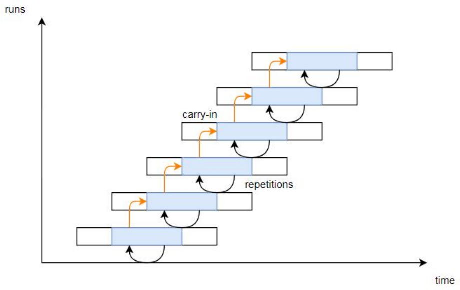

**Figure 8-1:** Methodological set-up

recovery cost per feature, or via a Neural Network (NN). The NN uses nonlinear regression to optimize for the synaptic weights and biases of each feature and combination thereof to estimate the recovery cost. The TS&AM is solved again with this additional input to generate a robust roster. The simulation of disruptions and recovery is redone for $x$ repetitions. The simulation results of the reference roster and robust roster can then be evaluated and compared. The evaluation is based on a combination of deterministic schedule cost and stochastic recovery cost, as explained in the definition of robustness throughout this thesis. A more detailed analysis is done to identify measures to proactively increase robustness of the cockpit crew training schedule, which is then translated into a set of recommendations.

A few parts of the complete model set-up have not been treated yet. This includes the evaluation criteria and experiments itself. First, the evaluation criteria, captured in the form of Key Performance Indicators (KPIs) are listed below.

- **Expected recovery cost:** The primary means of measuring robustness is recovery cost. A robust schedule is defined as one that operates on low expected recovery cost, as previously defined in this thesis. The expected recovery cost thus indicate the level of robustness of the schedule. It is expressed as cost per assignment to normalise for any schedule differences.
- **Total cost:** The definition of robustness used a condition on deterministic schedule cost, as inspired by Clausen et al. (2010). Robustness is concerned with the trade-off of schedule cost and expected recovery cost, meaning that the benefits of robustness is put into perspective using the total cost. It is also normalised to the cost per assignment.
- **Stability:** The first additional benefit that is quantified is the stability of the solution. The stability is captured in the Root Mean Squared Error (RMSE) of the expected recovery cost and total cost captured in terms of the cost per assignment.
- **Rate of missed due dates:** The impact of training disruptions can propagate into the flight schedule by means of reduced crew availability. Although missed due dates can be captured in terms of cost, it remains difficult to accurately quantify this. For that reason, the rate of missed due dates is used as a separate KPI.

## 8-2 Experiments and Results

A series of experiments is conducted to test various aspects of the model. Next to the aspects, the settings of the model have changed along the way as result of new insights. Due to limited time, the experiments have not been reconsidered such that the same settings apply to all. The differences in settings is explained at the start of the experiments. First, the robustness of the rosters generated using PF and NN is compared to the benchmark in subsection 8-2-1. Next, the recovery cost structure of the recovery model is updated to capture more cost factors. Details on these changes and the impact thereof on the robustness of the schedules are provided in subsection 8-2-2.

### 8-2-1 Robustness

The first experiment tests the ability of the model to generate a robust schedule referenced to a benchmark schedule generated with the exact same model and method. Starting, a reference roster is generated and disrupted for the year 2018. The first and last week serve as carry-in and carry-out and are thus neglected in the results and analysis of this experiment. The 50 remaining weeks are disrupted and recovered in blocks per two weeks, for 500 repetitions each. With exception of the second and second to last week, all weeks are part of the simulation two times, as is indicated in Figure 8-1. The output of the recovery model is used for both the PF and NN algorithms. The same methodology and number of repetitions is used to test both robust schedules. Figure 8-2 shows the spread (left) and convergence (right) of the normalised recovery cost in a single scenario tested for 500 repetitions. In total, 91.6 percent of the repetitions show a number of disruptions below twice the mean number of disruptions and a recovery cost below twice the mean recovery cost. Also, a positive linear trend is observed as each disruption adds to the recovery cost. The convergence plot shows that after approximately 250 runs, the mean recovery cost remains within five percent deviation up or down of the expected value (indicated by the dashed lines). Since the majority of the weeks are simulated twice, a reduction of number of repetitions is possible without losing accuracy of the results.

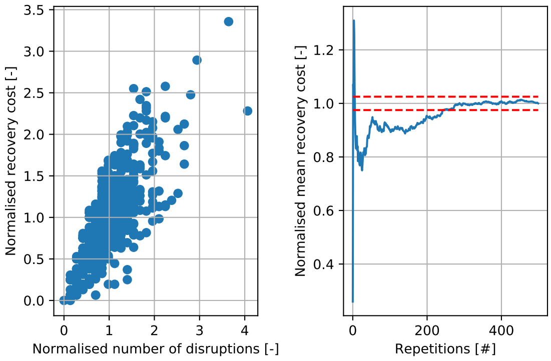

**Figure 8-2:** Spread and convergence of the normalised recovery cost of a single scenario tested in 500 repetitions

As previously explained, random decisions on spare simulator capacity, instructor selection and trainee selection are part of the Training Scheduling & Assignment Model (TS&AM). Together with the difference in methods this makes that the roster will consist of a slightly varying amount of assignments. An assignment is counted for each unique trainee assigned to a unique training session. This means that each day of a multiple-day long training program is counted separately and per trainee. The reference roster consists of 4141 assignments, the PF roster has 4233 assignments and the NN roster has 4123 assignments. The difference in the amount of assignments directly influences the schedule- and recovery cost, so all are normalized to the (recovery) cost per assignment.

On top, all results are normalized with respect to the mean recovery cost over the course of one year of the schedules with the minimum value. The results for the Reference Roster (RR), Proportional Feedback roster (PF) and Neural Network roster (NN) are displayed in Table 8-1. In terms of robustness, the best performing method proved to be the proportional feedback roster. It outperforms the reference roster and neural network roster by 1.36 and 3.42 percent respectively in terms of mean recovery cost per assignment. When looking at the stability of the solutions, the proportional feedback roster also shows an improvement over the other schedules. The neural network roster is also more stable than the reference roster. It is concluded that a robuster schedule can be generated under the normal conditions, but it needs to be put into

**Table 8-1:** Normalised mean and standard deviation of the cost per assignment

| | RR | PF | NN |
|---|---|---|---|
| Mean recovery cost per assignment | 1.0136 | 1 | 1.0342 |
| Standard deviation recovery cost per assignment | 0.6356 | 0.5975 | 0.6205 |
| Mean total cost per assignment | 1 | 1.0004 | 1.0050 |
| Standard deviation total cost per assignment | 0.0654 | 0.0606 | 0.0626 |

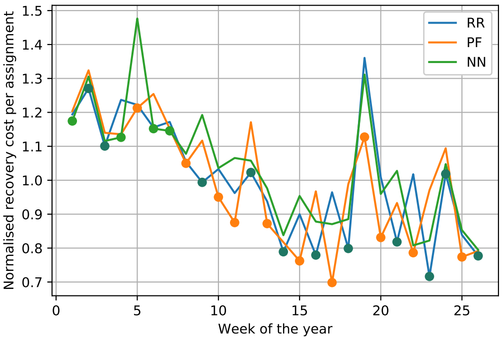

**Figure 8-3:** Normalized, weekly recovery cost per assignment for the RR, PF and NN rosters

perspective and further analysed. When looking at the total cost per assignment, which is the sum of deterministic schedule cost (fixed per week) and expected recovery cost, it is observed that the reference roster is most efficient by a small margin. The added robustness thus comes at the cost of an 0.04 percent decrease in total training cost per assignment.

The differences are further analysed by reviewing the differences in robustness per week. Instead of combining all data, the weekly data is separated. Figure 8-3 shows the mean recovery cost per assignment per week for the first semester of 2018 normalised by the mean recovery cost per assignment of the best performing roster. Note, a value above one means that the recovery cost is above average and the solution is thus less robust. For values below one it is the other way around. A trend is observed in which the first months (i.e. the winter season) have a low mean recovery cost per assignment as opposed to the summer season. This trend matches the trend of the disruption probabilities displayed in Figure 6-1, meaning that a high disruption probabilities leads to high values of the recovery cost per assignment and thereby less robust solutions.

The variation in the colour of the dots in Figure 8-3, indicating the most robust roster per week, make the analysis of robustness complex. An important reason for the variation is that robustness is linked to previous week(s) via the assignment rules, but it is also sensitive to instabilities, which are captured in the standard deviation. The peaks in week 5 (NN) and week 19 (RR and NN) are explained by preceding peaks in the assignment of unique instructors. In the week afterwards, less unique instructors are assigned leading to more courses per instructor. In case of illness or leave, more courses are disrupted and the value of the recovery cost per assignment increases whilst robustness decreases.

Next, the commonalities between the various schedules in the best performing weeks is analysed based on the coloured dots. For each schedule, the values of the performance indicators are selected for those weeks that are most robust. This set of values is then compared to the performance indicators of each of the schedules. Lastly, the Pearson correlation coefficient is determined between the recovery cost per assignment and the associated performance indicator. Values above one indicate a positive correlation. An increased value of recovery cost per assignment (i.e. a decrease in robustness) then leads to an increase in the performance indicator or vice versa. When the correlation coefficient is below 0, an inverse relationship is established. The closer the correlation

**Table 8-2:** Normalised performance indicators per schedule and correlation coefficients

| | RR | | PF | | NN | | |
|---|---|---|---|---|---|---|---|
| | mean | $\sigma$ | mean | $\sigma$ | mean | $\sigma$ | r |
| Unique instructors | 1 | 0.270 | 1.052 | 0.219 | 1.047 | 0.200 | 0.376 |
| *Unique SRE* | 1.075 | 0.448 | 1.046 | 0.423 | 1 | 0.382 | 0.247 |
| *Unique TRE* | 1 | 0.351 | 1.130 | 0.340 | 1.178 | 0.325 | 0.525 |
| *Unique TRI* | 1.136 | 0.480 | 1.088 | 0.414 | 1 | 0.437 | -0.329 |
| Courses per unique instructor | 1.042 | 0.144 | 1.034 | 0.144 | 1 | 0.128 | -0.275 |
| Overqualified instructors | 1.154 | 0.734 | 1.083 | 0.645 | 1 | 0.615 | 0.478 |
| Overqualified helpouts | 1.071 | 0.453 | 1 | 0.375 | 1.107 | 0.433 | 0.307 |

coefficient is to 1 or -1 respectively, the stronger the correlation. In this thesis, a value between 0 and 0.3 is considered to indicate a weak positive correlation, a value in the range of 0.3 to 0.7 is considered to indicate a moderate positive correlation and any value above indicates a strong positive correlation. The same ranges and conclusions apply to negative values.

Table 8-2 shows, per performance indicator, the normalised mean (in blue) and standard deviation (in green) of each of the schedules referenced to the best performing schedule in terms of mean robustness. The last column shows the Pearson's linear correlation coefficient $r$. The number of unique instructors assigned, unique Senior Type Rating Examiners (SRE), unique Type Rating Examiners (TRE), overqualified instructors and overqualified helpouts show a positive correlation to the recovery cost per assignment. An (independent) increase in one of these performance indicators results in a higher recovery cost per assignment and thus a less robust solution. The number of unique Type Rating Instructors (TRI) and courses per unique instructor show a negative correlation with the recovery cost per assignment. The negative correlation of courses per unique instructor is explained by the fact that it results in more instructors not assigned to any training duty. These can then be used as reserve crew or assigned to flight pairings that can be swapped with training duties. Moreover, the number of disrupted training events is independent of the number of crew members assigned and thus increasing the number of courses per unique instructor adds robustness to the solution. As seen in Table 8-2, the Neural Network roster performs best on this measure, indicating that the neural network is capable of learning ways to improve robustness. The opposite applies to the number of unique instructors and unique instructors per qualification. The higher the number of unique instructors, the less robust the solution will be as result of reduced instructor reserve availability. All these show a positive correlation coefficient, except for the number of unique TRIs. This is explained by the low percentage of unique TRIs assigned anyways in combination with the capabilities. This effects is intensified by the assignment of overqualified instructors as this is not penalized, nor restricted by the model. It turns out to restrict the recovery due to qualification requirements. As displayed in Table 8-2, the neural network schedule shows the lowest mean values on four out of the seven performance indicators. However, this is undesired as indicated by the two negative correlated performance indicators. As a result, the PF roster outperforms the NN roster in terms of robustness, as already showed in Table 8-1. Further experiments have to show if the neural network can outperform the proportional feedback algorithm and reference roster.

## 8-2-2 Robustness with Different Recovery Cost Structure

The previous experiment proved a small gain in robustness at a marginal increase of total cost per assignment. The reason for a limited benefit is the structure of measuring cost. The recovery model only adds salary cost and potentially simulator cost associated to the used recovery option. Salary cost of the crew member that is ill or on leave is neglected as this cannot specifically be accounted to the training schedule. The European airline at study also accounts for illness and leave on an overarching level. In the current implementation of the recovery cost structure, the stochastic recovery cost thus only depends on the available recovery options, their resource requirements and the associated dynamic availability rates.

For this experiment, an update is applied to define robustness in a more realistic way that suits the current model implementation and its characteristics. First, as not all legally required courses are scheduled by the model due to inflexibility, the cost of having more- or less instructors has to be computed on an annual basis including the extra salary for non-training duties as well. This allows to value robustness between the scenario's with different amounts of instructors. The impact of this change in the recovery cost structure is researched in the sensitivity analysis provided in section 9-2. Another change of the recovery cost structure is associated with cancellation. Currently, only rescheduling cost are added. Any further impact on the schedule in terms of crew unavailability is neglected. A linear penalty cost is implemented based on missing a due date. The penalty cost accounts for the salary of that crew member and an additional penalty for putting more strain onto the daily operation of flights due to decreased crew availability. The latter is highly nonlinear in reality and very difficult to quantify as it is dependent on both supply and demand of crew. It is therefore neglected, but noted to potentially be of great influence on the robustness of the schedule. Also note that the experiment assumes that legal, short-term extensions of due dates are not allowed. Normally, an airline could apply for an extension of qualification at the regulatory body for up to two weeks.

More specifically, a linear salary cost is added proportional to the number between the original due date and the date of regaining a qualified status. The cost of illness or leave is now attributed to the training schedule whenever the illness or leave extends beyond the due date of that trainee. As the model is not optimizing for placement of reserve simulator capacity nor reserve instructor capacity, the date of regaining a qualified status depends on two things: (1) the availability of a reserve simulator slot, or (2) the availability of a swappable session. The former requires a sufficiently qualified instructor and potentially a helpout instructor. The availability of the option of an open session thus depends on the probability of having these resources available. A swappable simulator session requires: (1) the original trainee(s) to have a due date outside of the current scheduling period such that they can be rescheduled without any impact, and (2) the instructor to be sufficiently qualified to provide the training event. It is assumed that the earliest available option is used by default. The model implicitly tries to learn the nonlinear effect of the gap to the next swappable- or reserve simulator session. This new recovery cost structure increased the recovery cost per trainee by approximately a factor of two. Again, the PF and NN outperformed the RR, where the PF performed best. The respective improvements are 1.11 percent and 0.60 percent in total cost per assignment and 28.50 and 24.97 percent improvements in stability. This is noticeable as the PF and NN added robustness to the schedule without impacting the deterministic schedule cost per assignment. That is, the stochastic recovery cost per assignment reduced by more than the increase in deterministic schedule cost per assignment. The only disappointing result is that the PF schedule outperformed the NN roster.

**Table 8-3:** Normalised mean and standard deviation of the cost per assignment with an updated recovery cost structure

| | RR | PF | NN |
|---|---|---|---|
| Mean recovery cost per assignment | 1.1675 | 1 | 1.0524 |
| Standard deviation recovery cost per assignment | 1.2551 | 1.0241 | 1.0458 |
| Mean total cost per assignment | 1.0111 | 1 | 1.0048 |
| Standard deviation total cost per assignment | 0.1366 | 0.1063 | 0.1093 |

When analysing the robustness of the best performing schedules on a weekly basis, the results are different compared to the results of the previous experiment. Figure 8-4 shows that the neural network roster outperforms the other roster in four of the thirteen weeks as opposed to zero in the previous experiment. Due to the different scales, it seems as if the peak in cost per assignment is

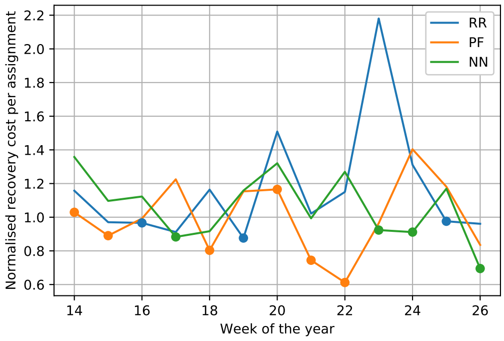

**Figure 8-4:** Normalized, weekly recovery cost per assignment for the RR, PF and NN rosters with an updated recovery cost structure

absent, but this is not true. However, the peak in recovery cost per assignment of the Reference Roster in week 23 is noticeable. The remainder of the analysis focuses on the correlation coefficients between robustness and the performance indicators treated in the previous experiment as well.

The variance in the recovery cost per assignment with the updated recovery cost structure is linked to the variance in the performance indicators as it turns out in Table 8-4. Again, this table shows the normalised mean and standard deviation of the recovery cost per assignment for all schedules and the correlation coefficient $r$ between the performance indicators and the recovery cost per assignment. The values in blue indicate the indicate the lowest mean per performance indicator (i.e. normalised to 1) and the values in green show the lowest standard deviation normalised to the lowest mean of that performance indicator. As seen, the reference roster has the lowest mean in unique number of instructors assigned per week, unique SRE assigned, unique TRE assigned, unique TRI assigned, overqualified instructor assignments and overqualified helpout assignments. The far right column shows that three of these show a moderate negative correlation and one a weak negative correlation with respect to the recovery cost per assignment. Higher mean values of these performance indicators lead to a reduction in recovery cost per assignment and thus a more robust solution. This corresponds to the conclusion of Table 8-3 that the RR is least robust. It is also the least stable solution, which is (partly) due to the presence of the peak. This peak is explained by the absence of a scaling function based the moving average of the training demand. The PF and NN rosters are able to schedule and assign based on the average demand of a three week period, meaning that a more stable solution is obtained. The RR acts purely based on the demand of the current week. As a result, the RR operates at high levels of unique instructors to cover the demand. However, due to assignment rules specifying that a non-training duty must exist between consecutive periods of instruction, the resources are exploited too much. As a result, an instability enters the system which is only recovered in case of steady, sub-average training demand. This does not occur in the period approaching summer holidays. On the other hand, the Neural Network roster shows that too much stability in instructor assignment (not the balance in instructor assignment between qualifications) could lead to the same problem due to the dynamic nature of the training demand. The neural network shows a tendency to operate in a stable manner and shows to do this, whilst providing flexibility within the assignment of instructors, as indicated in green in Table 8-4.

Noticeably, five out of seven correlation coefficients have converted in sign. Only the assignment of unique SRE instructors (positive correlation) and unique TRI instructors (negative correlation) have retained their direction of correlation. The positive correlation of SREs is explained by the limited number of qualified crew members and the relatively high demand in combination with assignment to duties requiring lower qualifications. The opposite applies to the assignment of unique TRIs. In fact, a higher utilisation of TRIs is stimulated from a robustness perspective,

**Table 8-4:** Normalised performance indicators per schedule and correlation coefficients

| | RR | | PF | | NN | | |
|---|---|---|---|---|---|---|---|
| | mean | $\sigma$ | mean | $\sigma$ | mean | $\sigma$ | r |
| Unique instructors | 1 | 0.265 | 1.162 | 0.208 | 1.138 | 0.197 | -0.307 |
| *Unique SRE* | 1 | 0.343 | 1.264 | 0.592 | 1.113 | 0.461 | 0.331 |
| *Unique TRE* | 1 | 0.283 | 1.274 | 0.332 | 1.274 | 0.285 | -0.256 |
| *Unique TRI* | 1.048 | 0.456 | 1 | 0.321 | 1.012 | 0.413 | -0.396 |
| Courses per unique instructor | 1.103 | 0.154 | 1.038 | 0.156 | 1 | 0.150 | 0.236 |
| Overqualified instructors | 1 | 0.668 | 1.112 | 0.543 | 1.012 | 0.502 | -0.531 |
| Overqualified helpouts | 1 | 0.403 | 1.033 | 0.364 | 1.018 | 0.348 | -0.453 |

but this is contradictory to using overqualified crew as a TRI is least qualified. Nevertheless, under-using TRIs is a waste of resources and creates an imbalance in workload. The remaining performance indicators that flipped signs are explained as follows. Instead of rewarding the assignment of consecutive instruction periods, the PF and NN algorithms as part of the Training Scheduling & Assignment Model (TS&AM) also reward the assignment of a balance of instructors. The benefit is that the gap to the nearest empty simulator slot or swappable training session becomes smaller and robustness is created. However, it contradicts the instructor assignment rule allowing at maximum a single step-back (i.e. one simulator slot earlier or later) for consecutive, efficient assignment. Using an increased number of unique instructors is inversely proportional to the number of courses per instructors. Thus, the same reasoning applies. The usage of overqualified instructors and helpouts also aids the swappability of instructors, which is more important to achieve robustness in the new recovery cost structure.

Based on the reversed signs of the correlation coefficients of the performance indicators, it is concluded that robustness is sensitive to the recovery cost structure. However, there is no right or wrong in what cost can be captured accurately or what is desired by the user. It is up to the user to specify a recovery cost structure adjusted to specific needs. Also, none of the methods proved to be able to combine all facets in the most favourable way. That is, high values for performance indicators contributing to robustness and low values for those reducing robustness. Finally, the PF algorithm performs better than the NN algorithm on mean mean recovery cost, mean total cost and stability of both categories. So a robust schedule is generated and in fact, a simple approach would suffice to yield up to a 1.11 percent more efficient schedule while also improving robustness by up to 16.75 percent. Still, the neural network showed performance approaching this level. These results are bench-marked to a schedule generated with the same model and method such that the conclusion is valid. However, the practical benefits cannot be assessed without validating the model and method against the training schedule of the European airline. The next section will focus on this validation step against an existing schedule.

## 8-3 Validation and Discussion

The number of assignments of the TS&AM is 95+ percent of that of the historical roster of the European airline, yet the rosters are different. The balance between recurrents and conversion courses is off. The TS&AM outputs a schedule with an approximately 50/50 balance in conversion training and recurrent training, whilst the European airline operates a 35/65 percent balance. As previously explained, this is caused by a mismatch in flexibility of the conversion training yardstick lengths with the result of consuming more resources for conversion training. Due to time constraints and the fact that all above experiments are based on the TS&AM as is, the validation will also use the model in which conversion yardsticks are strictly implemented. Also, the model and method rely on several assumptions that simplify the training scheduling and assignment process such as neglecting segregating the problem and using a concise set of assignment rules.

**Table 8-5:** Normalised mean and standard deviation of the cost per assignment with an updated recovery cost structure

| | RR | PF | NN |
|---|---|---|---|
| Mean recovery cost per assignment | 1.2741 | 1 | 1.0524 |
| Standard deviation recovery cost per assignment | 1.5732 | 1.0241 | 1.0458 |
| Mean total cost per assignment | 1.0326 | 1 | 1.0038 |
| Standard deviation total cost per assignment | 0.1694 | 0.1108 | 0.1138 |

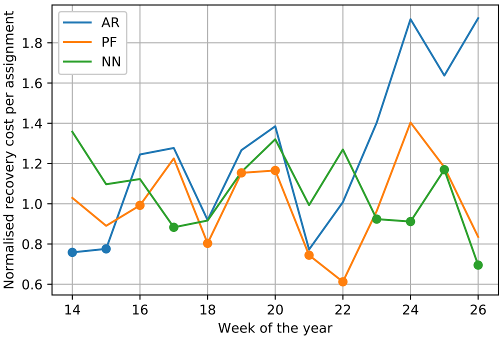

**Figure 8-5:** Normalized, weekly recovery cost per assignment for the AR, PF and NN rosters

The results are put into perspective of these assumptions after the results are presented

The validation consists of analysing robustness of the crew training schedule of the European airline on the months April, May and June, just as the previous experiment. Each week is simulated for 200 repetitions and the updated cost structure is applied. The schedules are tested with the developed Disruption Generator (DG) and Rule-Based Recovery (RBR) model instead of a historical disruption scenario because of the roster differences. The scheduled courses and assigned trainees and instructors are different for each day, meaning a historical disruption scenario would involve assumptions on how to process these such that a valid comparison is made. Instead, the simulation environment provides a basis for repetitive testing. The Reference Roster is now replaced with the Airline Roster (AR) and it is compared to both the PF and NN schedules. The results on robustness and total cost per assignment are displayed in Table 8-5. The PF and NN algorithms show an improvement in robustness of 27.41 percent and 21.06 percent respectively. The stability improved by 53.62 percent 50.43 percent. The noteworthy improvements translate into an improved total cost per assignment of 3.26 and 2.87 percent over the AR. The stability in total cost per assignment improved by 52.88 percent and 48.86 percent respectively.

Figure 8-5 shows a higher level of robustness in eleven of the 13 weeks for the PF roster over the Airline Roster. The NN schedule outperforms the AR in nine of the 13 weeks. The improved stability in recovery cost assignment also improves the stability in total cost per assignment and total cost per assignment. This improvement is explained by the decreased recovery cost per trainee, which represents between five and ten percent of the total cost, explaining the difference in total cost per assignment. The deterministic schedule cost remained approximately the same. Even, the airline assigns three trainees in approximately 3.5 percent of the training events, resulting in a benefit of 0.11 percent in schedule cost. This option is neglected in the model by assumption.

The noticeable difference in robustness is explained by the assignment of overqualified instructors and helpouts, as displayed in Figure 8-6 and Figure 8-7. Here, the rate of assignment of overqualified instructors and helpouts is normalised with the total rate of assignment of these resources. As indicated previously in Table 8-4, these performance indicators show a moderate negative correlation with robustness. That means, an increased value for the performance indicators yields

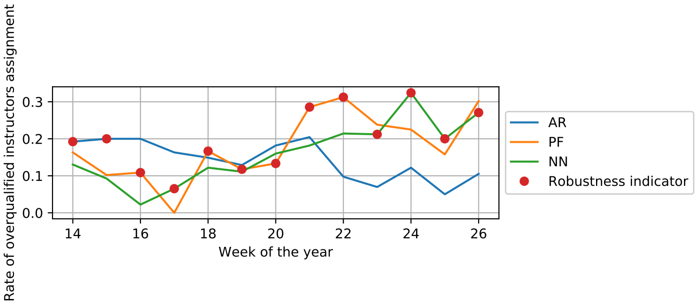

**Figure 8-6:** Normalized, rate of assignment of overqualified instructors for the AR, PF and NN rosters

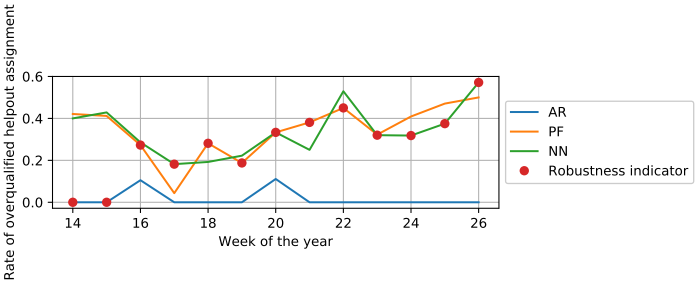

**Figure 8-7:** Normalized, rate of assignment of overqualified helpouts for the AR, PF and NN rosters

a more robust schedule. It is also observed in the plots, where the red dots indicate the most robust schedule of that week. In general, these red dots correspond to a high rate of assignment of overqualified instructors or helpouts. The Airline Roster showed a relatively high rate of assignment of overqualified instructors in week 14 and 15, which is translated into a good level of robustness. The opposite applies to the weeks 22 to 26. Figure 8-7 shows that the airline assigns minimum qualifications to the helpout positions. Although this balances the workload between the groups of instructors per qualification, it reduces the number of swap opportunities in the schedule, ultimately impacting the robustness and total cost per assignment.

The validation showed that added robustness can lead to a gain in total cost per assignment. However, the previous experiment indicates a smaller benefit. The difference is expected to be (largely) caused by the assumptions applied in this thesis. For example, the mismatch in balance between conversion training and recurrent training impacts the robustness. Conversion courses are guaranteed to be assigned to instructors on a consecutive basis, meaning that an illness or leave is likely to disrupt more sessions at once. The number of disruptions is directly proportional to the recovery cost and thus the robustness of the solution. Also, the assignment rules that cater for efficient assignment are fixed and restrictive. The airline is able to flexibly schedule and assign duties with changing gap lengths between blocks of training. Such options allow to better exploit robustness of the schedule. Lastly, two main assumptions are used to simplify the cockpit crew training problem and add schedule flexibility. The problem has a minimum interdependency with the flight schedule and all crew members are employed on a full-time basis. These assumption provide the model with a high degree of flexibility in instructor assignment. This flexibility steers the TS&AM to assign more unique instructors and use overqualified crew members to increase probability of having an efficient recovery action available and thus a robust solution. On the contrary, robustness provides added value when restricted in resources, meaning that reduced flexibility could also lead to increased robustness. Concluding, robustness of the cockpit crew training schedule is achieved and can even lead to a more efficient schedule at the same time, but further research is needed to challenge (some of) the assumptions made in this thesis.

# Chapter 9

# Sensitivity Analysis

As the experiments have shown, a (marginal) gain in robustness, stability and total cost per assignment can be achieved. Furthermore, it is concluded that the performance is dependent on the problem size, model settings and applied cost structure. For this reason, two sensitivities are checked before the thesis is concluded upon. The most obvious sensitivity to check is that of the disruption probability, as will be done in section 9-1. Secondly, the sensitivity of the number of instructors is reviewed in section 9-2. The number of instructors namely involves a trade-off between schedule cost and robustness.

## 9-1 Sensitivity of the Disruption Probability

The current model implementation focuses solely on illness and leave as sources of disruptions. The likelihood of this underestimating the real disruption probability is high, but as stressed before, accurately modelling other sources of disruptions proved not difficult. A straightforward way to check the validity of conclusion under different circumstances is to do a sensitivity analysis on the disruption probability. The model is consecutively tested on disruption probability that are multiplied by a factor of 1.5 and 2. This implicitly assumes that the other sources of disruptions follow the same probability distributions on length, extension, notification timing and timing of reporting for duties prior to the estimated disruption length as explained in chapter 6.

When increasing the disruption probability by a factor of 1.5 on the roster with additional instructors, the decrease in total cost per assignment is only 0.8 percent. This means that the total cost per assignment, determined by the total cost per trainee, only has a little dependence on disruption probability. This is remarkable as 1.5 times more disruptions apparently yield a similar performance in terms of recovery. As can be seen in Figure 9-1, the PF and NN show an improved performance of 0.15 and 0.26 percent respectively, which is very similar to that of the case with 'normal' disruption probabilities. The stability of the solution improved for the PF by a marginal 1.8 percent and 6.6 percent for the NN roster. Again this is similar to the performance of the reference case.

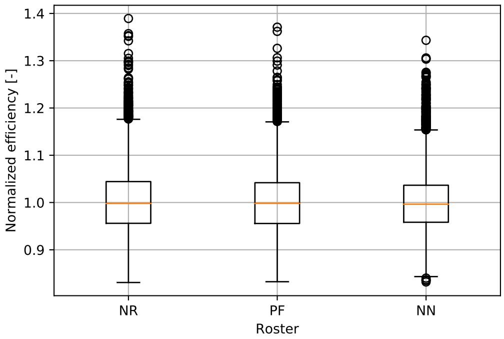

**Figure 9-1:** Boxplot on normalized total cost per assignment for the RR, PF and NN rosters with a 50 percent increase in disruption probability

When multiplying the disruption probability with a factor of two, similar results are observed. The decrease in total cost per assignment is 2.8 percent, which is larger but still low when considering the fact that the amount of disruptions doubled. The gain in total cost per assignment of proportional feedback and a neural network approach is 0.2 percent for both methods. The stability of the PF and NN improved over the reference by 2.3 and 8.6 percent respectively.

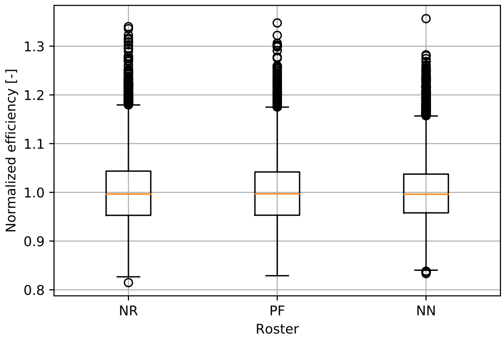

**Figure 9-2:** Boxplot on normalized total cost per assignment for the RR, PF and NN rosters with a 100 percent increase in disruption probability

From both runs of the model with increased disruption probabilities can be concluded that performance remains very similar, but the decreased total cost per trainee is too low. As the stochastic recovery cost accounts for five to ten percent of the total cost, doubling the disruption probabilities should lead to a larger reduction of total cost per assignment than just 2.8 percent. From inspection is learned that a higher disruption probability also leads to a higher cancellation rate. More leftover resources are thus put on standby, which can then be used to recover part of the additional disruptions. However, it is also observed that the availability of reserves is modelled by a general probability specifying the availability of reserves. the probability is not updated for previous usage of reserve crew. When the disruption probability increases, the chance of having more than one disruption per day increases. In such cases, the model could use reserve crew that are not available in reality and the discrepancy in the results becomes larger. This effects should therefore be noted to the results of this sensitivity analysis.

## 9-2 Sensitivity of the Number of Instructors

The experiments showed that a higher number of courses assigned per instructor improves robustness because of an increased availability of qualified instructors for recovery. The obvious choice is to increase the number of instructors, but this also adds salary cost of the additionally qualified instructors to the the total cost. The salary cost are added over the course of the entire year, whilst instructor capacity is not limited. In fact, the reference schedule assigned an average amount of 44.6 courses per SRE, 37.3 courses per TRE and 22.5 courses per TRI. All are lower than the limit of 60 annual simulator instruction events. The low utilisation of TRIs can be explained by the combination of capabilities, relatively high numbers and using overqualified instructors. The instructor utilisation rates provide an opportunity to review a reduced number of instructors as well. This reduces salary cost, but is expected to increase the recovery cost. Investigating the schedule robustness with a lower number of instructors is also interesting from a research perspective. Robustness is hypothesised to be more valuable near the boundaries of the balance between demand and supply, rather than at steady operation of the schedule.

To test this hypothesis, the amount of instructors per qualification is increased and decreased by 20 percent for each qualification. As a result, the total cost per assignment, adjusted for the cost of having more instructors, increased by 9.27 percent. The reduction of instructors leads to a reduction of 9.27 percent in total cost per assignment. A schedule is generated for each of the new instructor levels and algorithms to be tested afterwards. With a 20 percent increase in the number of instructors has led to an increase in the number of assignments with respect to the previous experiments. The benchmark roster has 4483 assignments (+ 8.3 percent), 4501 assignments have been output in the PF roster (+ 6.3 percent) and the NN roster contains 4417 assignments (+ 7.1 percent). With a 20 percent reduction in instructors, the number of assignments reduced with respect to the original experiment. The reference roster has 3861 assignments (-6.7 percent), the PF schedule contains 3633 assignments (-14.2 percent) and 3683 assignments are output in the NN schedule (-10.7 percent). The robust rosters show a larger decrease and a smaller increase in the number of assignments than the benchmark schedule because more unique instructors are assigned to gain robustness, as already indicated in the experiments. Due to the assignment rules this leads to a less efficient schedule and thus a larger discrepancy with the original schedule. However, note that a 20 percent decrease leads up to 15 percent lower assignments. This indicates that the decreased level of instructors approaches the critical values.

Noticeably, the behaviour of the models changed with the different levels of instructor resources. In the case of a 20 percent reduction in instructor capacity, the reference roster outputted the most robust roster and showed steady behaviour. Peaks in recovery cost per assignment are observed for the PF and NN, which are explained as an instability. Namely, these models have learned to assign higher levels of unique instructors than is efficiently available. When the model continuously assigns a higher level of unique instructors, it actually operates in a non-robust region. The solution to this problem is to redo the learning process of the PF and NN algorithms. As the focus of this section is to indicate the sensitivity compared for the various levels of instructors, the best performing roster is taken as a reference and the learning process is not restarted. In case of a 20 percent increase in instructors, the robustness of the RR, PF and NN rosters lies just 0.5 percent apart. This translates into a marginal difference of 0.2 percent in total cost per assignment. In fact, the availability of instructors has become so unrestricted that the schedules are very similar, leading to marginal differences.

The impact on cost of a change in the number of instructors is displayed in Table 9-1. All values are normalised with respect to the best performing method of the experiment on robustness of subsection 8-2-1. The table also shows that, in case of a 20 percent reduction in the amount of instructors, the recovery cost per assignment decreases by 29.61 percent, translating into a 5.19 percent higher total cost per assignment. The standard deviations also increase as the model is less able to deal with disruptions. In case of a 20 percent increase in instructors, the impact is almost negligible as already explained. The last two columns of Table 9-1 show the total cost per

**Table 9-1:** Normalised cost per assignment with an adjusted amount of instructors per qualification

| | Original | - 20 percent | + 20 percent | - 20 percent adjusted | + 20 percent adjusted |
|---|---|---|---|---|---|
| Recovery cost per assignment | 1 | 1.2961 | 0.9720 | - | - |
| Standard deviation recovery cost per assignment | 0.5975 | 0.8100 | 0.5698 | - | - |
| Total cost per assignment | 1 | 1.0519 | 0.9971 | 0.9233 | 1.1074 |
| Standard deviation total cost per assignmnet | 0.0654 | 0.0688 | 0.0618 | - | - |

assignment when the cost is corrected for a decreased or increased salary cost due to the changed number of instructors. This shows that a reduction of 20 percent of instructors actually improves the total cost per assignment by 7.67 percent. An increase of 20 percent of instructors adds 10.74 percent to the total cost per assignment. It can be concluded that the gain in robustness of increasing the amount of instructors does not outweigh the added salary cost. Similarly, the reduced salary cost does outweigh the reduction in robustness.

Chapter 10

# Conclusions, Recommendations and Discussion

This thesis has considered the problem of robust cockpit crew training scheduling. The posed research question is:

*What robustness measures can be taken to proactively minimize the impact of disruptions related to an airline cockpit crew training schedule considering relevant rules and regulations?*

The research objective is to make recommendations on increasing proactive robustness with respect to crew training schedule by identification and evaluation of proactive robustness measures and solution methods that utilize historical training disruption data. Again, robustness is defined as the capability to deal with- or absorb negative effects of unexpected events. The conclusions on the applied models, methods and associated research contributions are described in section 10-1. This section also contains the answer to the research question. A set of recommendations for further research is discussed in section 10-2. The directions for further research are both aimed at further theoretical research or at the implementation of this project in practice.

## 10-1 Conclusions

From a literature study it was concluded that literature on cockpit crew training scheduling is limited. Research only considered segregated training scheduling problems for conversion training, recurrent training or instructor assignment. Resource dependencies are all neglected. The most elaborate model on integrated scheduling of recurrent and conversion courses involved simplifications and took 18 hours to solve. A more integrated model and faster solution method are needed to generate a cockpit crew training schedule that serves as a benchmark for the remainder of the research. This remainder is to focus on introducing robustness to the cockpit crew training schedule.

To address the robust cockpit crew training scheduling problem, a research model was developed. It consists of a Training Scheduling & Assignment Model (TS&AM), a Disruption Generator (DG), a Rule-Based Recovery (RBR) model and a feedback loop using either Proportional Feedback (PF) or a Neural Network (NN) to compute the expected recovery cost. Each of the (combined) models is described below along with the research contribution(s). Afterwards, the section continues with the conclusion on the experiments and sensitivity analysis.

### Training Scheduling & Assignment Model

The novel TS&AM is integrates scheduling of courses and assignment of trainees, instructors and simulators. The objective is to minimise deterministic schedule cost for the reference roster and a combination of deterministic schedule cost and stochastic recovery cost for a robust roster. The TS&AM is solved iteratively using a Construction Heuristic (CH) that rolls over a predefined time horizon. The CH uses a Selection Heuristic to reduce the amount of (unnecessary) options to substantially reduce run time while maintaining solution quality. Despite the increased complexity of integrating resources, the model solves an annual training schedule involving 5,000 training events and up to 1,000 crew members in five minutes. With an adaptation in the settings applied, the schedule output by the TS&AM covered 97 percent of the original training demand.

### Disruption Generator

The output roster serves as input for a data-driven disruption generator based on Monte-Carlo Simulation. The disruption generator accounts for dynamic disruption probabilities for illness and leave of each individual crew member. A novel aspect is added in stochastic modelling of the dynamics of illness by including a mismatch between expected illness length and actual illness length. As a result, the model accounts for up to two extensions of illness with a conditional notification timing. The model also considers ending of illness prior to the estimated length. Due to this complex implementation, a monthly disruption scenario involving up to 150 crew members is generated in 0.1 second. This is computationally expensive, but not beyond limits.

### Rule-Based Recovery Model

The disruptions are then solved using a Rule-Based Recovery (RBR) algorithm. The application of the fast, data-driven, tree-like structured heuristic is novel. The complexity of the RBR is threefold: (1) a different branch exists per disrupted resource (trainee and instructor) and per assigned crew composition (standard or nonstandard), (2) adaptive conditions apply that depend on the type of disrupted resource and crew composition, and (3) updating of recovery resources for previously cancelled training sessions. As a result, the RBR solves each disruption scenario of two weeks involving an average of ten disruptions in approximately 0.1 second.

### Proportional Feedback and Neural Network

Both a Proportional Feedback (PF) and Neural Network (NN) algorithm are tested on the same set of features separately to learn from the output of the recovery model. The objective is to generate a robust crew training schedule using the TS&AM. Applying a feedback loop or learning algorithms to reward attributes of the roster is tested in literature before, but the application to robust cockpit crew training scheduling is novel. The PF algorithm uses a fixed scheme to determine expected recovery cost. However, this method relies on the assumption that recovery cost can be estimated for each feature independently in an accurate way. This might not be the case. Instead, the NN avoids this issue by using nonlinear regression to estimate recovery cost. However, the NN requires vast amounts of training data to converge, which could take, dependent on the settings and the training data, hours to converge.

Continuing with the conclusion on the complete model set-up. The research model is used to generate an initial roster which serves as a benchmark whilst the robust rosters are disrupted and recovered according to the same process. Based on a comparison of the ability of the schedules to deal with disruptions, recommendations are composed on how to make a cockpit crew training schedule robust. Two experiments are conducted the compare robustness of the benchmark and robust schedules in terms of recovery cost: (1) test robustness under a cost structure in which only the cost of the recovery action are attributed to the recovery cost, and (2) test robustness under an updated cost structure in which missed due dates are penalised linearly with the number of days until regaining the qualification. Additionally, the sensitivity of the number of available instructors is checked. In the first experiment, the PF algorithm provided the most robust roster. In terms of robustness, it outperformed the benchmark schedule by 1.36 percent and the NN schedule by 3.42 percent. However, the benchmark schedule showed to highest efficiency measured in schedule cost plus recovery cost. The PF roster and NN schedule showed a marginal increase of 0.04 and 0.50 percent respectively. The added robustness of the PF roster provides an improvement of stability of total cost of 7.92 and 4.47 percent for the benchmark and NN rosters respectively.

The second experiment, which uses an updated cost structure that linearly penalises crew unavailability due to missed due dates, showed a larger projected gain in robustness. The PF algorithm still performed best, followed by the NN algorithm. Both outperformed the benchmark roster. The gain in robustness of the PF and NN algorithms with respect to benchmark schedule amounts 16.75 and 10.93 percent respectively. The gain in robustness negated any (marginal) increase in schedule cost triggered by the TS&AM objective on robustness as opposed to a pure efficiency objective. The total cost of the PF schedule is 1.11 percent lower than that of the benchmark schedule and 0.63 percent lower for the NN roster. The respective gain in stability is 28.50 and 24.97 percent. In more practical terms, this means that robust roster leads to a projected saving of up to 16.75 percent of recovery cost or cost of crew unavailability due to missed due dates with respect to a roster that is generated using the same method and model, but a different objective.

The sensitivity analysis on the number of instructors puts the conclusions of the research into perspective. Under the assumption that the added salary cost of an instructor (with respect to a regular crew member) can be attributed to the training schedule, a reduction in total cost per assignment can be achieved. In fact, the total cost per assignment can be reduced by 7.67 percent when reducing the amount of instructors by 20 percent for each qualification. It comes at the cost of a 29.61 percent decrease in robustness. On the other hand, increasing the number of instructors improves robustness by 2.80 percent while increasing the total cost per assignment by 10.74 percent. From the sensitivity analysis can be concluded that the gain in robustness is marginal as long as the number of instructors can be optimised. The importance of robustness only increases when the amount of instructors decreases towards the limits.

In summary, the PF and NN algorithms are both viable methods of generating a robust cockpit crew training schedule. However, the full potential of the nonlinear regression of the Neural Network has not been exploited. Nonetheless, both methods show similar behaviour into obtaining a higher level of robustness, which is translated into recommendations. Assigning overqualified instructors shows a moderate correlation to robustness. The same applies to assigning high(er) levels of unique instructors, each instructor covering less courses. However, this only applies to Type Rating Examiners (TREs) and Type Rating Instructors (TRIs) as the demand and supply of these groups is in balance. The opposite applies to the group of Senior Type Rating Examiners (SREs). The utilisation of SREs is higher making an even higher utilisation of unique SREs impact robustness negatively. This is explained by the low availability rate of SRE resources to cover disruptions. Assigning more unique instructors and overqualified instructors increases the number of swap opportunities. Another method to increase swap opportunities is to simplify the schedule by assigning similar courses on the same day. This is also related to using overqualified instructors in case a simplified schedule cannot be obtained due to variance in demand per course. Lastly, a stable schedule is beneficial for robustness, but the TS&AM must remain sufficiently agile to deal with inherent peaks in training demand and resource supply, also on a course specific basis and per instructors qualification. This stability ensures that a balance is obtained between resources used for training and for non-training duties. Crew assigned to the latter category can then be used for swap opportunities or reserves.

## 10-2 Recommendations and Discussion

This research is a first step into the domain of robust cockpit crew training scheduling. For this reason, directions of further research are identified and practical aspects are listed that need to be arranged for practical use of the work presented.

Starting, the research focused on identification and evaluation of proactive robustness measures that can be applied to cockpit crew training. The experiments and sensitivity analysis showed what parameters impact the robustness and in which direction (positively or negatively). It is also seen that the correlation can flip sign when operating near the boundaries of what is capable. An opportunities lies in further researching the robustness measures and associated optimal levels. Further research into the performance of the Neural Network can aid robustness as so far the Neural Network has not outperformed the PF algorithm. This primarily involves improving the set of features used, but also other settings to better enable the model to capture nonlinear relationships. A last theoretical opportunity is to find a solution method that can solve the LP model to its global optimum. The local optimisation of the construction heuristic worked well for the benchmark schedule as this was solely based on deterministic, static cost. The PF and ANN introduced dynamic cost differentiation, meaning that local optimization is suboptimal. Although a benefit in total cost cannot be guaranteed, global optimisation of the robust cockpit crew training schedule is an interesting direction.

Next to the theoretical research opportunities, some practical aspects can be further researched. Assumptions need to be challenged such that the output of the model is sufficiently close for practical use. Important practical aspects to model are the dependency with the flight schedule both for resource dependencies and disruption propagation, chaining of activities and accurate modelling of crew employment. In the process, it is important to test the model and prove concepts on other aircraft types and applications. Here, different rules might apply leading to different results. An example of which is the difference between a short-haul network and long-haul network. Similar differences exist for simulator training performed at external facilities.

The thesis is ended with a note on the practical value of the current model implementation. The integrated TS&AM proved to be a fast and accurate way of scheduling training events for realistically sized problems. It is therefore suitable for conducting analysis. Examples of aspects to analyse are the impact of (adding or removing) assignment rules in a Collective Labour Agreement (CLA) or having a different amount of instructors. These are recurring practical issues for airlines that can be tested independently of the operational processes in a fast way. The effects of such changes can be accurately assessed in both a deterministic and stochastic environment. Finally, an opportunity arises to integrate the (robust) cockpit crew training scheduling model(s) into a larger, overarching crew scheduling model. integrated approaches lead to more accurate decision making and this more efficient airline operations. The models in this thesis are developed in such a way that integration with other models is possible.

# Bibliography

Abbink, E., Mobach, D., Fioole, P., Kroon, L., Van Der Heijden, E., and Wijngaards, N. (2009). Actor-agent application for train driver rescheduling. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 356–363.

Abdelghany, A., Ekollu, G., Narasimhan, R., and Abdelghany, K. (2004a). A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127(1-4):309–331.

Abdelghany, K., Abdelghany, A., and Ekollu, G. (2008). An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185(2):825–848.

Abdelghany, K., Shah, S., Raina, S., and Abdelghany, A. (2004b). A model for projecting flight delays during irregular operation conditions. *Journal of Air Transport Management*, 10(6):385–394.

Ahmed, M. and Alkhamis, T. (2009). Simulation optimization for an emergency department healthcare unit in kuwait. *European Journal of Operational Research*, 198(3):936–942.

Barmby, T. (2002). Worker absenteeism: a discrete hazard model with bivariate heterogeneity. *Labour Economics*, 9(4):469–476.

Barnhart, C., Belobaba, P., and Odoni, A. (2003). Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391.

Bayliss, C. (2016). *Airline reserve crew scheduling under uncertainty.* PhD thesis, University of Nottingham.

Bayliss, C., De Maere, G., Atkin, J., and Paelinck, M. (2012). Probabilistic airline reserve crew scheduling model. *OpenAccess Series in Informatics*, 25:132–143.

Bayliss, C., De Maere, G., Atkin, J., and Paelinck, M. (2013). Scheduling airline reserve crew to minimise crew related delay using simulated airline recovery and a probabilistic optimisation model. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1944–1950.

Bayliss, C., De Maere, G., Atkin, J., and Paelinck, M. (2017). A simulation scenario based mixed integer programming approach to airline reserve crew scheduling under uncertainty. *Annals of Operations Research*, 252(2):335–363.

Belobaba, P., Odoni, A., and Barnhart, C. (2015). *The global airline industry.* John Wiley & Sons.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

Bijvank, M., Byrka, J., Van Heijster, P., Gnedin, A., Olejniczak, T., Świst, T., Zyprych, J., Bisseling, R., Mulder, J., Paelinck, M., and de Ridder, H. (2007). Cabin crew rostering at klm: optimization of reserves. In *European Study Group Mathematics with Industry*, volume 58.

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., and Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37.

Campbell, G. (1999). Cross-utilization of workers whose capabilities differ. *Management Science*, 45(5):722–732.

Campbell, G. (2011). A two-stage stochastic program for scheduling and allocating cross-trained workers. *Journal of the Operational Research Society*, 62(6):1038âĂŞ1047.

Clarke, J.-P., Melconian, T., Bly, E., and Rabbani, F. (2007). Means - mit extensible air network simulation. *Simulation*, 83(5):385–399.

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

Devore, J. (1999). *Probability and statistics for engineering and the sciences.* Brooks/Cole CENGAGE Learning.

Dück, V., Ionescu, L., Kliewer, N., and Suhl, L. (2012). Increasing stability of crew and aircraft schedules. *Transportation Research Part C: Emerging Technologies*, 20(1):47–61.

Dunbar, M., Froyland, G., and Wu, C.-L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Science*, 46(2):204–216.

Easton, F. (2014). Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand. *Production and Operations Management*, 23(4):660–675.

Ehrgott, M. and Ryan, D. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150.

Gamache, M. and Soumis, F. (1998). A method for optimally solving the rostering problem. In *Operations Research in the Airline Industry*, pages 124–157. Springer Boston, MA.

Haase, K., Latteier, J., and Schirmer, A. (1999). Course planning at lufthansa technical training: constructing more profitable schedule. *Interfaces*, 29(5):95–109.

Holm, A. (2008). Manpower planning in airlines - modeling and optimization. Master's thesis, Linköpings Universitet.

Ingels, J. and Maenhout, B. (2015). The impact of reserve duties on the robustness of a personnel shift roster: an empirical investigation. *Computers and Operations Research*, 61(1):153–169.

Ingels, J. and Maenhout, B. (2017). Employee substitutability as a tool to improve the robustness in personnel scheduling. *OR Spectrum*, 39(3):623–658.

Ionescu, L. and Kliewer, N. (2011). Increasing flexibility of airline crew schedules. *Procedia - Social and Behavioral Sciences*, 20:1019–1028.

Ionescu, L., Kliewer, N., and Schramme, T. (2010). A comparison of recovery strategies for crew and aircraft schedules. In *Operations Research Proceedings*, pages 269–274.

Kasirzadeh, A., Saddoune, M., and Soumis, F. (2017). Airline crew scheduling: models, algorithms, and data sets. *EURO Journal on Transportation and Logistics*, 6(2):111–137.

Kohl, N. and Karisch, S. (2004). Airline crew rostering: problem types, modeling and optimization. *Annals of Operations Research*, 127(1-4):223–257.

Kohl, N., Larsen, A., Larsen, J., Ross, A., and Tiourine, S. (2007). Airline disruption management - perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162.

Lagerholm, M., Peterson, C., and Söderberg, B. (2000). Airline crew scheduling using potts mean field techniques. *European Journal of Operations Research*, 120(1):81–96.

Lapp, M., AhmadBeygi, S., Cohn, A., and Tsimhoni, O. (2008). A recursion-based approach to simulating airline schedule robustness. *Winter Simulation Conference*, pages 2661–2667.

Lettovský, L., Johnson, E., and Nemhauser, G. (2000). Airline crew recovery. *Transportation Science*, 34(4):337–348.

Maenhout, B. and Vanhoucke, M. (2010a). Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling*, 13(1):77–93.

Maenhout, B. and Vanhoucke, M. (2010b). A hybrid scatter search for personalized crew rostering in the airline industry. *European Journal of Operations Research*, 206(1):155–167.

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

Maher, S. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1):216–239.

Mańdziuk, J. and Świechowski, M. (2017). UCT in capacitated vehicle routing problem with traffic jams. *Information Sciences*, 406-407:42–56.

Medard, C. and Sawhney, N. (2007). Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183(3):1013–1027.

Mendes-Moreira, J., Moreira-Matias, L., Gama, J., and Freire de Sousa, J. (2015). Validating the coverage of bus schedules: A machine learning approach. *Information Sciences*, 293(1):299–313.

Nissen, R. and Haase, K. (2006). Duty-period-based network model for crew rescheduling in european airlines. *Journal of Scheduling*, 9(3):255–278.

Petersen, J., Sölveling, G., Clarke, J.-P., Johnson, E., and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transportation Science*, 46(4):482–500.

Potthoff, D., Huisman, D., and Desaulniers, G. (2010). Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, 44(4):493–505.

Potvin, J.-Y., Shen, Y., and Rousseau, J.-M. (1992). Neural network for automated vehicle dispatching. *Computer and Operations Research*, 19(3-4):267–276.

Qi, X., Bard, J., and Yu, G. (2004). Class scheduling for pilot training. *Operations Research*, 52(1):148–162.

Quintiq (2017). Klm to optimize flight simulator capacity with quintiq. https://www.quintiq.com/klm-to-optimize-flight-simulator-capacity-with-quintiq.html.

Rezanova, N. and Ryan, D. (2010). The train driver recovery problem - a set partitioning based model and solution method. *Computers and Operations Research*, 37(5):845–856.

Rosenberger, J., Schaefer, A., Goldsman, D., Johnson, E., Kleywegt, A., and Nemhauser, G. (2000). Simair: A stochastic model of airline operations. *Winter Simulation Conference Proceedings*, 2:1118–1122.

Rosenberger, J., Schaefer, A., Goldsman, D., Johnson, E., Kleywegt, A., and Nemhauser, G. (2002). A stochastic model of airline operations. *Transportation Science*, 36(4):357–377.

Ross, S. (2010). *Introduction to probability models*. Elsevier.

Salazar-Gonzáles, J.-J. (2015). Approaches to solve fleet assignment, aircraft routing, crew pairing and crew rostering problems for a regional carrier. *Omega*, 43:71–82.

Schaefer, A., Johnson, E., Kleywegt, A., and Nemhauser, G. (2005). Airline crew scheduling under uncertainty. *Transportation Science*, 39(3):340–348.

Sejdovic, S., Hegenbarth, Y., Ristow, G., and Schmidt, R. (2016). Industry paper: proactive disruption management system: how not to be surprised by upcoming situations. In *ACM International Conference on Distributed and Event-Based Systems*, pages 281–288.

Shebalov, S. and Klabjan, D. (2006). Robust airline crew pairing: move-up crews. *Transportation Science*, 40(3):300–312.

Sohoni, M., Bailey, T., Martin, K., Carter, H., and Johnson, E. (2003). Delta optimizes continuing-qualification-training schedules for pilots. *Interfaces*, 33(5):57–70.

Sohoni, M., Johnson, E., and Bailey, T. (2004). Long-range reserve crew manpower planning. *Management Science*, 50(6):724–739.

Sohoni, M., Johnson, E., and Bailey, T. (2006). Operational airline reserve crew planning. *Journal of Scheduling*, 9(3):203–221.

Sohoni, M., Lee, Y.-G., and Klabjan, D. (2011). Robust airline scheduling under block time uncertainty. *Transportation Science*, 45(4):451âĂŞ464.

Tam, B., Ehrgott, M., Ryan, D., and Zakeri, G. (2011). Increasing flexibility of airline crew schedules. *OR Spectrum*, 33(1):49–75.

Van Den Bergh, J., Beliën, J., De Bruecker, P., and Demeulemeester, E. (2013). Personnel scheduling: a literature review. *European Journal of Operational Research*, 226(3):367âĂŞ385.

Veelenturf, L., Potthoff, D., Huisman, D., and Kroon, L. (2012). Railway crew rescheduling with retiming. *Transportation Research Part C: Emerging Technologies*, 20(1):95–110.

Walędzik, K. and Mańdziuk, J. (2018). Applying hybrid monte carlo tree search methods to risk-aware project scheduling problem. *Information Sciences*, 460-461:450–468.

Walędzik, K., Mańdziuk, J., and Zadrożny, S. (2015). Proactive and reactive risk-aware project scheduling. In *IEEE Symposium Series on Computational Intelligence*, pages 1–8.

Weide, O., Ryan, D., and Ehrgott, M. (2010). An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers and Operations Research*, 37(5):833–844.

Xie, L. and Suhl, L. (2015). Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum*, 37(1):99–136.

Xu, J., Sohoni, M., McCleery, M., and Bailey, T. (2006). A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems. *European Journal of Operational Research*, 169(3):978–993.

Yen, J. and Birge, J. (2006). A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14.

Yu, G., Argüello, M., Song, G., M. S., and White, A. (2003). A new era for crew recovery at continental airlines. *Interfaces*, 33(1):5–22.

Yu, G., Dugan, S., and Argüello, M. (1998). Moving toward an integrated decision support system for manpower planning at continental airlines: optimization of pilot training assignments. *Industrial Applications of Combinatorial Optimization*, 16:1–24.

Yu, G., Pachon, J., Thengvall, B., Chandler, D., and Wilson, A. (2004). Optimizing pilot planning and training for continental airlines. *Interfaces*, 34(4):253–264.