Airline reserve crew pairing optimisation

R. Janssen

TUDelft

KLM Royal Dutch Airlines

Airline reserve crew pairing optimisation

by

R. Janssen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 19, 2018 at 10:00 AM.

Student number:	4147413	
Thesis committee:	Prof. dr. R. Curran,	TU Delft
	Dr. ir. B. F. Lopes dos Santos,	TU Delft
	Dr. D. Ragni,	TU Delft
	Ir. drs. L. Scherp,	TU Delft

Preface

The next time I will be on an aircraft, without any doubt its cockpit crew will come to my mind. Those two or three or four persons behind the closed door in front of the aircraft, working another day in their tiny office. After ten months at the Cockpit Crew Services department of KLM Royal Dutch Airlines, I might be wondering if any of them really specifically wanted to fly this flight, for how long they have already been flying this aircraft type at this rank, or when they have last practised the crosswind approach of our flight in one of the flight simulators back home. But most likely, I will be asking myself if any of them was on reserve duty just hours before. He or she would have been waiting in the airline crew lounge at the airport, anxious about whether or not a reserve crew member would be required today. Then, his or her phone would have rung and the operations control centre would have been on the line, sharing details about the surprise flight that was ahead. From there on perhaps as little as one hour had passed until the departure of the flight that was just assigned. The pilot may have been happy with the destination or absolutely devastated, fact is that thanks to this reserve crew my flight could depart without delay.

This thesis is the final writing before my studies in Aerospace Engineering at Delft University of Technology are completed. It is concerned with my graduation research project about airline reserve crew pairing optimisation that has been performed in collaboration with KLM Royal Dutch Airlines, as part of a larger research project about crew scheduling. It has been a great opportunity to do my graduation research in-house at a company so relevant to my studies.

I would like to take the opportunity to acknowledge a few people who have supported me during the course of the research project.

Firstly, I want to thank my daily supervisor, Lennart Scherp, for the guidance provided during the last ten months. Apart from progress meetings and discussions about the topic, your project setup at KLM provided ideal working conditions for a thesis project. It has been fun to see the team expanding along the way, starting with two desks, later to four and now to six. It was a pleasure to contribute to your research on crew scheduling by investigating the airline reserve crew pairing problem, and I am curious to see where the project stands in a couple of years time.

Secondly, thanks to Bruno Santos, for the additional supervision during the project. The feedback during the monthly progress meetings has provided valuable. It has helped me to maintain a critical attitude towards the performed work and to keep the scientific scope of the project in mind.

Thirdly, my gratitude goes to all welcoming employees at the Cockpit Crew Services department of KLM. In particular, I would like to thank Nico Scheeres, Thea Groot, Shirah van den Hoek, Leon Ceelen and Taco Eyck for their direct involvement in the project, being able to answer questions at any time and provide valuable feedback from a practical perspective.

Last but not least, thanks to my family and friends who have supported me along the way. They have provided ample opportunity for pleasure and recreation during the course of this project, which has been really helpful in the long term.

R. Janssen
Schiphol, June 2018

Executive summary

In an operational airline environment, disturbances to the planned flight schedule cannot be avoided. Disturbances can have propagating disruptive effects in the schedule, because airline resources are all interconnected in airline networks (Belobaba et al., 2015). Consequently, airline schedules cannot be executed exactly as planned, resulting in inefficient use of airline resources (Barnhart et al., 2003; Barnhart and Cohn, 2004). Therefore, it is important to airlines that schedule disruptions are resolved.

Disruption management focuses on resolving schedule disruptions. Within disruption management, robust airline scheduling aims to introduce stability and flexibility in schedules such that disruptions can be absorbed. One possible robustness measure is the use of reserve crew. When a crew absence is reported after the crew schedule has been published, open crew positions arise in the schedule, causing the airline schedule to become inoperable. Reserve crew can be used to cover the open positions, so that the airline schedule becomes operable again. In European airlines, reserve crew generally consist of regular flight crew that have been assigned reserve pairings in their crew schedules (Nissen and Haase, 2006). That is, crew are periodically assigned reserve pairings instead of regular flight pairings. The airline reserve crew pairing problem consists of finding the set of reserve pairings, called a reserve pattern, that minimises the effect of expected disruptions to an airline flight schedule. The optimal number, length, and start times of the pairings should be determined in solving the airline reserve crew pairing problem.

Efficient reserve crew pairing is a difficult problem due to the inherent unpredictability of reserve demand. Given that crew costs are, next to fuel, the biggest expense for an airline (Belobaba et al., 2015), the importance of efficient reserve crew pairing is evident. Therefore, methods are required that predict the demand for reserve crew as accurately as possible. Given a predicted reserve demand, the reserve crew pairing problem involves finding a set of reserve crew pairings that covers the unpredictable reserve demand as well as possible. In other words, the mismatch between the expected reserve demand and the available reserve capacity should be minimised. The benefits lie in an increased reserve utilisation rate and a decreased flight cancellation rate due to crew absence (Bayliss, 2016), ultimately decreasing airline expenses.

Despite the importance of efficient reserve crew pairing, the problem has received little attention in scientific literature. The current state of the art has focused on solving the airline reserve crew pairing problem for cabin crew. For this, the flexibility in how reserve pairings can be defined is limited: reserve crew pairings are fully defined through pairing start times. It is assumed that the length of the reserve pairings is constant and known in advance. Moreover, reserve pairings are assumed to be fully dedicated to reserve duties. These assumptions can be challenged: reserve pairings may vary in length and they can be combinations of reserve duties and regular flight duties. These characteristics are especially relevant to *long-range cockpit* crew, because cockpit crew are the most expensive type of airline crew and long-range crew typically operate longer flight pairings than short-haul crew. To cover these long pairings, long reserve pairings are required as well. If these reserve pairings remain unused, the combination of long reserve pairings and expensive crew leads to a high waste of resources. If a (shorter) regular flight is included in the reserve pairing, then called a mixed reserve pairing, that flight can be executed if the reserve remains unused. This increased flexibility in reserve pairing definition can lead to higher reserve utilisation rates and lower waste of (unused) reserve resources.

However, the design of reserve pairings for long-range airline cockpit crew facing these characteristics has never been considered before. Therefore, this thesis addresses the airline reserve crew pairing problem for long-range cockpit crew. The goal is to make recommendations on how reserve patterns for long-range cockpit crew should be constructed in order to minimise the gap between expected reserve demand and scheduled reserve capacity.

To solve the airline reserve crew pairing problem, a research design framework is developed, with a reserve pattern evaluation model and four reserve pattern optimisation algorithms at its core. The evaluation model aims to determine specific performance measures of existing reserve patterns, which indicate the quality of reserve patterns. The evaluation model is used in combination with the optimisation algorithms, which iteratively generate and evaluate reserve patterns. The optimal reserve patterns generated by the optimisation algorithms are then compared to each other and to manually constructed reserve patterns. Based on these comparisons, recommendations can be made about how the gap between expected reserve demand and scheduled reserve capacity can be minimised.

The objective function of the optimisation process is defined as the sum of the number of reserve days in the pattern (i.e. the reserve budget) and the number of premium days, which are required to cover disruptions that cannot be covered by reserve pairings. A corresponding service level constraint is introduced that enforces a minimum reserve budget.

The evaluation model uses repeated airline flight schedule simulations to evaluate reserve patterns. It includes the functionality to evaluate reserve patterns that consist of reserve pairings with varying lengths and of mixed reserve pairings, which cause additional flight disruptions in the future. These are novel elements of the evaluation model. The model can simulate approximately a thousand weeks of airline operations in one second, for realistically sized problems.

The optimisation algorithms are a pure random search technique, a Learning Automata Search Technique (LAST), and two variants of a Greedy Randomised Adaptive Search Procedure, called GRASP and GRASP-LF. The random search algorithm generates reserve patterns by randomly selecting a subset of reserve pairings from a list of feasible reserve pairings, which are evaluated using the reserve pattern. Over a large number of iterations, the best reserve pattern is selected. The LAST algorithm is a novel adaption of an existing adaptive random search technique from Thathachar and Sastry (1987). It has been adapted so that airline reserve crew pairing problem specific characteristics could be utilised to improve the algorithmic performance. The GRASP and GRASP-LF algorithms are also adaptions from an existing construction based algorithm from Feo and Resende (1995). The novel GRASP and GRASP-LF algorithms for the airline reserve crew pairing problem utilise a concept called reserve pairing potential, which is a measure of the amount of premium days that can be prevented by a reserve pairing, to determine which reserve pairing should be added in each construction iteration. Compared to the GRASP algorithm, the GRASP-LF algorithm first covers the longest flight of each day in the airline schedule.

Three comparison experiments are done between the optimisation algorithms: (1) comparing performance by objective function value, (2) comparing performance by number of premium days while constraining the reserve budget, and (3) comparing static reserve pairing, where one reserve pattern is made for an entire season, against dynamic reserve pairing using the GRASP algorithm, where reserve patterns are made per month, including seasonality effects in crew absence rates.

In the first experiment, the GRASP algorithm achieves the best results. Compared to the manually constructed reserve pattern a 12.4% decrease in required working days is observed, which corresponds to a saving of 5.9 working days per week. Given the size of the total workforce of 104 FTEs, a total decrease of 1.1% of the total workforce costs can be projected. The primary factor in the objective function value improvement was shown to be a decrease in reserve budget. Next to the GRASP algorithm, the GRASP-LF and LAST algorithms also generate solutions that are better than the manually constructed reserve pattern.

In the second experiment, the GRASP-LF algorithm performs best, with a reduction of one premium day per week. The reason that the GRASP-LF algorithm outperforms the GRASP algorithm in this experiment is that the GRASP-LF algorithm covers more than the GRASP algorithm.

The final experiment shows that dynamic reserve pairing can be used to further reduce expected costs of reserve patterns. Compared to the static manually created reserve patterns a 25.3% improvement has been realised. The dynamic GRASP method was 14.3% better than the static GRASP method. However, the size of the data analysis that has been done to predict monthly absence rates, should be increased to obtain more accurate performance measures of dynamic reserve crew pairing.

Following the experiments, it is concluded that the GRASP and GRASP-LF methods are both viable methods to minimise the gap between scheduled reserve capacity and expected reserve demand. The GRASP method performs better when the reserve budget is unconstrained, whereas the GRASP-LF method should be used when the reserve budget is fixed. Additionally, it is found that dynamic reserve crew pairing can be used to further increase the efficiency of the reserve crew pairing process compared to current practice.

List of Figures

List of Tables

Contents

1 Introduction

In an operational airline environment, disruptions to the planned flight schedule are unavoidable. Possible causes are weather conditions, technical failures, crew absenteeism, and many others. Disturbances can have propagating disruptive effects in the airline schedule, because aircraft, crew, and passengers all are interconnected in airline networks (Belobaba et al., 2015). For example, a single aircraft delay can cause delays to other aircraft, cause misconnections to passengers, and cause reserve crew to be called in to replace delayed crew. Consequently, airline schedules cannot be executed exactly as planned, resulting in significant airline losses (Barnhart and Cohn, 2004; Barnhart et al., 2003; Kohl et al., 2007). Therefore, it is imperative that schedule disruptions are resolved.

Within airline scheduling, disruption management focuses on resolving disruptions. Proactive disruption management is concerned with the creation of robust schedules, which perform well in practice, given that disruptions are unavoidable. Robustness can be integrated in an airline schedule by various means, such as introduction of slack times in aircraft and crew schedules, increasing the number of resource swap opportunities, and the planning and scheduling of reserve crew and aircraft (Kohl et al., 2007; Lettovskỳ et al., 2000).

When a crew absence is reported after the crew schedule has been published, open crew positions arise in the schedule, causing the airline schedule to become inoperable. Reserve crew can be used to cover the open positions, so that the airline schedule becomes operable again. In European airlines, reserve crew generally consist of regular flight crew that have been assigned reserve pairings in their crew schedules (Nissen and Haase, 2006). That is, crew are periodically assigned reserve pairings instead of regular flight pairings. The airline reserve crew pairing problem consists of finding the set of reserve pairings, called a reserve pattern, that minimises the effect of expected disruptions to an airline flight schedule. Finding the optimal reserve pattern involves finding the number, length, and start times of the pairings.

Since cockpit crew costs are, next to fuel, the biggest expense for an airline (Belobaba et al., 2015), efficient reserve crew pairing is important to airlines. The benefits of reserve pairing optimisation are increased reserve utilisation rates and decreased flight cancellation rates due to crew absence (Bayliss, 2016). Efficient reserve crew pairing is a difficult process due to the inherent unpredictability of reserve demand, resulting from crew absenteeism and propagating effects due to the interdependence of airline schedule resources. The process is particularly difficult for intercontinental airline crew, who operate schedules in which flight pairings are generally long and of irregular length. Reserve patterns should be designed that cover these long irregular pairings to a maximum extent, given that the number of reserve resources is limited. Despite the importance of designing efficient reserve crew patterns for long-range cockpit crew, the problem has never been considered in academic literature before.

This thesis is involved with the airline reserve crew pairing problem for intercontinental cockpit crew. It aims to answer the following research question:

How should airline long-haul cockpit reserve crew patterns be constructed in order to minimise the gap between scheduled reserve capacity and expected reserve demand?

The research objective is to make recommendations on minimising the gap between scheduled reserve capacity and expected reserve demand of airline long-haul cockpit crew, by identification and evaluation of possible reserve crew pairing methods that utilise airline schedule data.

To solve the airline reserve crew pairing problem for long range cockpit crew, a novel reserve pattern evaluation model and a range of novel reserve pattern optimisation algorithms are developed. The evaluation model can be used to measure the quality of existing reserve patterns, giving the user detailed information about the strengths and weaknesses of existing reserve patterns. For the optimisation algorithms, four algorithms are implemented that iteratively generate reserve patterns and use the evaluation model to rate the quality of the generated reserve patterns, with the aim of finding reserve patterns with low expected costs. The optimisation algorithms are random search, an adaptive random search procedure called Learning Automata Search Technique (LAST), and two variations of a construction based algorithm, called Greedy Randomised Adaptive Search Procedure (GRASP) and GRASP-LF. The latter three algorithms are novel adaptions from existing default algorithms, in which specific characteristics of the airline reserve crew pairing problem are utilised to increase the algorithm performance.

Using the evaluation model and the optimisation algorithms, a comparison is made between the different algorithms in terms of their suitability towards creating high quality reserve patterns. Based on this, recommendations are made on how the gap between scheduled reserve capacity and expected reserve demand can be minimised.

This thesis is structured as follows. In Chapter 2 an investigation into existing literature related to the airline reserve crew pairing problem is presented. The literature study consists of two parts: (1) defining the research gap, and (2) identifying relevant methodologies towards solving the airline reserve crew pairing problem, also from other personnel scheduling domains. The high level research design of this project is presented in Chapter 3, after which the evaluation model and optimisation algorithms are detailed in Chapters 4 and 5, respectively. The setup and results of the experiments comparing the different optimisation algorithms are presented in Chapter 6. Afterwards, the findings of the model sensitivity analysis that has been done are shown in Chapter 7. Finally, the conclusions following from this project, along with a number of recommendations, are presented in Chapter 8.

2 Literature review

This chapter presents an exhaustive review of literature geared towards airline reserve crew. In particular, the airline reserve crew pairing problem is considered. In Section 2.1, a knowledge gap in the available literature concerning airline reserve crew pairing is identified. This section introduces the reader to the airline scheduling problem and increasingly narrows the scope until reserve crew pairing is considered. Afterwards, the state of the art is compared with the industry practice and based on this a knowledge gap is identified. With the gap known, feasible problem approaches are investigated in Sections 2.2 to 2.4. Because the amount of literature dedicated towards airline reserve crew is limited, personnel scheduling references from other domains are also consulted, focusing explicitly on stochastic problems. Possible problem approaches are decomposed into three parts: Section 2.2 presents various scheduling objectives and approaches to evaluate personnel schedules, Section 2.3 considers demand determination methods, and Section 2.4 discusses solution methods to generate personnel schedules. The three parts are integrated in Section 2.5, where a complete synthesis of the literature review is given.

2.1. Research gap

This section starts out with an succinct definition of the overall airline scheduling problem in Subsection 2.1.1. Next, crew scheduling is discussed more elaborately in Subsection 2.1.2, focusing specifically on robust crew scheduling and crew recovery (i.e. disruption management). The scope is then narrowed further in Subsection 2.1.3, zooming in on existing literature about airline reserve crew. After this, a gap in the literature is identified in Subsection 2.1.4.

2.1.1. Airline scheduling

Airlines face multiple challenging optimisation problems across the entire spectrum of their activities. Among these is airline scheduling, which involves designing aircraft and crew schedules to maximise airline profitability (Barnhart et al., 2003). These problems are perceived too complex to solve integrally due to factors such as network characteristics, crew regulations, maintenance requirements, dynamic operating environments, and problem size (Barnhart et al., 2003). Therefore, the airline scheduling problem is generally decomposed into several smaller problems that are solved sequentially (Barnhart et al., 2003; Belobaba et al., 2015; Clausen et al., 2010; Weide et al., 2010). The subproblems can be defined as follows:

1. **Schedule design** Definition of the airline flight schedule, including destinations, frequencies and flight times.
2. **Fleet assignment** Allocation of aircraft types to each flight in the schedule.
3. **Aircraft maintenance routing** Allocation of individual aircraft (i.e. aircraft tail numbers) to each flight in the schedule, while ensuring that aircraft maintenance requirements are satisfied.
4. **Crew scheduling** Allocation of crews to each flight in the schedule such that personnel costs are minimised.

The crew scheduling problem is typically subdivided into crew pairing and crew rostering (Kohl et al., 2007). In the crew pairing phase, round trips (pairings) from and to airline home bases are constructed. The pairings must cover all crew positions defined by the flights in the airline timetable. In the crew rostering phase the pairings are assigned to personal crew schedules (Kohl et al., 2007). On top of this, all other activities (e.g. simulator training, spare time) are allocated in the crew rostering phase.

2.1.2. Crew disruption management

Operational disturbances such as extreme weather, technical failures, or crew absenteeism cause disruptions to the planned airline schedule. Airline disruption management focuses on resolving schedule disruptions as effectively as possible (Kohl et al., 2007). Crew absenteeism can originate from various causes, among which are crew illness and missed crew connections or crew expiration due to delay (Belobaba et al., 2015). These resource shortages cause open crew positions in the airline flight schedule, resulting in an inoperable flight schedule. To cover open crew positions, reserve crew can be scheduled and utilised (Kohl et al., 2007). Thus, airline reserve crew and crew disruption management are related research topics. This section presents previous research in airline disruption management tailored towards crew scheduling, and identifies to what extent reserve crew have been included. First, robust crew scheduling is considered, after which crew schedule recovery is treated.

Robust crew scheduling

Robustness can be seen as proactive disruption management (Clausen et al., 2010): airlines anticipate that disruptions will occur and consequently incorporate a degree of stability and flexibility in their schedules. Stability of a schedule is defined as being able to continue operating as planned by absorbing minor schedule disruptions, whereas flexibility of a schedule is defined as the ability to recover easily and quickly after a schedule disruption has occurred (Ionescu and Kliewer, 2011; Dück et al., 2012; Soykan and Erol, 2014). Kohl et al. (2007) explain some techniques to incorporate robustness in an airline schedule:

- **Add slack time** to minimum connection times. This introduces stability, since small disruptions can be endured without making any changes to the schedule.
- Let **aircraft and crew stay together** as much as possible. This introduces flexibility, because knock-on effects of disruptions are minimised.
- **Synchronise crew** connections **and aircraft** connections. This introduces flexibility, since crew and aircraft can be swapped, allowing quick schedule recovery.
- Create **single return flight pairings** from and to the airline hub. This leads to a decreased probability of disruptions occurring at outstations, increased flexibility in cancellation of entire pairings, and efficient pooling of airline resources at the hub.
- Scheduling **reserve crew** and **reserve aircraft**, which can be used to increase schedule stability by absorbing disrupted resources. Kohl et al. (2007) mention the high costs incurred with this measure.

Ehrgott and Ryan (2002) aim to increase the scheduled ground time available for crew connections (i.e. introduction of slack times). A multi-objective optimisation is performed, acknowledging that an increase in robustness will lead to an increase in crew costs. They generate Pareto optimal solutions with varying degrees of robustness and costs. Schaefer et al. (2005) also aim to introduce robustness by increasing slack times of original schedules. They differentiate between the deterministic (theoretically optimal) and the stochastic (practically optimal) crew scheduling problem. Their model assumes that no flights are cancelled and that resource swapping is not possible: there are no interaction effects between aircraft and crews. Yen and Birge (2006) challenge this assumption. They solve the stochastic scheduling problem while taking into account dependencies between aircraft and crew, with the aim to minimise the propagated delay in the airline schedule. Dunbar et al. (2012) also aim to minimise the propagated delay, but where Yen and Birge (2006) only solve the crew scheduling problem, Dunbar et al. (2012) focuses on the integrated aircraft routing and crew scheduling problem. They claim that better decisions can be made when the dependencies between both problems are considered. Finally, Soykan and Erol (2014) object to minimise propagated delay within a crew schedule. Like Ehrgott and Ryan (2002), they consider a trade-off between robustness and schedule costs. Even though the scheduling and planning of reserve crew are possible measures to include schedule robustness (Kohl et al., 2007), none of the above references included reserve crew in their analyses.

Shebalov and Klabjan (2006) aim to increase schedule flexibility by increasing the number of possible crew swaps. Their objectives are to minimise crew costs and to maximise the number of crews that can potentially be swapped. Their model does allow for reserve crew, but it assumes that a reserve schedule is already known in advance. Gao et al. (2009) mention that crew scheduling should be integrated with aircraft routing to increase solution quality. Therefore, they consider crew connections within a fleet assignment model. Their aim is to limit the number of fleet types and crew bases (the crew home airport) that serve each airport, which increases the possibility of resource swaps. Dück et al. (2012) also solve the integrated fleet and crew scheduling model, but aim to maximise the number of possible crew swaps. They mention the use of reserve crew to increase schedule robustness, but do not explicitly model it. Similar to integrated crew and aircraft scheduling, Weide et al. (2010) try to increase schedule robustness by keeping aircraft and crew together to a maximum extent. Finally, Ionescu and Kliewer (2011) solve the stochastic crew scheduling problem with the aim to increase swap opportunities for crew. While the use of reserve crew is mentioned as a method to recover a disrupted schedule, it is not explicitly modelled.

Summarising, the majority of previous research has focused on robust crew scheduling, not on robust crew pairing. Common robustness measures are increasing slack times and the number of swap opportunities in crew schedules. Since the focus is on crew scheduling, reserve pairing has not been considered in the above references. Some mention the possibility of using reserve crew to increase robustness, but do not explicitly include the robustness measure in their solutions. At best, a set of already known reserve pairings is assigned to crews in the rostering phase.

Crew schedule recovery

Crew recovery can be seen as reactive disruption management: once a disruption has occurred, the schedule should be adjusted such that the original schedule is restored while minimising costs (Kohl et al., 2007). After a disruption has occurred, the schedule must be recovered, possibly by exploiting one or more robustness techniques. The crew recovery problem is similar to the crew scheduling problem. However, the objectives of both problems differ: in the crew scheduling problem one optimal solution is desired and the solution time is not restrictive, whereas in the crew recovery problem (multiple) acceptable solutions should be generated in a short time window (Clausen et al., 2010; Wei et al., 1997). To decrease the solution time, the problem size is decreased by considering small time windows (e.g. a couple of hours) and by limiting the number of crew considered for rescheduling (Clausen et al., 2010).

Wei et al. (1997) were the first to consider irregular operations in crew scheduling. They use a multi-commodity network flow problem to repair crew pairings. For each crew disruption, a candidate crew list is generated that contains the crews that can be swapped with the disrupted crew. Reserve crew are included in this candidate crew list. The reserve pairings (i.e. the number of reserves and their starting times) are assumed known in advance. A multi-commodity network representation is also used by Stojković et al. (1998), who aim to minimise crew costs and the number of deviations from the original schedule. Reserve crew are part of the set of candidate crew members that are considered for recovery, but the reserve pairings are required as model input. Yu et al. (2003) and Abdelghany et al. (2004) use similar problem setups, where the reserves are used as input to the list of candidate crew members under consideration. Nissen and Haase (2006) specifically focus on crew rescheduling at the duty period level instead of the pairing level (a pairing can consist of multiple duty periods). This leads to shorter rescheduling horizons, resulting in faster solution times. Nissen and Haase (2006) also assume a known set of reserve pairings that can be used in recovery.

The crew recovery problem has been integrated with other sub-problems of the airline scheduling problem. Stojković and Soumis (2001) and Stojković (2005) expand their original multi-commodity network model to include flight rescheduling. Even though these models are able to take reserve crew into account, they were solved without consideration of reserve crew. The combined crew and aircraft recovery problem has been approached by Le and Wu (2013) and Zhang et al. (2015). The problems were modelled as time-space networks with the aim of minimising recovery costs. Recovery measures included were resource swapping, crew deadheading, and delaying of flights. Use of reserve crew was also possible, but the reserve pairings were assumed to be known in advance by both studies. However, Zhang et al. (2015) did not use any reserve pairings in their experiments, because reference data could not be found. Finally, the integrated aircraft, crew, and passenger recovery problem has been the subject of research by Kohl et al. (2007), Abdelghany et al. (2008), Petersen et al. (2012), Maher (2015*a*), and Maher (2015*b*). They all assume that reserve schedules are available in advance. However, Abdelghany et al. (2008) mention that reserves are assumed to be available at any time the schedule of an originally scheduled crew member becomes invalid and no crew swaps are available. Petersen et al. (2012) do not explicitly model reserve crew, and assume that all crew that is on duty can be used.

Similar to robust crew scheduling, most references on crew recovery focus on crew rescheduling, not on the crew pairing problem. If reserve crew is considered in recovery models, it is incorporated by adding a set of reserve crew members to a set of candidate crew members, which can be used to replace or swap with disrupted crew. The set of reserve pairings is used as input for the recovery models, where it is assumed that the reserve pairings are determined in advance.

2.1.3. Airline reserve crew

The amount of research specifically dedicated to airline reserve crew is limited. The first research towards the topic has been conducted by Gaballa (1979), who optimises the balance between costs from reserve cabin crews and the costs of overnight delays at Qantas Airways. This is done using analytical methods, where the expected reserve demand originates from historical data. The expected amount of overnight delays per period is computed using conditional probabilities. Various sets of reserve pairings (differing in reserve numbers and start times) are compared to each other. Gaballa (1979) found that the number of reserve crews in operation was too conservative. However, the research was limited to a single case study for which a limited number of manually constructed reserve schedules was analysed. Methods to automatically generate reserve schedules were not developed.

Dillon and Kontogiorgis (1999) performed a research for US Airways to optimise the bid-lines of dedicated reserve crew. Through a set-covering approach all possible month long legal bid-lines were enumerated. A planner manually had to determine the reserve demand for each day, fleet type and crew rank. An integer optimisation model was used to determine the bid-lines to be used, where the quality of each bid-line was measured in the objective function.

The amount and start times of cabin crew reserves at KLM Royal Dutch Airlines (KLM) were determined by Paelinck (2001). An analytical model was developed to predict the remaining number of reserves per time period, based on conditional probabilities. The expected reserve demand per day, originating from flight disruptions, was derived from historical data. The expected remaining number of reserves could be computed per day, given that they are used to cover disrupted flights. A set of reserve pairings with known length and start times was required as model input. The model by Paelinck (2001) only served to evaluate sets of reserve pairings, while the creation of reserve pairings was not considered. Instead a trial and error approach was proposed to iteratively devise a set of reserve pairings that matches the expected demand.

Sohoni et al. (2004) focused on estimating long-term staffing levels of reserve crew by considering the cost trade-off between the amount of reserves and the amount of premium pay hours that must be endured when there is a reserve shortage. They also consider voluntary flying and reserve demand resulting from vacation and training activities. Using a similar line of reasoning, Sohoni et al. (2006) simultaneously address the planning and scheduling of airline reserve crew by predicting reserve demand based on open time trips resulting from bidding-invoked conflicts that are present in U.S. based airlines, and from disturbances in daily operations. This is followed by an optimisation where the expected reserve demand is covered with a minimal number of (dedicated) reserve crew. The optimisation is done in two phases: first the expected reserve demand is covered with a minimum amount of reserve duty periods, with lengths ranging from three to five days. Then, the duty periods are combined into month long legal work patterns. This research was conducted in collaboration with a U.S. airline. Therefore, the majority of reserve demand exists due to bidding-invoked conflicts (instead of due to operational disturbances, as in European airlines). Total reserve demand compared to European airlines is therefore much higher, which explains why up to 30% of the total crew in U.S. airlines may be dedicated reserve crew (Sohoni et al., 2006).

Another research in collaboration with KLM was performed by Bijvank et al. (2007), who also considered cabin crew reserve pairing at KLM. Their aim was to determine a good reserve strategy, consisting of the number of reserve pairings that have to start each time interval, and the configuration of the reserve pairings (i.e. the length and the placement of on-duty and off-duty days). To determine the number of starting reserves, they use a minimal flight reserve cover ratio (i.e. it should be able to cover at least a percentage of all flights starting at any point in time). For the configuration of the reserve pairings, several approaches were investigated. One of these was to set reserve pairing length equal to the length of the longest regular flight pairing at each start time. A statistical model (based on historical data of crew disruptions) was employed to estimate the number and length of reserve pairings so that with a certain confidence interval sufficient reserves are available. Note that contrary to U.S. based airlines, European airlines often do not have dedicated reserve crew. Reserve duties are covered by regular crew who are periodically scheduled to be on stand-by. Related to this, Bijvank et al. (2007) also consider the risk of secondary disruptions, which may occur when a reserve is used to cover a pairing that is longer than its original reserve duty length. The original pairing of this reserve after its reserve duty should then be covered by a second reserve. They account for this by scheduling soft flight pairings, with corresponding reserves (i.e. reserves are scheduled while keeping in mind that secondary disruptions may occur).

Recently, Bayliss (2016) wrote a doctoral thesis concerning the airline reserve cabin crew pairing problem under uncertainty, from which parts are presented in Bayliss et al. (2012), Bayliss et al. (2013), and Bayliss et al. (2017). In Bayliss et al. (2012), an analytical model is introduced that considers crew unavailability to evaluate a set of reserve crew pairings. The aim of the model is to evaluate the set of pairings by computing the total expected crew absence when an airline flight schedule is executed. Note that the analytical model is only used to *evaluate* the reserve pairings. Therefore it is combined with a number of solution methods to *generate* the sets of reserve pairings. The methods are dynamic programming, several (meta)heuristics and a number of simpler rule based approaches. The evaluation model was used during each iteration of the optimisation to compute the total expected crew absence for the set of reserve pairings. After optimisation, the generated sets of reserve pairings from the solution methods were validated by repeat simulations of airline flight schedules. It is mentioned that the probabilities for crew absence should be obtained from historical data, but the model was tested with uniform crew absence probabilities.

A similar methodology has been applied to consider crew related delay instead of crew absence in Bayliss et al. (2013). Instead of using historical data to predict crew absence, a simulation model is used to estimate (propagated) delay probabilities, given that other recovery actions such as resource swapping are used first. With these delay probabilities known, the reserve pairings are generated similarly to Bayliss et al. (2012), now aiming to minimise the expected delay. Again, the generated reserve pairings were evaluated by repeat simulations. In the thesis (Bayliss, 2016), the two previous models were integrated to account for simultaneous crew absence and crew delay.

A different approach is introduced in Bayliss et al. (2017), where disruption scenarios are generated using airline operations simulations that account for journey time and crew presence uncertainty. A set of disruption scenarios is then used as input to model a mixed integer programming model, which has the objective to find a reserve crew schedule that minimises the expected overall disruption level for the set of input scenarios. Contrary to the probabilistic approaches, this approach simultaneously solves for a reserve schedule and a reserve use policy (i.e. which reserve should be used at what time).

Throughout the research of Bayliss (2016), reserve pairings are only defined through the starting times of reserve crew. The length of reserve pairings is a fixed value of two weeks. Furthermore, it is assumed that reserve pairings consist entirely out of reserve duties. Bayliss (2016) states that this definition is representative of KLM practices, from which data was acquired.

These assumptions can be challenged: reserve pairings can be variable in length and they may contain regular flight duties. This leads to increased flexibility in creating reserve schedules, which is especially relevant to long-range cockpit crew. There are two reasons for this. Firstly, cockpit crew are the most expensive human resources for an airline (Bayliss, 2016), indicating the need for cost efficient reserve crew pairings. Secondly, as a result from longer flight duties and crew rest requirements, long-haul crew pairings are generally longer than short-haul pairings. Reserve crew pairings should be able to cover the long-haul flight pairings in full, resulting in a high number of unused reserve days (i.e. a waste of resources) if the reserve pairing remains unused. If a regular flight duty is included as part of the reserve pairing, that flight can be executed when the reserve remains unused. Additionally, the variability in the length of the regular flight pairings requires that reserve pairings are also variable in length to attain a cost effective set of reserve crew pairings. No previous work has been found that approaches the reserve crew pairing with these problem characteristics.

2.1.4. Defining the research gap

In airline disruption management, the use of reserve crew is widely recognised as a measure to recover from operational disruptions. However, crew recovery literature assumes that the reserve crew pairings are already known. In robust airline scheduling, scheduling reserve crew is named as a measure to increase the robustness of an aircraft schedule. Yet, the majority of the literature aims to introduce robustness by increasing slack times in schedules and by increasing crew swap opportunities. The design of the pairings for airline reserve crew is often assumed to be solved already. In short, reserve crew pairing optimisation is overlooked, even though crew are expensive resources (Kohl et al., 2007).

Specific research on airline reserve crew has focused on planning and scheduling problems for reserve crew. The integrated planning and rostering approach by Sohoni et al. (2006) considers the problem at a tactical level, by producing monthly schedules of on-duty and off-duty days. The work of Paelinck (2001) and Bayliss (2016) is dedicated to airline reserve crew pairing, focusing on cabin crew. The reserve pairings in these references are defined only through the pairing start times. It is assumed that the length of the reserve pairings is constant and known in advance. This assumption is too conservative: reserve pairings can vary in length from day to day, or even on the same day. Additionally, previous research assumes that the reserve pairings are entirely dedicated to reserve duties. Yet, reserve pairings may be combinations of reserve duties and regular flight duties.

In practice, it is found that these characteristics for reserve crew pairing (variable pairing length and inclusion of regular flight duties) are most relevant to *long-range* (i.e. intercontinental) *cockpit* crew. A number of reasons can be given for this. Firstly, cockpit crew are the most expensive type of crew (Bayliss, 2016) for an airline. Secondly, intercontinental pairings generally have a higher duration than short-haul pairings. The full length of these longer pairings should be covered by reserve pairings in a reserve schedule. If the reserve pairings consist entirely of reserve duty days, there is a high waste of resources when the reserve remains unused. If a (shorter) regular flight is included as part of a reserve pairing, that flight can be executed if the reserve remains unused. On top of this, it may be advantageous to vary the reserve pairing length, depending on the length of the pairings in the flight schedule. This yields a higher reserve utilisation rate and a lower waste of resources, which is important for scheduling costly cockpit crew.

These characteristics, specifically observed in long-range cockpit reserve crew pairing, have never been applied to the airline reserve crew pairing problem before. Hence, the research gap is defined as the design of reserve pairings with variable reserve pairing lengths and combined reserve and flight duties.

The remainder of this literature review focuses on solution approaches to the airline reserve crew pairing problem. The problem can be decomposed into three parts: (1) schedule objective and evaluation, (2) demand determination, and (3) capacity allocation. Sections 2.2 to 2.4 consider each of these parts sequentially. In Section 2.5, a synthesis of all methodologies from Sections 2.2 to 2.4 will be given. Here, it will be shown that combining methodologies from each part leads to complete problem approaches for the airline reserve crew pairing problem.

Because the amount of literature dedicated towards airline reserve crew is limited, references from other personnel scheduling domains are also consulted, focusing explicitly on stochastic problems. This is in line with the characteristics of the airline reserve crew pairing problem: the amount of reserves required depends on the probabilities of flights being disrupted. That is, the reserve crew pairing problem is stochastic.

2.2. Scheduling objectives and evaluation

The quality of personnel schedules should be evaluated based on assessment criteria. These criteria are expressed in objective functions whose values express the quality of the schedule. This allows to compare the quality of schedules generated by different solution methods independently to each other. In Subsection 2.2.1 a number of optimisation objectives are presented. Subsection 2.2.2 discusses evaluation of complex objective functions. A synthesis of the approaches presented in this section is given in Subsection 2.2.3.

2.2.1. Scheduling objectives

The quality of a personnel schedule is measured by an objective function. This section presents a number of objectives that are of interest to reserve crew pairing. The general objective function formulation of the reserve crew pairing research of Bayliss (2016) is shown in Equation 2.1. The objective is to minimise the expected value of disruptions E, given a set of reserve pairings, a reserve use policy, an airline schedule, uncertainty, and an airline recovery policy.

$$\min_{x \in X, y \in Y} E(f(x, y, S, U, P)) \tag{2.1}$$

where

x	=	the set of reserve pairings from all possible sets of reserve pairings X
y	=	reserve use policy from the set of reserve use policies Y
S	=	the airline flight schedule
U	=	uncertainty in the airline operations
P	=	the airline recovery policy

The paper by Bayliss et al. (2012) focused purely on crew absence as the source of disruptions. Therefore, the objective was to find a set of reserve pairings that minimised the resulting crew absence probabilities. A number of objective functions were compared: the minimum of the sum of all probabilities, of the maximum probability, of the standard deviation, and of the coefficient of variation. Experiments were conducted with each of these objective functions, and the total sum of absence probabilities over the entire flight schedule was found to yield the lowest cancellation rate and the highest reserve crew utilisation rate. Therefore, Bayliss et al. (2012) used this objective function for further research.

The expected reserve crew utilisation rate and the expected cancellation rate were derived from crew absence probabilities (Bayliss et al., 2012). The first is computed using Equation 2.2, where p_i is the probability of crew absence for flight i, $p_i^{'}$ the same probability after taking into account the reserve schedule, and R the total number of reserves. The cancellation rate is defined as the number of flights without crew over the total number of flights N, as shown in Equation 2.3. For both the crew utilisation rate and the cancellation rate, expected theoretical values and experimental values were computed.

$$\text{Expected reserve crew utilisation rate} = \frac{(\sum_{i=1}^{N} p_i) - (\sum_{i=1}^{N} p_i^{'})}{R} \tag{2.2}$$

$$\text{Expected cancellation rate} = \frac{\sum_{i=1}^{N} p_i^{'}}{N} \tag{2.3}$$

The paper of Bayliss et al. (2013) has similar objectives to Bayliss et al. (2012), but focused on crew related delay instead of crew absence. The objective was to find a set of reserve crew pairings such that the total expected crew related delay was minimised. On top of the cancellation rate, the reserve crew utilisation rate, and the solution time, the average crew delay, the average total delay and the probability of delays larger than 30 minutes were used as assessment criteria. Given that the objective was to minimise crew related delay, the average crew delay was used as the main performance measure (Bayliss et al., 2013).

Bayliss et al. (2017) described a simulation scenario based method aimed to minimise the total cancellation measure over all disruptions in all scenarios. The disruptions included both delays and cancellations. To devise a single performance measure for both delays and cancellations, delays were converted to a cancellation measure, where the severity of the delay influenced the value of the cancellation measure. The equation for the cancellation measure for a single departure in a single scenario cm_h is given in Equation 2.4, where td_h is the total delay of departure h, cd_h the crew related delay at departure h (i.e. the delay that can be absorbed by reserve crew), CT the cancellation threshold at which delayed flights are cancelled, and n the delay exponent, which allows the decision maker to set a balance between absorbing delays or cancellations.

$$cm_h = \left(\frac{td_h - cd_h}{CT}\right)^n \tag{2.4}$$

Throughout the research by Bayliss (2016), the objective was to minimise the total number of expected operational disruptions, given a fixed capacity of reserve crew. Even though reserve crew utilisation was used as a performance measure, a low reserve utilisation rate is not explicitly penalised. Therefore, it is impossible to investigate the effects of changing number of reserve crew, other than through trial and error. In other words, the optimal balance between reserve capacity and reserve demand is not being considered.

In contrast, the research towards reserve bus driver scheduling by Jönsson (1987) does consider the balance between capacity and demand. In the objective function a weighted sum of average costs of unused reserve bus drivers and the costs of cancelled bus tours is considered. The first part increases with the size of the reserve capacity, whereas the second part decreases with the size of the reserve capacity. Therefore, an optimum can be found for a certain reserve capacity. This allows for an implicit sizing of the reserve capacity and thus find an optimal balance between capacity and demand that minimises the total expected costs. A fixed ratio between the costs of cancelled tours and unused reserves was defined as input for the model.

A parallel to the airline reserve crew pairing problem can be seen here. The costs for unused airline reserve crew and the costs for cancelling a flight (or paying premium fee to crew working overtime) can be defined. A balance between the amount of reserve crew and the amount of cancelled flights can then be found such that the overall costs for both factors are minimised.

The objective of minimising costs while satisfying a predefined service level is widely used in references concerning call centre problems (Alfares, 2007; Cezik and L'Ecuyer, 2008; Ertogral and Bamuqabel, 2008; Ingolfsson et al., 2010). This can be translated to the airline reserve crew pairing problem: a service level can be defined for a certain flight schedule that expresses the percentage of time periods during which sufficient reserve crew should be available to cover flight disruptions. The required reserve levels for a flight schedule can be determined to satisfy the service level. Service levels can be defined globally (i.e. for a complete schedule) or locally (i.e. per discrete time period). The global approach may result in variations of the service level over time, whereas it is expected that reserves are overstaffed to guarantee the minimum service level in each period with locally defined service levels.

In the paper by Bijvank et al. (2007), a statistical model was developed using an equivalent to a service level. The minimum required reserve capacity so that the probability of requiring more reserves than available is below a threshold, had to be determined. Three assessment criteria are defined by Bijvank et al. (2007): (1) the expected number of unused reserves; (2) the expected number of secondary disruptions; (3) the expected number of disruptions the cannot be covered by the reserve schedule.

Schedule cost minimisation is widely used in personnel scheduling literature as an objective in the optimisation. For example, Sohoni et al. (2006) aim to minimise total crew costs while covering all expected uncovered trips. In addition, over-covered trips (i.e. trips with possibly redundant crew allocated) are penalised. As an assessment criteria, the number of uncovered trips is used (Sohoni et al., 2006). Cost minimisation is also used as an objective in Elshafei and Alfares (2008), Yan et al. (2008), and Parisio and Jones (2015), among others. Here, each crew is given a certain cost to be scheduled. It is expected that this approach is not directly applicable to the airline reserve crew scheduling problem. This is because reserve crew are not dedicated reserve crew, but are simply regular flight crew that are sometimes allocated reserve duties. No direct extra costs are incurred for reserve duties, which makes a direct cost minimisation difficult.

2.2.2. Schedule evaluation models

For optimisation problems encountered in a stochastic setting, it is difficult to directly formulate an objective function (Gosavi, 2015). For example, this can be caused by non-linear or probabilistic effects. Separate evaluation models can be created for these problems that find the objective function value. These models can either be analytical models or simulation models.

Analytical evaluation models

Paelinck (2001) developed an analytical model to evaluate a set of reserve pairings. The objective was to calculate the number of remaining (unused) reserves, which was calculated using conditional probabilities. The probability P_u of using a reserve crew on a given day was computed as in Equation 2.5. Then, the probability P_a of a reserve still being available n days after the start day was computed as in Equation 2.6. The expected number of remaining reserve crew on a single day was found by summing the result of Equation 2.6 for all reserves covering that day.

$$P_u = \frac{\text{average \# of reserves used per day}}{\text{\# of daily reserve crews}} \tag{2.5}$$

$$P_a = (1 - P_u)^n \tag{2.6}$$

By performing these calculations for each day in the schedule, the quality of the reserve pairings can be evaluated. To increase the accuracy, Equation 2.6 was also evaluated with hourly time periods. A drawback of this model is that it used constant usage probabilities per day, whereas in reality there may be large fluctuations over the day. Also, this model does not take into account dependencies between overlapping reserve pairings (i.e. the use of a reserve pairing may reduce further demand).

The model from Paelinck (2001) served as the starting point for the research of Bayliss et al. (2012), who researched the airline reserve crew pairing problem for cabin crew. The objective was to generate a set of reserve pairings that minimises the overall probability of crew absence. The model requires crew absence probabilities Q for each individual departure in the flight schedule which are assumed to be independent from each other. The model then calculates the resulting crew absence probabilities P, given that absent crew can be replaced by reserve crew. For each flight j covered by each reserve k the reserve availability probability r_k^j and crew absence probability p_j were updated according to Equations 2.7 and 2.8.

$$r_k^{j+1} = r_k^j(1 - p_j) \tag{2.7}$$

$$p_j = p_j(1 - r_k^j) \tag{2.8}$$

By propagating through all reserve pairings, the total effect of reserve crews on the absence probabilities can be calculated. Due to the overlapping reserve crews, the objective function is a non-linear function, which is evaluated step-by-step by the evaluation model. The model considers crew absence as the only source of reserve crew demand, and reserve use as the only recovery measure. However, given that Bayliss et al. (2012) include short-haul flights in the research, this model is not a realistic representation of the problem. Especially for short-haul traffic, crew delay can be a significant factor in reserve demand, and there are additional recovery measures such as resource swapping.

The crew absence model (Bayliss et al., 2012) was adapted in Bayliss et al. (2013) to model crew delay probabilities. Similar to crew absence, crew delay can be expressed considering delay distributions from historical data. Added difficulties for this model are incorporation of delay propagation and resource swapping. Therefore, the delay probabilities for all crew members were derived through a simulation. The probabilistic model from Bayliss et al. (2013) then calculates the resulting delay probabilities under a set of reserve pairings. The probabilistic model accounts both for direct delays and delay that propagated through the schedule.

In Bayliss (2016), the previous models are extended further into a statistical delay propagation model. This model considers both crew absence and crew delay. In fact, this model takes the output of the crew absence model as an input, and then accounts for delay propagation, while including the possibility of crew and aircraft swaps to recover from disruptions. To capture both crew absence and delay in one model, its complexity and required evaluation time increase accordingly. In fact, the crew delay induced reserve demand was too demanding in terms of computational time when used iteratively with an optimisation method. Therefore, crew absence was used as the only reserve demand source for all models. Note that the evaluation models by Bayliss (2016) are already too computationally demanding when there is one type of reserve pairing. It is expected that model complexity increases when reserve pairings of variable length and with regular flight duties must be included.

In the call centre domain, queuing models are employed to evaluate service levels for given staffing levels. In Section 2.2 it is said that the objective generally concerns a cost minimisation while satisfying minimum service levels. The service levels are expressed as non-linear constraints in the mathematical model. Hence, instead of using an evaluation model to measure the objective function, a queuing evaluation model is used to measure the value of the service level constraints.

Ingolfsson et al. (2010) use this technique iteratively to solve the call centre staffing problem. Generated staffing levels are evaluated by a queuing model that evaluates non-linear service level constraints. Because these constraints are non-linear, it is possible that generated schedules do not satisfy these service levels. At time periods where the service level is not reached, a scheduling model tries to improve the staff schedules at these points. The improved schedules are then tested in the queuing model again.

Simulation evaluation models
Cezik and L'Ecuyer (2008) and Avramidis et al. (2010) solve comparable problems which focus on call centres where agents have multiple types of skills. Agents are part of groups that possess a fixed subset of all available skills. The goal of the research is to minimise the operating costs under a minimum service level. The service level is defined such that a fraction of calls that is answered within a certain time limit, exceeds a set threshold. The service levels are defined as non-linear functions. Cezik and L'Ecuyer (2008) mention that the service levels functions are too complicated to evaluate exactly. Hence, they revert to a simulation model that can estimate the service levels. Based on the results of these simulations, additional constraints are added to improve the service level approximation. By this method, the optimal solution can be approximated, and convergence to the optimal solution is only fully obtained if the number of simulated scenarios goes to infinity. However, this would significantly increases the required solution time.

Gurvich et al. (2010) also utilise a simulation evaluation model to solve a call centre staffing problem. In an iterative manner, a simulation model evaluates the service level constraints for a certain staffing level. When any of the constraints is violated, the staffing levels are updated and re-simulated to check for constraint violation.

Bayliss (2016) developed a simulation model with the purpose of validating and comparing reserve crew schedules that were generated through a variety of solution methods. The model simulated the operations of an airline schedule in a single hub and spoke network. The simulation propagated through all flights in the schedule in the order of departure. For each departure, crew are either present or absent, based on a statistical distribution. In case of crew absence, depending on reserve crew availability, a flight is either cancelled or covered by reserve crew. The simulation then checks if the flight is delayed due to propagated effects of previous events. If the delay is above a certain threshold, possible recovery actions are considered, or the flight is cancelled. Otherwise, the flight is executed, with a flight duration that is based on a statistical distribution as well.

To obtain reliable comparison results, 20,000 repeat simulations were run for each set of reserve pairings that was evaluated. The performance measures that were tracked included the cancellation rate, the reserve utilisation rate, and the amount of delay. These performance measures correspond to the performance measures presented in Section 2.2 and thus express a measure for the quality of the generated solution. Through this, the performance of the analytical model could be validated via the simulation model.

The simulation model was not used to evaluate reserve pairings during the solution process (i.e. to evaluate the objective function during optimisation). It is unknown why this approach was not pursued, given that the analytical models became too complex when representing a realistic operational environment. In these cases, simulations can be used to evaluate the solution, without a closed expression for the objective function (Gosavi, 2015).

2.2.3. Synthesis of scheduling objectives and evaluation
This chapter has focused on scheduling objectives and on schedule evaluation. A synopsis of possible scheduling objectives is given below:

- Minimise resource shortages;
- Minimise combined unused resources and resource shortages;
- Minimise required resources while satisfying service level;

- Minimise schedule costs.

The research by Bayliss (2016) aimed to minimise the eventual number of expected disruptions, given that a set of reserve pairings can be used to cover these disruptions. The pairings are evaluated by measuring the number of disruptions that could not be covered by the reserve pairings. In other words, resource shortages are measured. This objective is useful when the available reserve capacity (i.e. the number of reserve days that can be allocated) is fixed.

If capacity requirements are relaxed, a combined minimisation of unused resources and resource shortages is a feasible alternative. The objective function is a weighted sum of unused resources and resource shortages, where one part increases with the number of resources and the other part decreases. This has two of advantages. First, automatic sizing of the resource capacity is possible. This gives the decision maker insight in the required reserve capacity for cost effective operations. Second, the weights for each part can be adjusted, so that the decision maker can express preferences for one of the objective parts.

Schedule cost minimisation can also be the objective for personnel scheduling problems. In these problems, each crew has a certain cost when it is scheduled. The aim of the optimisation model is to find the set of crews that meets the minimum staffing or service levels with the lowest overall costs. This approach is not feasible for the airline reserve crew pairing problem, because airline reserve crew are simply regular crew that are sometimes allocated reserve pairings. No direct extra costs are incurred for executing reserve pairings compared to regular flight pairings.

Finally, analytical and simulation schedule evaluation models were discussed. These models can be employed when the objective or constraints are too complex to express in a single function. They are generally used iteratively during the optimisation process, to calculate the value of the objective function or a constraint. Whether these models will be useful in solving the airline reserve crew pairing problem, depends on the problem approach that is pursued.

2.3. Demand determination

Section 2.2 presented a range of personnel scheduling objectives and methods to evaluate schedules, whereas Section 2.4 will cover methods to create schedules. However, before schedules can be created, the demand for the resources in the schedule (i.e. required staffing levels) should first be known. This section discusses how this demand can be determined.

The demand for airline reserve crew originates from absence of originally scheduled crew. Crew absence is uncertain, for example due to illness, meaning that the demand for reserve crew is stochastic. In other domains for personnel scheduling, demand for resources can also be stochastic. For example, in retailing or in the call centre domain, customer arrivals are often random. Therefore, it is possible to identify approaches from these domains and translate those to the airline reserve crew pairing problem.

2.3.1. Demand data distributions

Required resource levels can be estimated by analysing historical demand data. Parisio and Jones (2015) do this when considering an employee scheduling problem for a number of retail stores. They analyse 44 weeks of historical data on customer arrival patterns to estimate the demand for employees. The arrival distribution patterns are later used to determine the set of optimal work shifts for a fixed number of employees. Parisio and Jones (2015) mention the importance to include demand fluctuations and employee absences as significant external factors that should be considered in employee schedule design.

Bayliss et al. (2012) analyse seven years of historical crew absence data to determine the probability that a crew member is absent for each flight in a flight schedule. These probabilities are assumed to be independent from each other, and to be the only cause of reserve crew demand. Therefore, this data distribution could be used to estimate reserve crew demand. Bayliss et al. (2013) derived flight duration distributions from historical data, but the size of this data set was only one month. These distributions were used to estimate reserve crew demand originating from delayed flights.

Instead of deriving a demand distribution directly from historical data, the data can also be fitted to a theoretical probability distribution, which is then used to generate a demand profile. Yan et al.

(2008) take this approach when determining the optimal design of work shifts for an air cargo terminal. It is stated that the majority of the research towards shift design problems assumes deterministic demand profiles. However, the performance of these deterministic models is reduced when stochastic demands occur in actual operations. Therefore, stochastic demand should be taken into account to maintain optimal scheduling solutions (Yan et al., 2008). The demand distribution was based on a normal distribution, where the mean and standard deviation were based on one month of actual manpower demand data.

Similarly, Jönsson (1987) created a model to determine the number of bus driver reserves for a Swedish bus company. This is the only application of reserve crew shift design in other modes of transportation that was found. The demand distribution follows from variation in demand over time resulting from the company timetable and unpredictable short-term absenteeism of bus drivers. A Poisson probability distribution was used for the absence of bus drivers in each time period, which is expressed by Equation 2.9. In this equation, X_t represents the number of absent drivers in period t and λ_t the absence rate of period t.

$$p_t(X_t = k) = \frac{\lambda_t^k}{k!}e^{-\lambda_t} \tag{2.9}$$

Vassilacopoulos (1985) solved the problem of allocating doctors to shifts in an emergency department. It is mentioned that the patient arrival rate varies considerably in these environments. An analysis of patient arrival data of a hospital revealed that the hourly patient arrival rate could be approximated by a Fourier series. There were three harmonics, with periods of 8, 12, and 24 hours that caused varying arrival rates throughout the day. On top of this, statistically significant differences between individual days of the week were found, as well as seasonality effects. The seasonality effects were not included in generating the demand patterns, because the time span of the problem was only one week.

The planning problem for a financial centre across multiple locations with limited office space and constrained recruitment capacity, under fluctuating workforce demand was addressed by Zhu and Sherali (2009). They compared planning models both under deterministic and stochastic demand and found that using stochastic demand resulted in schedules that required less adaptions during operations. For the stochastic demand profiles, a normal distribution was assumed where the average was based on demand data provided by the financial centre. The standard deviation was assumed to be twice the mean.

Demand profiles can also be generated from synthetic probability distributions, where the parameters are assumed instead of derived from historical data. This approach is taken by Pinker and Larson (2003), who determine staffing levels in a general service environment with flexible working hours and uncertainty in demand and employee presence. A Poisson probability distribution was used to determine the amount of work arriving in each time period, and a binomial distribution was used to determine employee absence. This data is purely theoretical though: the work arrival rate is assumed to have a mean of 20, where units are not specified.

Similar approaches are taken in two examples of the widely studied nurse scheduling problem. Gnanlet and Gilland (2009) compare various combinations of resource flexibility (bed upgrades and cross-trained and external nurses) to satisfy stochastic patient demand. The demand was defined as the number of patients per hospital unit. These values were generated from a uniform probability distribution. Campbell (2012) considers the scheduling of on-call overtime shifts for nurses to decrease costs and increase the quality of service. The demand was directly defined as the number of nurses required per shift, with values derived from a probability distribution with a mean of 30 nurses per shift. The type of distribution was not specified.

Easton and Goodale (2005) and Easton (2011, 2014) wrote a series of papers about workforce scheduling considering uncertain demand and uncertain employee presence. Cross-trained employees are also taken into account. Here lies a parallel to the airline reserve crew pairing problem, where airline crew are allowed to fly below their rank (e.g. a captain flying as first officer), and crew may be trained to operate multiple aircraft types (Bayliss, 2016). Easton (2014) mentions that cross-training employees helps to increase service levels by being able to pool resources, leading to increased productivity levels. The models are constructed assuming that probability distributions for demand profiles and employee attendance are known in advance. The demand profiles are presented as a general random variable with a probability density function and cumulative density function, whereas the absence rate is expressed as a mean value over all periods.

2.3.2. Demand modelling

The previous section describes how demand data distributions can be expressed. These distributions do not always directly express the required staffing levels per time period. For example, they may represent a customer arrival distribution or an employee absence distribution. This section shows how demand models can use such distributions for further demand determination. First, analytical models are presented and afterwards simulation models are considered.

These methods are applied differently compared to the evaluation models presented in Section 2.2. Those models are used to evaluate generated personnel schedules, either iteratively during the solution process or afterwards for validation purposes. Here, the models are used before solving the problem to determine a demand profile which is used as input for the solution process.

Analytical demand models

Jönsson (1987) used a Poisson distribution to express the number of short term absent bus drivers in each time period. This distribution was used in combination with a minimum service level that is defined as the percentage of scheduled buses that will be in operation specified per time period. Through this, the minimum number of drivers in the buffer (i.e. reserve) required in each period is derived from the desired service level, given the Poisson distributed crew absence variables.

A similar approach is taken by Vassilacopoulos (1985), who proportionally assigns a fixed available capacity of doctor hours corresponding to arrival rate of patients in each time period. Equation 2.10 is a simplified representation that was used to determine the number of doctors per time period.

$$q_t = \frac{N}{\lambda}\lambda_t \quad t = 1, 2, \ldots, T \tag{2.10}$$

In this equation q_t is the number of doctors required in period t, N is the total number of available doctor hours per week, T is the number total number of periods in the scheduling horizon, λ_t is the patient arrival rate in period t, and λ is the average patient arrival rate over all periods. By a dynamic programming algorithm, these variables were converted to integer numbers.

Bijvank et al. (2007) used a statistical model to determine the number of airline reserve cabin crew at KLM required in each time period. Their model uses a binomial distribution to ensure that the probability of requiring more reserve capacity than allocated is small (for example 5%). This is a comparable approach to the service level that was used in Jönsson (1987).

In the call centre domain, the arrival rate of incoming calls is often regarded as the stochastic variable (Pot et al., 2008; Gurvich et al., 2010; Ertogral and Bamuqabel, 2008). On top of this, the required time to serve a customer varies with a certain distribution and customers can be put in a waiting line before they are served. Also, multiple types of personnel are often considered: those specialised in one type of service and those that are cross trained to handle multiple call types. To include these effects, the staffing requirements are often determined in two steps (Ertogral and Bamuqabel, 2008):

1. Estimate the number of incoming calls for each period in the planning horizon. This is determined through a call arrival rate (Ertogral and Bamuqabel, 2008).
2. Given a certain service level requirement, determine the number of agents that are required in each time period. The service levels are usually expressed in terms of customer waiting times. The staffing levels are generally determined through queuing or simulation models (Alfares, 2007).

Alfares (2007) aimed to determine the minimum number of agents required in the short-term staffing of a call centre on an hourly basis while satisfying a specified service level. The workload was derived from five months of historical data on the number and the duration of calls received in each hour. This data was applied in a queuing model that could determine the required staffing levels per hour. A 10% buffer was added to each time period to account for employee absenteeism, based on annual vacations, training, and illness. This number was taken from the standard policy of the case study company. Alfares (2007) did not have any data available on the distribution of inter-arrival times of the customers. This is given as the primary reason why no simulations were used to predict the hourly demand. It is mentioned that simulation models are more flexible for this purpose, but require that all stochastic parameter distributions are fully known.

A queuing model was also used by Ertogral and Bamuqabel (2008) in the call centre environment. Given an hourly customer arrival pattern from historical data, a constant arrival rate for each time period is approximated. The minimum number of agents to serve these customers is based on a queuing model under a pre-specified service level.

Simulation demand models

Instead of using analytical models, simulations can be used to derive demand levels. This approach is common in the formulation of stochastic programming models (Shapiro and Philpott, 2007; Shapiro et al., 2009), where scenarios are constructed by random realisations of the stochastic variables. In solving these problems it is aimed to find a solution that works well for the entire set of scenarios. A challenge is to find realistic representations of the demand distributions, while aiming to keep the size of the scenario set, and corresponding problem size, tractable (Yan et al., 2008; Zhu and Sherali, 2009; Parisio and Jones, 2015).

Bayliss et al. (2017) took a similar approach for the airline reserve crew pairing problem. Through simulations of an airline schedule without any reserves a set of disruption scenarios was generated. For each of these scenarios, sets of feasible reserves to solve the disruptions in these scenarios were constructed. These sets were later used to select the best possible reserves such that the total delay and cancellations in the airline schedule were minimised. It is mentioned that the number of scenarios has a large impact on both the quality and speed of any scheduling methods that are used in this approach. Therefore, a scenario selection heuristic (SSH) was designed. The aim of this heuristic was to make a smart selection of disruption scenarios, such that the operating environment can be represented accurately with a smaller set of scenarios compared to adding each disruption scenario that is simulated.

2.3.3. Rule-based methods

Aside from the approaches that have been described in Subsections 2.3.1 and 2.3.2, some other methods to quantify demand have been identified. Bijvank et al. (2007) use a minimal flight reserve cover ratio, which is computed using Equation 2.11.

$$\frac{\sum_j T_{jk}}{\sum_j S_{jk}} \geq \alpha, \quad \forall k \tag{2.11}$$

Here, $\sum_j T_{jk}$ is the number of starting reserves of type j on day k, $\sum_j S_{jk}$ is the number of flights of type j on day k, and α is the cover ratio. The type j can involve characteristics as length, rank, and aircraft type. Bijvank et al. (2007) reported that a cover ratio of 4% is used in practice as well as in their demand estimation.

Dillon and Kontogiorgis (1999) developed a model to automatically create monthly schedules for reserve crew at US Airways. To determine the demand over the course of the month, an employee from the planning department manually specified a demand target on each day of the month for all combinations of crew base, aircraft type, and crew posting.

In other domains, similar methods were used by Davis and Reutzel (1981) and Elshafei and Alfares (2008). Davis and Reutzel (1981) consider work processing at a bank, where the demand per time period is variable and work that has not been processed in previous periods must be taken into the next period. Even though the demand per period is allowed to vary, it is predetermined as model input. Elshafei and Alfares (2008) consider a general personnel scheduling problem with the unique characteristic that employee wages are dependent on scheduling decisions made previously. With respect to the demand determination they assume that required labour demands per time period are known in advance.

Bayliss (2016), who considered the airline reserve crew pairing problem for cabin crew, mentions that it is not necessarily required to determine the expected demand at all. In theory, it is possible to allocate a fixed, predetermined, capacity without any knowledge of expected demand. However, this was shown to perform badly in closing the gap between demand and allocated capacity.

2.3.4. Synthesis of demand determination methods

In this section various methods to determine the demand in a personnel scheduling environment have been presented. A synopsis of these methods is given below:

- Derive demand distributions from historical data;
- Assume synthetic demand distributions;
- Use analytical models to determine demand levels;
- Use simulation models to determine demand levels;
- Use simple rules to express demand;
- Assume known demand;
- Do not take any demand into account.

The use of historical data allows to represent demand profiles based on operating environments in practice. An advantage is that corresponding solutions are relevant to the problem encountered in practice. A disadvantage of this approach is that the solutions have only been shown to be effective for one data set. Another difficulty is that the data should contain the right information: the required parameters should be available directly or be derivable from other data. The data set should also be large enough to obtain reliable parameter values.

Synthetic data distributions counter the disadvantages above. Diverse operating environments can be represented by varying the parameters in these distributions, such as arrival rates, averages and standard deviations. It is also easier to implement synthetic data distributions, because no data analysis is required. On the other hand, it can not be ensured that the data distributions are representative for practical problems. Combining real and synthetic data can improve overall results: a range of synthetic data distributions can be used to ensure that solution methods are robust to diverse demand profiles, while data from practice can be used to validate the results.

Separate models can be developed to determine the demand levels for scheduling problems. These models can be used when the data distributions described above do not directly express personnel requirements. Both analytical and simulation models can be developed. Commonly used analytical models are queuing models, where customer arrival rates and minimum service levels are model input to determine the required staffing levels per time period. Simulation demand models are used to generate demand scenarios which are random realisations of the stochastic variables. A set of scenarios is then used to represent the demand distribution. The challenge for scenario generation models is to find accurate representations of data distributions while keeping the scenario set small, in order to keep the problem size tractable.

Demand models are generally used before solving the optimisation problem (i.e. they are used to derive model input). Therefore, their use is limited to scheduling problems where the decision variables do not influence the demand themselves. In the reserve crew pairing problem, the opposite occurs when reserve pairings with regular flight duties are used. When such a reserve pairing is used to cover an open position, another open position is created further on the horizon (the regular flight duty cannot be executed anymore). Because the scheduling decisions influence the demand profile, demand models are not feasible for the airline reserve crew pairing problem.

The use of one of the miscellaneous methods from Subsection 2.3.3 to determine demand can be useful during the development stages of the research. When the demand representation is simplified, it is easier to verify correct functioning of the models. However, if high quality reserve schedules are to be obtained, these approaches are expected to be too simplistic.

2.4. Capacity allocation

This section explores solution methods that are used to create personnel schedules. Methods from both the airline reserve crew domain and the general personnel scheduling domain will be presented. Subsection 2.4.1 introduces various representations for the decision variables in a scheduling problem. The values of these variables are decided on by expressing the variables in a model and solving it with a solution method. Exact solution methods are presented in Subsection 2.4.2, heuristic methods are presented in Subsection 2.4.3, and dynamic programming approaches are treated in Subsection 2.4.4. Subsection 2.4.5 introduces some solution methods aimed specifically at simulation optimisation. A number of simpler approaches toward capacity allocation are discussed in Subsection 2.4.6. Finally, a synthesis and reflection of all presented methods is given in Subsection 2.4.7.

2.4.1. Schedule representation

Depending on how an optimisation model is set up, the decision variables can have varying meanings. In scheduling problems, the prevailing strategy is to first generate all allowable unique shifts or tours (a tour consists of multiple shifts and includes days off). Each unique shift or tour is then represented by a decision variable, which expresses the number of employees working that shift or tour. A vector with values of each decision variable representing a shift or tour expresses a scheduling solution (Avramidis et al., 2010).

Shifts can be distinct from each other through differences in start time, duration, required employee qualifications, or type of work. For example, Ingolfsson et al. (2010) allow shifts to take 4, 6, or 8 hours and start at the beginning of each discrete time interval. Breaks can be of varying duration and placed flexibly in a shift, leading to a maximum of 288 allowable shifts. Similarly, Avramidis et al. (2010) define shifts based on length, start time, and brake placement. On top of this, two agents types were defined. Therefore, the decision variables represented the number of agents per type having a certain shift.

Easton (2011) defines a tour as a cyclic work schedule for one employee over a typical planning horizon of one to four weeks. A tour consists of five consecutive workdays per 40 hour workweek. Shifts take 9 hours with a one hour break after 4 hours of work. Shifts can start at any hour of the day, given that they end on the same day. In the airline domain, Dillon and Kontogiorgis (1999) generate employee bidlines with one month duration for reserve flight crews at US Airways. An integer variable was included for each bidline in the model. The total number of bidlines led to about 15,000 decision variables in the model.

Easton (2011) stresses that the number of decision variables rapidly grows with increased scheduling flexibility. The tour definition of Easton (2011) only includes one shift type, to keep the problem size tractable. However, Van den Bergh et al. (2013) mention that current scheduling problems tend to prioritise employee satisfaction by offering increased flexibility in part-time contracts, flexible work hours and employee preferences. It is challenging to incorporate this required flexibility in problem representations like these, where a trade-off exists between scheduling flexibility and the number of decision variables.

Bayliss (2016) assumes a fixed number of reserve crews with equal pairing length. Therefore, the scheduling decisions reduce from determination of the number of employees per shift or tour, to determination of the start time of each of the reserve crews. Thus, in Bayliss (2016) an employee schedule was represented by a vector of reserve duty start times.

2.4.2. Exact solution methods

Alfares (2007) presents an integer programming model to solve the call centre staffing problem, where the number of employees assigned to various weekly tours must be determined. The number of required employees per time period was determined by a queuing model. A constraint was added to ensure that the scheduled capacity exceeded the number of required employees per period. A similar model is presented by Ertogral and Bamuqabel (2008), who used a combination of a queuing model and a simulation model to determine the number of employees per time period per skill type. On top of the general scheduling model, an extended scheduling model is presented that allows to schedule flexible workers. A case study that was done with the flexible worker model with an hourly requirement of approximately 50 agents, 168 time periods, and 1792 allowed schedules, was solved using the branch-&-bound algorithm to optimality in roughly 1.5 hours of computational time. It is noted that further research should be aimed towards a more computationally efficient method to solve the flexible agent case.

Cezik and L'Ecuyer (2008) represent an employee call centre staffing problem with multi-skilled employees through an integer programming model. Requirements on minimum service levels are expressed through non-linear constraints. To handle the non-linearity of the constraints, simulations are performed to estimate the service level. Corresponding linear constraints were added using the cutting plane method. The authors report that adding the cuts is complex, and numerous heuristics are employed to simplify the process. It is mentioned that when the problem size grows, the time required for solving the integer program grows exponentially. To counter this, they propose to relax the integrality constraints and round up any non-integer solutions. Cezik and L'Ecuyer (2008) report on an example problem with 5 call types and 12 agent skill groups. A single iteration of the IP model was solved using the simplex algorithm of the CPLEX optimisation suite in about a minute. After each iteration, new service level estimates should be obtained and linear constraints should be added. The simulations required for estimation of the service level were the dominant factor in computational time, leading to a total solution time of approximately 6 hours.

Avramidis et al. (2010) report that the iterative approach used by Cezik and L'Ecuyer (2008) yields suboptimal solutions. They propose to integrate the simulation phase and the integer programming problem with cut generation. In some cases better solutions were obtained, but a price was paid in terms of computational complexity. For a problem with 52 time periods and 123 possible schedules, computational times of more than 10 hours are reported, even with LP relaxations. Again, most of the computational effort was spent on getting a representative set of scenarios. For this problem 35,000 simulations were performed, and the authors still experienced difficulty to find feasible solutions.

Bayliss et al. (2017) applied mixed integer programming to the airline reserve cabin crew pairing problem. This integer programming model, called the mixed integer programming simulation scenario model (MIPSSM), uses disruption scenarios that were generated through a simulation model, as has been explained in Chapter 2.3.2. The aim of the model is to solve over the set of disruptions scenarios to find the best set of reserve crew pairings. Best in this context is defined as the schedule that minimises the combined effect of delays and cancellations in a simulated airline schedule. Example problems were solved both with and without the scenario selection heuristic (SSH), based on a 3 day flight schedule with 243 flights, 148 teams of crew, 37 aircraft, and 11 reserve crews, was considered. Without the SSH, 50 disruption scenarios were generated from this flight schedule, and the optimal reserve schedule was found through the branch-&-cut algorithm in CPLEX in an average of 28 minutes over 20 repeats (Bayliss et al., 2017). With the SSH, only 15 disruption scenarios were included, and the same problem was solved in under 3 minutes, with comparable performance.

The schedules derived from the MIPSSM showed a relatively high spread in solution quality over all repeats (Bayliss et al., 2017). This may indicate that the number of scenarios included in the MIPSSM has not been large enough. There is a delicate balance here between a large set of disruption scenarios and the exponential increase in problem size (Bayliss et al., 2017). For further research, it is suggested that the MIPSSM may be solved faster by a heuristic procedure.

Scheduling problems with stochastic demand can be represented by stochastic programming models (Shapiro et al., 2009). Yan et al. (2008) mention that two-stage stochastic models with recourse are often used to solve scheduling problems under uncertainty. It is stated that in two-stage optimisation problems, the decision variables are split into two groups. The first stage of decision making occurs before random demand has occurred. That is, these decisions can be taken independent of uncertain parameters. In the second stage, decisions are taken when random demand is known. Usually the uncertainty is expressed through a set of scenarios that are individual realisations of the demand. Optimal solutions are those that perform the best over all scenarios.

Yan et al. (2008) mention the importance of taking uncertainty into account when creating scheduling solutions. They state that the performance of schedules created assuming known demand can be reduced when applied to actual, stochastic operations. In their stochastic programming model, the first-stage decision variables are set to be staffing variables and shift variables (i.e. the number of personnel, and the shifts that they must work). In the second stage the decision variables are expressed as excess manpower variables and insufficient manpower variables. The optimisation model then minimises the number of man-hours given that the assigned crew meets the manpower demands of each simulation scenario, during each time slot in the planning horizon. Example problems were solved with approximately 700 man-hours to be scheduled and 6 different shift starting times. 60 scenarios were simulated to represent the stochastic demand. To decrease solution times, the integrality constraints were relaxed. The problem was solved to optimality by the simplex algorithm in CPLEX, in about one hour, but it was noted that an increase in the number of scenarios leads to sharp increases in the model run time.

A comparison for solving the personnel scheduling problem at a financial centre by an integer programming approach and a stochastic programming approach was made by Zhu and Sherali (2009). The integer programming model assumed a deterministic demand profile, whereas the stochastic programming model used two-stage decisions where scenarios represented random realisations of the demand data. These variations were based on a normal distribution, where the average was derived from historical data. An example problem with 11 skill groups, 6 locations, and 12 time periods was solved by the branch-&-cut algorithm in 22 seconds. For the stochastic programming approach only

15 scenarios were used to vary the demand. A Benders' decomposition was done to decompose the problem, avoiding memory issues. An optimal solution was reached in over 10 hours. However, the results obtained from the stochastic model were actually worse than from the deterministic model (more adaptions to the original schedule were required). Other example problems with more scenarios (100+) yielded better results for the stochastic approach, at the cost of even higher run times for comparable problem sizes.

Easton (2011, 2014) developed stochastic programming models to solve personnel scheduling problems in general service environments while facing uncertain demand and employee attendance. Similar strategies are used as in Yan et al. (2008) and Zhu and Sherali (2009): in the first stage optimal staffing and scheduling decisions are made without random realisations of demand and employee absence known. Then, based on realised demand and attendance scenarios, the cross-trained employees are re-allocated to different times and locations, in order to hedge against these uncertainties. Easton (2011) mentions that this approach leads to increased scheduling flexibility, which is an efficient method to mitigate the effects of employee absenteeism. The way in which the uncertain parameters are realised is different for both approaches. Easton (2011) uses simulations to express the distributions for demand and attendance, whereas Easton (2014) uses predetermined estimates of the expected sales numbers, given a certain personnel schedule and a certain realised demand and employee attendance profile. Both models were solved through the branch-&-cut algorithm, but no computational efforts required to solve any of the hypothetical example problems were reported.

2.4.3. Heuristic solution methods

The airline reserve crew pairing problem is a combinatorial optimisation problem (Bayliss, 2016). These kind of problems have a finite solution space, which may quickly become too large to enumerate fully (Bianchi et al., 2009; Yang et al., 2012). Various exact algorithms for combinatorial optimisation problems are subject to exponentially increasing solution times, which may lead to computational intractability. Heuristic algorithms are designed to derive high quality solutions with limited computational effort, without guarantee of finding a global optimum (Bianchi et al., 2009). In other words, computational performance can be gained at the cost of solution accuracy (Yang et al., 2012).

This subsection investigates the applicability of heuristic solution methods to the airline reserve crew pairing problem. First, the use of heuristic algorithms towards stochastic and dynamic scheduling problems is presented, after which hybrid heuristics are introduced.

Applications to stochastic scheduling problems

Bianchi et al. (2009) mention in a survey paper that the application of metaheuristics to stochastic combinatorial optimisation problems is a recent, but growing research area. However, most of the applications that are considered in this survey are stochastic travelling salesman problems and stochastic vehicle routing problems. The use of heuristics to solve personnel scheduling problems with stochastic demand is limited.

Easton and Mansour (1999) applied a genetic algorithm to personnel scheduling, with the aim to find a single procedure to solve general set covering, and deterministic and stochastic goal programs. Genetic algorithms are population based metaheuristic algorithms based on evolutionary principles (Bianchi et al., 2009). In Easton and Mansour (1999), it is mentioned that the workforce demand in the deterministic problem is usually determined by separate analysis techniques, such as queuing models. This corresponds to the demand models presented in Section 2.3. The stochastic problem in Easton and Mansour (1999) is modelled similarly to a stochastic programming model. For each realisation of the demand, staffing constraints are defined for each time period. It was shown that for the general set covering problem, the genetic algorithm was a factor 10 faster and closer to the true optimum than a branch-and-bound algorithm. For the stochastic scheduling problem, the genetic algorithm was compared to simulated annealing and tabu search metaheuristics, which were slightly outperformed in solution quality by the genetic algorithm, for a comparable solution time.

The workforce scheduling problem with uncertain demand and variable employee productivity has been approached by Thompson and Goodale (2006). They claim the linear representations of staff scheduling problems are often inaccurate due to the stochastic nature of demand arrival. To cope with the stochastic demand, Thompson and Goodale (2006) use a set of linear approximation functions, implemented by a look up table. The resulting problem was solved by a variety of simulated annealing heuristics, which yielded results that were close to optimality. The authors did not report any computational efforts required to solve the problem, nor did they attempt to compare the algorithms to exact or other heuristic approaches.

Bayliss (2016) is the only reference to apply heuristic solution methods to the airline reserve crew pairing problem. The objective was to minimise the number of unresolved disruptions in a flight schedule, given that reserves can cover disruptions. A variety of heuristic solution methods was used in combination with analytical models to evaluate the objective function. Below an overview and comparison between the heuristic applied by Bayliss (2016) is given. Afterwards, the heuristics are compared to each other. Martí and Reinelt (2011) distinguish between construction methods and improvement methods. Construction methods are heuristics that build up a solution from nothing, whereas improvement methods are heuristics that start with some feasible solution and, step by step, aim to improve the initial solution (Martí and Reinelt, 2011).

The Forwards Heuristic allocates reserve crew pairings one by one at starting times that decrease the resulting probability of disruptions the most. The Backwards Heuristic starts with reserve crew assigned to each departure in the flight schedule. Reserves are then removed one by one until a solution is found that has the maximum number of allowable reserve crews. Note that the resulting schedules from both the Fowards Heuristic and the Backwards Heuristic may be different, since the non-linear objective order may be sensitive to the order in which reserves are added or removed.

The Basic Greedy heuristic builds a complete solution before evaluating the schedule. It directly allocates reserves to flights with the highest original disruption probability, until all reserves are assigned.

The improvement heuristics below are divided into local search based methods and population based methods. Local search based methods explore solutions that are in the neighbourhood of the current solution (i.e. solutions that only slightly differ from the current solution). Each iteration, the solution moves to the most promising neighbouring solution, by which a path through the solution space appears (Martí and Reinelt, 2011). All local search algorithms used the cut and insert neighbourhood, which corresponds to shifting the start time of one reserve crew.

Hill Climbing is a basic local search heuristic. From an initial solution, it moves towards the best neighbour if and only if it is better than the current solution. A considerable drawback of this approach is that this search method tends to get stuck in local optimal solutions, which may perform relatively bad from a global point of view.

The Tabu Search metaheuristic, as originally proposed by Glover (1989), uses the same neighbourhood as the Hill Climbing heuristic, but to avoid getting stuck in local optima a tabu list is incorporated in the algorithm. This tabu list prevents that previously made moves are made again within a short term. Also, non-improving moves are accepted, which means that the search path can escape from local optima. In Bayliss (2016), 200 iterations were used and the tabu list had a length of 50 iterations.

Another metaheuristic employed is Simulated Annealing. This method is based on the cooling (annealing) process of metals, aimed to intensify the search towards later stages of the solution procedure (Kirkpatrick et al., 1983). Bayliss (2016) uses a temperature reduction every 4 iterations, and a temperature reduction factor of 0.999. The initial and final temperatures are 3 and 0.001, respectively. Non-improving moves may be selected with a certain probability, which decreases as the search process progresses.

The Variable Neighbourhood Search method was proposed as a new metaheuristic by Mladenović and Hansen (1997). It uses a multitude of predefined neighbourhoods in sequence, in combination with a local search algorithm. The neighbourhoods are used in a fixed order and in each iteration a random solution from the current neighbourhood is generated. From this solution, a local search technique is applied to improve the best solution found so far. If no improving solution can be found, the search progresses to the next neighbourhood in the sequence. If a better solution is found, the first neighbourhood in the sequence is used again. The solution procedure ends when the final neighbourhood in the sequence yields no improving solutions. Bayliss (2016) used five neighbourhoods, but did not explain how these neighbourhoods work.

Genetic algorithms belong to the class of population based algorithms, which perform search processes that describe the evolution of a set of solutions in the search space, instead of the trajectory of a single solution, as in local search methods (Blum and Roli, 2003). Genetic algorithms can be defined as computational models of evolutionary processes (Blum and Roli, 2003), where solutions are defined as strings of genetic code (Bayliss, 2016). These strings are mutated to create offspring, which base most of their genetic code on their parents, but are changed according to certain crossover rules. The idea is that parents with a high fitness level (i.e. a favourable objective function value) create fit offspring. At each iteration the least fit strings are eliminated, which guides the search process towards a global optimal solution (Blum and Roli, 2003). For the reserve crew pairing problem, Bayliss (2016) uses single point crossover and a mutation rate of 0.001 (i.e. one in a thousand genes gets changed). The population size is 100 and 100 iterations are performed (Bayliss, 2016). It is mentioned that the mutations may lead to unfeasible solutions, where the incorrect amount of reserve crew was scheduled. This was solved through application of the Forwards Heuristic or Backwards Heuristic which added or removed reserves one by one.

Ant Colony Optimisation was originally proposed by Dorigo et al. (1996), and is based on the foraging behaviour of ants (Blum and Roli, 2003). Ants are able to efficiently find shortest paths between food sources and their nest by leaving pheromones on these paths, which are sensed by other ants. As a system converges, shorter paths gradually obtain a higher concentration of these pheromones, leading to increased activity by the ants on these paths. Dorigo et al. (1996) describes the depositing of pheromones as a means to provide positive feedback, which allows for rapid discovery of good solutions. Also, Dorigo et al. (1996) says that the use of a colony is a means of distributed computation, which prevents early convergence towards a possible local sub optimal solution. Bayliss (2016) describes a system with 100 ants where each ants visits a number of reserve duty start times equal to the total number of reserve crew. That is, each route of an ant defines a complete set of reserve pairings. Based on the cumulative distribution of disruptions, a certain amount of pheromones are deposited. Over 200 iterations, 5% of the pheromones is removed on each route in each iteration. Through this scheme, improved sets of reserve crew pairings emerge as paths with large amounts of pheromones on them (Bayliss, 2016).

All of the heuristic procedures described above, were compared to each other and to an enumeration algorithm by Bayliss (2016) in a series of 20 example problems. The algorithms were compared by objective value (sum of disruption probabilities) and required solution time. The enumeration algorithm required 1296 seconds per problem to obtain optimal solutions. The Tabu Search and Genetic Algorithm methods were able to derive optimal solutions. The Tabu Search method required 32 seconds, and the Genetic Algorithm required 9 seconds. The Simulated Annealing and Variable Neighbourhood Search methods also yielded low optimality gaps, but slightly worse than the Tabu Search algorithm with similar run times. The Hill Climbing heuristic yielded near optimal objective function values with run times of only a couple of seconds. The other heuristics all performed worse than the aforementioned ones.

Hybrid heuristic methods

The solution methods in Bayliss (2016) have been used to create sets of reserve crew pairings that are defined only by their start times. The performance of these methods in terms of objective function values and required solution time is unknown when reserve pairings vary in length and are allowed to include regular flight duties. This added flexibility increases the size and changes the structure of the solution space, which can influence the effectiveness of above heuristics. For example, the local search algorithms use a neighbourhood definition that only allows changing pairing start times. This does not allow a full exploration of the solution space when the pairings are defined more flexible.

Van den Bergh et al. (2013) find that hybrid heuristic techniques are increasingly being used to solve personnel scheduling problems. These techniques combine multiple (meta)heuristic algorithms into new solution procedures, aimed to increase the overall method effectiveness (Van den Bergh et al., 2013; Blum et al., 2010). Blum and Roli (2003) describe that intensification and diversification are central concepts in driving such heuristic methods towards high performance. Intensification is defined as adaptation of the search strategy to encourage move combinations and solution features that were found to perform well before, whereas diversification is defined as encouraging the search process to visit regions of the solution space that have not been visited before (Glover and Laguna, 2013). Hybrid heuristics have not been considered by Bayliss (2016), so it is unknown if their application to the airline reserve crew pairing problem is an effective way of solving it, in terms of solution quality and time. Below, some examples from the general personnel scheduling literature are presented that combined multiple (meta)heuristic methods and could be translated to the airline reserve crew pairing problem.

Brusco and Jacobs (1998) apply a simulated annealing algorithm to general personnel scheduling problems. Their aim is to minimise employee costs while satisfying a minimal staffing requirement over a number of time periods. They claim that simulated annealing algorithms require high computational effort before converging to an optimal solution. To converge to the optimal solution, separate procedures were defined that removed (drop routines) and added (build routines) employees from and to the schedule. The drop routines were based on the sum of understaffing that would result from dropping each active tour from the schedule. This sum was divided by the costs of the tour, where the tour with the smallest ratio was removed. The build routines were based on the sum of understaffing covered by a tour, divided by the costs of the tour. The tour with the largest ratio was added to the schedule. It was found that the best solutions generated were 28% closer to the lower bound than a practical solution implemented for airport ground scheduling.

Burke et al. (2003) describe a framework based on a variable neighbourhood search approach that is used to solve nurse scheduling problems. They describe that large neighbourhoods promote diversification and that small neighbourhoods promote intensification. They also propose the use of neighbourhood-algorithm pairs. For large neighbourhoods, fast solution algorithms are required so that interesting regions in the solution space are quickly identified. On the other hand, thorough algorithms should be used for smaller neighbourhoods that can intensely search for optima. For each neighbourhood-algorithm pair, unique stopping criteria are defined. Burke et al. (2003) use a multitude of neighbourhood-algorithm pairs in a variable neighbourhood search framework, which was shown to yield favourable schedules in highly constrained environments. They conclude that for problems with complex search spaces problem specific neighbourhoods can be developed that increase the applicability of regular heuristics. They claim that it is beneficial to perform intensive local search in the immediate surrounding of a schedule until an optimum is found. Afterwards, the exploration of wider environments by varying the neighbourhood is advised.

The usage of a search framework that includes multiple optimisation algorithms to balance diversification and intensification is explored by Xu et al. (2015). They developed a teaching-learning based algorithm consisting of three phases to solve a job-shop scheduling problem where the job processing times are stochastic. In the first (teaching) phase a genetic algorithm is used that creates offspring from the best solution (the teacher) and each other solution (the students) in the population. In the second (learning) phase a fraction of the best students are crossed-over with each other. In the third (studying) phase, a greedy local search is used to improve the best solution so far. A comparison in solving some benchmark problems with a number of existing algorithms demonstrated the effectiveness of the proposed algorithm.

2.4.4. Dynamic programming

Hillier (2012) defines dynamic programming as a modelling technique to make a sequence of interrelated decisions, where an optimal combination of decisions must be found. It is mentioned that no standard mathematical formulation exists for dynamic programming problems, but that careful understanding of a problem structure is required to determine whether or not a problem lends itself for dynamic programming.

A basic dynamic programming problem consists of a number of stages, where a decision is required at each stage. At each stage, a number of states describe the various conditions that a system can be in. A decision must be taken at each stage that transforms the system from a current state to a state corresponding to the next stage. The aim of dynamic programming is to find a series of decisions (an optimal policy) for the overall problem, so that an objective is achieved (Hillier, 2012).

A multi-stage problem is faced by Davis and Reutzel (1981), who consider processing operations at a bank. This problem structure is dynamic because work may be backlogged: when not enough staff was available in a previous period to process all incoming work, the work that is leftover may continue into the next period. The objective function is a cost minimisation, where costs are incurred both from staffing numbers and unprocessed work at the end of the day. Davis and Reutzel (1981) define the stages as the number of periods until the end of the operations. The state of the system is defined through the number of employees who started their shift prior to the current stage and the amount of processing backlog. The amount of backlog was discretised to decrease the number of possible states, which was necessary to prevent a computationally intractable problem. It is recognised that the number of unique shifts that continues into future stages grows rapidly, and an increase in the number of shifts may lead to excessive computing times. The decision variables were defined as the number of employees that would start a shift in the current stage. A case study with 9 stages and 9 types of shifts was solved for two step sizes in the discretisation of the backlog levels. The number of possible combinations for staff levels ranges from 38,760 to 593,775. Solution times were reported of about 5 minutes for the largest problem instances. However, in terms of solution times and solution quality, the dynamic programming model did not outperform an integer programming approach (Davis and Reutzel, 1981).

The allocation of doctors to shifts in a hospital emergency department was solved through a dynamic programming approach by Vassilacopoulos (1985). The stages were defined as the number of time periods in the scheduling horizon, whereas the number of states followed from the change in staffing levels with respect to the previous period, which is a strategy to reduce the possible number of states of the system. Vassilacopoulos (1985) did not report any solution times required to solve the problem.

Jönsson (1987) applied dynamic programming to determine the number of reserve bus drivers. The aim is to find the policy that minimises the costs for unused bus drivers, the costs for cancelled bus tours, and the costs for changing the reserve buffer level. The state of their model is defined through the current buffer level, the total number of reserves that started before the current stage, and the total number of buffer level changes up to and including the current period. The decision variable is defined as the change in buffer level size in the current period. The minimum shift length was defined to be 4 hours in sequence. Through this requirement, the number of possible decisions in each stage was implicitly constrained, which was important to keep solution times within acceptable limits. It is also noted that the number of allowed changes in the buffer level can be constrained manually if the shift length limitation is not sufficient. No results were given with respect to solution times for the examples that were considered. A comparison with other approaches or the current practice in industry has not been done either.

With respect to the reserve crew pairing problem, Bayliss et al. (2012) describe a pruned dynamic programming algorithm to determine the optimal starting times for reserve cabin crew. The stages are defined through the departure number of flights in an airline schedule, and the states are defined by the number of reserve crew assigned. The decision variables are defined binary: whether a reserve is scheduled at this departure or not. To reduce the size of the solution space upper bounds and lower bounds are heuristically estimated. Therefore, Bayliss et al. (2012) describe that this method is essentially a heuristic algorithm, but with a high probability of attaining optimal solutions. It is stated that this method can be made faster and more ruthless, or slower and more careful by tuning the upper and lower bounds. The aim of the model was to minimise the sum of cancellation probabilities for all flights in the schedule, given a set of reserve pairings. Therefore, this algorithm was used iteratively with the analytical evaluation model (Bayliss et al., 2012; Bayliss, 2016) that has been described in Section 2.2. Optimal solutions were obtained for an example problem with 25 stages and 9 available reserves in 38 seconds (3% of the time required for full enumeration). Note that the definition of reserve pairings is rigid: it is only defined by the pairing start time and the pairing length is fixed. For reserve pairings with variable lengths it is expected that the number of decisions and states increases exponentially, because there is more flexibility in defining possible pairings.

Elshafei and Alfares (2008) describe a general personnel scheduling problem with a unique cost structure, where employee costs are dependent on scheduling decisions made in previous stages. Therefore, dynamic programming is applied to solve this problem. Their objective is to minimise the total labour costs. The stages are the days in the planning horizon and the states are combinations of feasible days-off schedules defined for all employees at a certain time period. The amount of employees is considered fixed, and determined based on rule-based methods and a certain demand. The authors do not report on the total number of states in each stage, which makes it difficult to judge the computational efficiency of this method.

Dynamic programming can also be applied when the outcome of decisions is uncertain. These are Markov decision problems, which can be termed as dynamic programming where decisions are made under uncertainty (Koole and Pot, 2005). This approach is taken by Pinker and Larson (2003), who try to determine the number of regular and flexible employees that should be employed such that labour costs and the costs incurred by backlogging work are minimised. They consider uncertainty in demand and in employee absenteeism, where each employee has an independent probability of being present. Different decisions are taken in even and odd stages: in even stages the usage of contingent workers is decided on, in uneven stages decisions are made on worker overtime. Pinker and Larson (2003) mention that solving the dynamic programming model can be computationally intensive. It is noted that every time the problem size doubles, the expected run time increases by a factor 16.

Problems in the call centre environment have been solved using dynamic programming by Koole and Pot (2005). The objective of this research was to obtain good call routing policies (i.e. which agent skill group answers which call). Koole and Pot (2005) mention that a shortcoming of dynamic programming is that for high-dimensional systems (e.g. call centres) the state space becomes too large, leading to problems that are computationally intractable. In the literature, this is known as the curse of dimensionality (Bertsimas and Demir, 2002; Powell, 2009). Therefore, Koole and Pot (2005) revert to approximate dynamic programming (ADP). Powell (2009) coins ADP as an umbrella for modelling techniques that solve problems that are large, complex, and usually stochastic. The essence of ADP is that the function that expresses the value of a state is approximated by a statistical function. It is recognised that the effectiveness of ADP strongly depends on the specific problem structure, and how that structure is exploited. Powell (2009) says that the technique often works well in theory, but does not always deliver in the field.

Koole and Pot (2005) employ an ADP technique that only considers a subset of all possible states, which is called the set of representative states. They do this by running simulations of the call centre environment, that randomly pass some of the possible states in the system. The simulation scenarios are selected either at random or such that the error in linearisation of the value function is minimised. Koole and Pot (2005) report that the random selection of scenarios yields results that are hardly any better than manually constructed policies. The second method yielded improved results, but for this method no reports were made on the computational burden of the program.

2.4.5. Simulation optimisation methods

Optimisation problems where a closed expression for the objective function is not available, or where the objective function is too complex to evaluate exactly, can be solved by simulation optimisation. A simulation model is then used to obtain a noisy estimate of the expected performance of a solution, as has been described in Subsection 2.2.2. Andradóttir and Prudius (2009) mention the importance to design specialised methods that are able to obtain near-optimal solutions in the presence of noise. This section presents a number of solution methods suitable for simulation optimisation.

First, two general frameworks from Andradóttir and Prudius (2009) and Xu et al. (2010) are discussed. Both frameworks balance computational effort between explorative global search, exploitative local search, and simulation estimation. The aim of global search methods is to identify promising regions in the entire solution space. For global search methods, the probability of identifying better solutions decreases over time. Therefore, local search methods should be used to search for better solutions in the promising region (Andradóttir and Prudius, 2009).

Andradóttir and Prudius (2009) propose to integrate the three components in their Balanced Explorative and Exploitative Search with Estimation (BEESE) framework. The framework switches between global search and local search depending on what is expected to be most appropriate at the current state of the search process. The switching point is determined by tracking the improvement in the objective function value between fixed review points. When the improvement between these points is below a certain threshold, a switch is made. When the switch is performed too early, a non-optimal subregion is locally searched, whereas switching too late puts too much effort in finding promising subregions. The simulation estimation component has also been integrated in the framework: with a fixed probability an optimisation algorithm is used, otherwise additional simulations of the best solution found so far are done to increase the accuracy of the estimated objective value. No specific algorithms were proposed as global or local search methods. Instead, it is said that the framework is general enough to include most search methods available in the literature.

Xu et al. (2010) proposed a similar framework that identifies solutions that are statistically guaranteed to be close to the optimum, without having to visit all solutions. Instead of integrating the framework components, the Industrial Strength COMPASS (ISC) framework performs a global search phase, a local phase, and a clean up phase in sequence. For the global search phase, a genetic algorithm is used to identify promising regions in the solution space. When the genetic algorithm fails to improve the solution quality in a fixed number of iterations, or a fixed computational budget is exceeded, the local phase is started. For each of the feasible regions identified in the global search phase, the COMPASS algorithm is used. The COMPASS algorithm is a search method developed for simulation optimisation, guaranteeing local convergence (Hong and Nelson, 2006). The COMPASS algorithm is basically a hill climbing algorithm with an adaptive neighbourhood structure, from which solutions are randomly sampled. In the ISC framework, the neighbourhood structure is defined by the adaptive hyperbox algorithm from Xu et al. (2013). This algorithm constructs the neighbourhood based on previously visited solutions that are closest to the incumbent solution. The benefit of the adaptive hyperbox algorithm is its scalability for high-dimensional problems (i.e. problems with many decision variables). In the ISC clean-up phase, a statistical ranking and selection procedure is used to identify the best solution from the set of local optima. Ranking and selection procedures guarantee that the probability of selecting the best solution from the set exceeds a user-specified value (Gosavi, 2015).

The above frameworks are flexible to the specific optimisation algorithms that are used. Below, a number Stochastic Adaptive Search (SAS) methods from the simulation optimisation literature are presented which can be used as search algorithms in an optimisation framework. SAS techniques tend to adapt during the search process, based on experiences with previous solutions (Gosavi, 2015). They have been the dominant paradigm for designing simulation optimisation algorithms with large solution spaces (Xu et al., 2013).

Thathachar and Sastry (1987) developed a global search method to learn optimal decisions by modelling a problem as a game played by a team of cooperating learning automata, with the aim of maximising the reward in the game. Gosavi (2015) describes that the method starts like a pure random search, but starts adapting by updating the probabilities of selecting decision variable values based on the objective function values of previous solutions. Decision variable values that produce good solutions are rewarded through an increase in their selection probability, whereas bad decision variable values see their selection probabilities decrease. For the algorithm to converge quickly, accurate upper and lower solution bounds should be set. When these are too conservative, the convergence of the search process is slowed.

Similar to the learning automata method, the Model Reference Adaptive Search (MRAS) method uses a probabilistic distribution to generate candidate solutions (Hu et al., 2007). The distribution is updated based on observations in the previous iteration, aiming to bias the search towards high quality solutions. A difference between the learning automata method and the MRAS method is that MRAS learns about complete solutions in the solution space, instead of individual decision variables. Therefore, for large solution spaces, computer memory issues can arise.

Gosavi (2015) mentions the Greedy Randomised Adaptive Search Procedure (GRASP) as an SAS method that can be used for combinatorial simulation optimisation problems. Feo and Resende (1995) describe that the GRASP method consists of a greedy adaptive construction phase, followed by an improvement phase on the constructed solution. In the construction phase, solutions are iteratively generated, one element at a time. The element to be added is selected randomly from a list of elements that have the largest benefit to the objective function. After an element is added, the benefit of each element is adapted based on the solution so far. The aim of the construction phase is to create high quality initial solutions for the local search procedure (Feo and Resende, 1995). This is comparable to identifying promising regions during the global search in the above frameworks. Feo and Resende (1995) indicate that GRASP is flexible to the algorithm used in the improvement phase. An example is given where genetic algorithm mutation procedures are used to enhance the local search.

Stochastic comparison is a global random SAS method proposed by Gong et al. (2000). The method uses random search to avoid dependence on a neighbourhood structure, realising that poor neighbourhood structures can hurt algorithm performance. The aim of the algorithm is to guarantee convergence to an optimal solution while the evaluation function is noisy. Given are an initial solution X_0 and a randomly generated candidate solution in iteration k, X_k. X_0 should be replaced by X_k if the objective function value for X_k is better than X_0 for all M_k times both solutions are compared to each other. M_k increases as k increases, so that the influence of noise decreases as the iteration count increases. This is beneficial, because the convergence of the objective function value is asymptotic. Gong et al. (2000) showed that the algorithm has better convergence properties than a simulated annealing algorithm, where convergence was harmed due to noise in the objective function values.

The Nested Partitions (NP) method has been developed by Shi and Ólafsson (2000) and has been adapted for stochastic optimisation in Shi et al. (2000). The algorithm divides the solution space into a most promising region and a surrounding region. At each iteration, the promising region is partitioned (i.e. divided) into M disjoint subregions, according to a predetermined partitioning scheme. Each of the subregions and the surrounding regions are randomly sampled and the sampled solutions are evaluated by simulation. The region that on average yields the best objective function value, is selected as the most promising region for the next iteration. Eventually, there will be regions that contain only one point in the solution space, called singleton regions. The optimal solution of the NP method is the solution in the singleton region that has been visited most often. The partitioning scheme (i.e. how the partitions are defined) implicitly imposes a structure on the solution space. Therefore, the effectiveness of the NP method depends on how the partitioning is done. This can be a disadvantage for simulation optimisation problems, where little is known about the structure of the solution space (Amaran et al., 2014).

The global search algorithms presented above converge to the optimal solution, even if the objective function is evaluated by a simulation model. However, they are asymptotically convergent, meaning that the global optimum is only obtained if the number of iterations goes to infinity. In practice, the available computational effort is always limited, which is a disadvantage for these global search algorithms. The proposed frameworks from Andradóttir and Prudius (2009) and Xu et al. (2010) aim to cover this drawback by enhancing the global search with local search, with the goal to obtain high quality solutions with limited computational effort.

2.4.6. Rule-based methods

Similar to demand determination, some straightforward capacity scheduling approaches have been used in the literature. For instance, Bayliss (2016) used a full enumeration to find the best set of reserve pairings. This approach is guaranteed to find the best solution possible, since each and every combination is considered. However, for many combinatorial optimisation problems, a full enumeration quickly becomes computationally intractable (Bayliss, 2016).

Another approach taken by Bayliss (2016) is to base the start times of reserve pairings on demand that follows from repeated simulations of an airline schedule without a reserves present. The total reserve demand over all simulation runs can be represented in a graph with the departure number on the horizontal axis and the number of required reserve crew on the vertical axis. Reserve crew is then allocated at intervals with an equal area under the required reserve graph. This method was also used iteratively, where the generated reserve pairings were used for new simulations. Bayliss (2016) states that it is possible that this approach never converges to an optimal solution.

Bijvank et al. (2007) investigated the effect of the length of the reserve pairings on the quality of the sets of reserve pairings. Their strategy was to look at the longest flight in each day of the flight schedule, and copy that length to the length of the reserve pairings. The number of reserve pairings was adjusted so that the total reserve capacity remained approximately equal. Throughout the paper, Bijvank et al. (2007) make a number of assumptions that undermine the validity of this research. For example, they assume that a reserve can be called for duty for a period of 24 hours (i.e. everything is rounded to days). They also assume complete knowledge of all disruptions before the reserves pairings are designed, and days off for reserves are completely neglected.

Finally, Paelinck (2001), who developed an analytical model to evaluate a set of reserve pairings, proposed to derive optimal reserve pairings through trial-and-error. Manually constructed solutions can be evaluated using the analytical model. Similarly, Gaballa (1979) also used manually constructed sets of reserve pairings as an input.

2.4.7. Synthesis of capacity allocation methods

In this section, various methods to allocate the capacity in a personnel scheduling environment have been presented. A synopsis of these methods is given below:

- Exact solution methods;
- Heuristic solution methods;
- Dynamic programming;
- Simulation optimisation methods;
- Rule-based methods.

Exact solution methods for stochastic scheduling problems are generally used to solve (stochastic) integer or linear programming models. The models are solved by optimisation packages, which employ algorithms such as branch & bound or the simplex algorithm. When the model for a scheduling problem is non-linear, the objective function or constraint values can be approximated through relaxation of the non-linear functions. By application of linear cuts, the optimal solution is approximated in an iterative manner. The relaxation and cutting process is found to be complicated and computationally complex. Numerous heuristic procedures are applied to simplify the problem. The airline reserve crew pairing problem is non-linear because of the possibility of overlapping reserve pairings. Therefore, the suitability of using linear programming for the airline reserve crew pairing problem appears limited.

Stochastic programming models have been applied in a variety of scheduling problems with stochastic demand. These kind of problems are generally solved as two stage problems, where initial staffing levels are determined in the first step, assuming known demand. The second stage variables are often adaptions to the initial staffing levels, which reduces the number of decision variables. This is only possible when the number of available shifts is small, which does not hold for the airline reserve crew pairing problem in this research. It is expected that due to the variable reserve pairing length and the ability to include regular flight duties, the available number of patterns (shifts) is too large to effectively employ stochastic programming.

Dynamic programming can be used to solve multi-stage problems, possibly with stochastic demand. For dynamic programming methods, it is vital that the model structure is defined such that it lends itself well for a dynamic programming setup (Powell, 2009). It is especially important that the number of stages and states in the problem is limited, for the problem to remain computationally tractable. In many of the problems encountered in the literature, this is done by adjusting a certain staffing level through all stages. The possible number of adjustments usually has a smaller range than the absolute staffing levels, which reduces the number of states. For the airline reserve crew pairing problem, it is impractical to represent the large number of possible reserve pairings in a structure that is viable for a dynamic programming model.

Above, it was found that it is difficult to use exact solution methods for non-linear optimisation problems. Also, it was seen that the expected performance of exact solution methods and dynamic programming solutions suffers from increasing problem scales. The airline reserve crew pairing problem is both non-linear and has a large solution space. For such problems, heuristic solution methods can be used to derive high quality solutions for large scale non-linear problems with limited computational effort (Bianchi et al., 2009). However, a drawback of heuristic methods is that there is no guarantee that a global optimum is found during the solution process.

The application of heuristic solution methods to scheduling problems with stochastic demand is limited (Bianchi et al., 2009). For the airline reserve *cabin* crew pairing problem, a variety of heuristic methods was applied (Bayliss, 2016). In this problem, the reserve pairings were only defined by their starting time. That is, their length was fixed and no regular flight duties were placed in the reserve pairings. It is unknown how the heuristics perform for problems with more flexible reserve pairing definitions, resulting in a larger solution space. Resulting, the neighbourhood structure for local search heuristics in Bayliss (2016) does not allow full exploration of the solution space when the reserve pairings are defined more flexible.

Hybrid heuristic solution methods are increasingly being used to solve personnel scheduling problems (Van den Bergh et al., 2013). These techniques combine multiple heuristic algorithms into new solution procedures that balance exploration and exploitation of the solution space, hereby increasing the overall performance of the solution method. No hybrid heuristics have been developed for the airline reserve crew pairing problem, so it is unknown if these methods are effective in solving it.

Simulation optimisation can be used when analytical evaluation of the objective function is too complex, and a simulation model is used instead. If this is required, solution methods are available that have been shown to converge when there is noise in the objective function values. Two frameworks were developed to obtain high quality solutions in simulation optimisation problems. Comparable to the hybrid heuristic solution methods, the frameworks balance between explorative global search and exploitative local search. The global search algorithms for simulation optimisation generally converge asymptotically to the global optimum. Hence, local search algorithms should enhance the global search to obtain high quality solutions with limited computational effort.

2.5. Synthesis of literature review

This literature study consists of two parts. The first part establishes a knowledge gap in the available literature concerning airline reserve crew. The second part identifies possible problem approaches that can be employed to solve the airline reserve crew pairing problem in the novel application area.

When establishing the knowledge gap in Section 2.1, it is found that the amount of research concerned with airline reserves is limited. In particular, the design of reserve crew pairings that are assigned to regular crew who are periodically assigned reserve pairings has received little attention, even though this approach is considered industry practice. References focusing on crew disruption management generally assume that a set of reserve crew pairings has already been determined, which can be used to cover open crew positions. The current state of the art has focused on solving the airline reserve crew pairing problem for cabin crew. For this, the flexibility in how reserve pairings can be defined is limited: reserve crew pairings are fully defined through pairing start times. It is assumed that the length of the reserve pairings is constant and known in advance. Moreover, reserve pairings are assumed to be fully dedicated to reserve duties. These assumptions can be challenged: reserve pairings may vary in length and they can be combinations of reserve duties and regular flight duties.

These characteristics are especially relevant to *long-range cockpit* crew. Two reasons can be given for this. Firstly, cockpit crew are the most expensive type of crew for an airline (Bayliss, 2016). Secondly, regular long-range (i.e. intercontinental) flight pairings generally have a higher duration than short-haul pairings. To cover these long pairings, long reserve pairings are required as well. If these reserve pairings remain unused, the combination of long reserve pairings and expensive crew leads to a high waste of resources. If a (shorter) regular flight is included in the reserve pairing, that flight can be executed if the reserve remains unused. This increased flexibility in reserve pairing definition can lead to higher reserve utilisation rates and lower waste of (unused) reserve resources. However, the design of reserve pairings for long-range airline cockpit crew facing these characteristics has never been addressed before. Therefore, solving the airline reserve crew pairing problem for long-range cockpit crew is a relevant novel application.

Based on the identified gap in the literature, a research question has been defined that will be answered if a research is conducted that closes the knowledge gap:

How should airline long-haul cockpit reserve crew patterns be constructed in order to minimise the gap between scheduled reserve capacity and expected reserve demand?

The objective is to make recommendations on minimising the gap between scheduled reserve capacity and expected reserve demand of airline long-haul cockpit crew, by identification and evaluation of possible reserve crew pairing methods that utilise airline schedule data.

Sections 2.2 to 2.4 have reflected on problem approaches to solve the airline reserve crew pairing problem. Because the available literature specifically tailored towards this problem is limited, other personnel scheduling domains have been consulted, focusing on stochastic problems. From this, an overview of available problem approaches could be created, and assessed whether these approaches can be translated to suitable strategies for solving the airline reserve crew pairing problem.

The problem approaches from the personnel scheduling literature were divided into three parts that in general compose a personnel scheduling problem: scheduling objectives and evaluation, demand estimation, and capacity allocation. Sections 2.2 to 2.4 present an overview of the available methods for each of these parts, and reflect on their suitability to the airline reserve crew pairing problem. However, the discussion so far has only focused on each of the parts separately, even though a complete problem approach has to link these parts together. Below, a synthesis of the available methods is provided, such that complete problem approaches can be defined.

In Table 2.1 an overview of the methods for each of the parts of a personnel scheduling problem is given, based on the findings from Sections 2.2 to 2.4. In essence, this table is a schematic representation of the identification of the methods for each of the parts. Complete problem approaches can be defined by choosing a method from each of the columns in Table 2.1, and combining them. However, in the previous sections it was found that some methods are ill-suited to be included in problem approaches. Below, the suitability of the methods in Table 2.1 is further discussed.

Table 2.1: Overview of all methods discussed in this literature study. Complete problem approaches can be devised by combining methods from each column.

Scheduling objective	**Demand determination**	**Capacity allocation**
• Minimise resource shortages	• Directly from historical data	• Exact solution method
• Minimise combined resource shortages and unused resources	• Derive theoretical data distribution from historical data	• Heuristic solution method
• Minimise required resources while satisfying service level	• Theoretical data distribution	• Dynamic programming
• Minimise schedule costs	• Rule based method	• Simulation optimisation method
	• Assume known demand	• Rule-based method

The objective of minimising resource shortages is feasible when there is a fixed number of available resources that must be allocated in the schedule. Given that the demand is unpredictable in stochastic scheduling problems, the optimisation consists of allocating the limited available resource capacity such that the expected demand is covered to a maximum extent. This scheduling objective was used by Bayliss (2016) for the airline reserve crew pairing problem, who aimed to minimise the eventual number of expected disruptions, given that a fixed number of allocated reserve crew could be used to cover disrupted flights.

A disadvantage of the previous objective is that the resource capacity cannot be based on the expected resource demand. That is, the resource capacity cannot be sized automatically. A combined cost minimisation of unused resources and resource shortages solves this limitation. In addition, decision makers are able to shift the focus of the optimisation, either towards minimising unused resources or towards minimising resource shortages. A drawback of this objective is that the problem must be solved independently from other operations. If this is not the case, the optimised resource capacity can be inaccurate. In this literature review, no references were found that approached the airline reserve crew pairing problem with this scheduling objective.

Depending on the operational environment, the objectives posed above can be computationally intensive to evaluate. It requires an expression for stochastic demand, and computation of how well the allocated capacity covers the demand. When the objective is too complex to express in a single function, separate evaluation models can be used to calculate the value of the objective function. Both analytical and simulation evaluation models from the literature have been presented. Even though these models are able to evaluate complex objectives, a limitation is that it can be computationally intensive to do so. With respect to the reserve crew pairing problem, only analytical evaluation models have been used to derive the objective function value. The effectiveness of using simulation evaluation models is unknown.

Significantly less complicated to evaluate are the costs incurred for allocating the resources in a personnel schedule. However, minimising schedule costs, possibly while satisfying a minimum service level, is not a feasible objective for the airline reserve crew pairing problem. This is because reserve pairings are executed by regular flight crew who are periodically assigned reserve pairings, meaning that there are no direct costs incurred for scheduling reserve pairings.

Given that stochastic scheduling problems have been consulted, probability distributions are generally employed to represent demand profiles. They can be derived using historical data from practical problems, or synthesised using theoretical parameters. A drawback of using historical data from practice is that eventual solutions may only be effective for a small range of problems. In addition, the data from practice should be available and there should be sufficient data to obtain reliable parameter values. These drawbacks do not occur when synthetic data distributions are devised. However, such distributions can be non-representative for practical problems. A combination of real and synthetic data distributions can improve overall results: a range of synthetic data distributions can be used to ensure that solution methods are robust to diverse demand profiles, while data from practice can be used to validate that the used methods are also effective in solving practical problems.

The required resource demand can also be assumed, or simple rule-based approaches can be used to determine the demand levels. This can be helpful during the development stages of the research. However, when high quality solutions are to be obtained, it is expected that these demand determination methods are too simplistic.

For the capacity allocation part, it was found that exact solution methods in combination with linear programming models require complicated linear approximations when the problem characteristics are non-linear (i.e. the objective function or constraints are non-linear functions). In particular, the evaluation of the non-linear functions to determine if linear cuts should be added is computationally intensive and requires excessive solution time. Depending on the scheduling objective, the objective function of the airline reserve crew pairing problem can be non-linear. This occurs due to possibly overlapping reserve pairings, which cause interaction effects that affect the objective function value. Given these characteristics, it is expected that an exact solution process of a linear programming model for the airline reserve crew pairing problem is difficult to implement.

Stochastic programming models using exact solution methods to solve stochastic scheduling problems suffer from scalability issues. A set of demand scenarios is generally used to represent the uncertainty in the demand profile, where the optimisation objective is to find a solution that works well for the set of scenarios. The challenge that arises here is to find a a set scenarios that accurately represents the demand, but is still small enough to remain computationally tractable. Due to this, most references experienced excessive solution times to solve their stochastic programming models, and had to allow linear relaxations so that faster solution methods could be used.

Similarly, dynamic programming models should be structured such that solving the model remains computationally tractable. In particular, the number of stages and states in the problem should be limited. Whether this can be done successfully, depends on the nature of the problem. For the airline reserve crew pairing problem, it is impractical to represent the large number of possible reserve pairings in a structure that is viable for a dynamic programming model.

Given the non-linear problem characteristics, and the scalability issues of the airline reserve crew pairing problem, heuristic solution methods can be employed. Heuristic methods can derive high-quality solutions with limited computational effort, at the price of not being able to guarantee global optimality in solving the problem. Heuristic solution methods have been applied to the reserve crew pairing problem before. However, these heuristics were considered in problems where the reserve pairings were only defined by their starting time. The length of the pairings was fixed and no regular flights duties could be included in the reserve pairings. Heuristic solution methods have not been used in solving airline reserve crew pairing problems where reserve pairings can be defined more flexible. Therefore, their performance in solving these kind of problems is unknown.

In addition, hybrid heuristic methods were identified as techniques that combine multiple heuristic algorithms into new solution procedures. The aim of these methods is to effectively balance exploration and exploitation of the solution space, hereby increasing solution quality. Such hybrid heuristics have been successful in solving scheduling problems in other domains, but have never been applied to the airline reserve crew pairing problem. Therefore, the effectiveness of these methods in solving this problem is still unknown.

Finally, specific simulation optimisation methods can be used to generate personnel schedules if the objective function must be evaluated by a simulation model due to its complexity. This introduces noise in the objective function value, which influences convergence properties of solution methods. Despite the positive convergence behaviour under stochastic evaluation of these methods, they are generally asymptotically convergent to the global optimum. This means that infinitely many iterations are required to obtain the optimum, which is a large drawback for these solution methods. To improve the effectiveness of simulation optimisation, frameworks have been proposed that combine explorative global search and exploitative local search techniques. Simulation optimisation solution methods have never been considered for the airline reserve crew pairing problem.

All in all, feasible objectives were found to be minimisation of resource shortages or a combined minimisation or resource shortages and unused resources. The second objective has never been applied to the airline reserve crew pairing problem before. With respect to demand determination, it is feasible to express the stochastic demand using data distributions from historical demand data or from synthetic parameters. A combination of both methods can be used to neutralise the drawbacks observed in the individual methods. For the capacity allocation, the use of hybrid heuristic methods has been successful in solving personnel scheduling problems, but these methods have never been applied to the airline reserve crew pairing problem before. Therefore their effectiveness towards solving this problem is still unknown. The same holds for simulation optimisation methods. If the reserve crew pairing problem requires a simulation model to evaluate solutions, simulation optimisation methods are another original method to solve the airline reserve crew pairing problem.

3
Research design

The remaining chapters of this thesis aim to answer the research question that has been defined. This chapter discusses the high level research design of the project. First, the scope of the project is explained in Section 3.1. In Section 3.2 a research framework is presented that has been made with the goal of solving the airline reserve crew pairing problem, and thus reaching the research objective. Finally, the assumptions that were made in developing the models and performing the research are listed in Section 3.3.

3.1. Project scope

This section defines the context of the airline reserve crew pairing problem with respect to the general airline crew scheduling process. In Figure 3.1, the airline crew scheduling process is shown. The input for crew scheduling is an airline flight schedule, indicating which flights have to be flown at what times, and the corresponding aircraft types that the flights should be flown with. Using this information, crew pairings are constructed that consist of sequences of duties and layovers, beginning and ending at a crew base. The crew pairing problem consists of creating a set of pairings that covers all flights in the flight schedule while complying with crew regulations such as minimum and maximum layover rest and pairing lengths. For long-haul flights, the crew pairing problem is relatively straightforward, since pairings in general exist of single return flight from and to an airline hub, satisfying the minimum rest requirements.

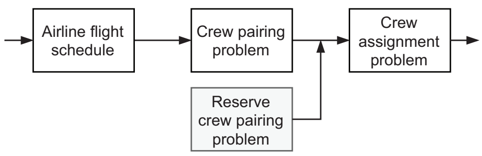

Figure 3.1: Position of reserve crew pairing problem in crew scheduling process.

In Figure 3.1, it can be seen that the airline reserve crew pairing problem is a separate problem that is solved in parallel to the regular crew pairing problem. In the airline reserve crew pairing problem, reserve patterns are constructed, which are defined as sets of reserve pairings. Reserve pairings are pairings that include at least one day of reserve duty. Differentiation is made between pure reserve pairings and mixed reserve pairings. In pure reserve pairings, the full reserve pairing consists of reserve days.

Pure reserve pairing

A crew pairing of which each day is a reserve duty day.

In mixed reserve pairings, a regular flight pairing follows after a number of reserve days.

Mixed reserve pairing
A crew pairing in which a regular flight follows after a number of reserve duty days.

In Figure 3.2, examples of a pure and a mixed reserve pairing are given. In this figure, RES indicates a reserve duty and FLT represents a day of a flight pairing. The advantage of mixed reserve pairings is that long flight pairings can be covered with a relatively low number of reserve days that have to be invested. This gives the airline the opportunity to cover a wide range of flight pairings, of varying length and start time, while keeping the number of invested reserve days low. A disadvantage of using mixed reserve pairings is that secondary disruptions are initiated by using a mixed reserve pairing, because the initial flight assigned to the reserve crew cannot be executed by that crew anymore. These secondary disruptions have to be covered by other reserve crew, possibly causing tertiary disruptions, or more. This snowball effect of disruptions can be stopped by using pure reserves, whose usage does not cause a secondary disruption.

	MON	TUE	WED	THU	FRI	SAT	SUN
Pure	RES	RES	RES	RES	RES	RES	
Mixed	RES	RES	FLT	FLT	FLT	FLT	

Figure 3.2: Pure reserve pairing (top) and mixed reserve pairing (bottom). Usage of a mixed reserve pairing causes secondary disruptions, because the original flight in the mixed reserve pairing cannot be executed by its original crew anymore.

The output from the regular crew pairing problem and the reserve crew pairing problem is a set of regular flight pairings and reserve pairings that have to be assigned to crew. This is done in the crew assignment problem, which has the goal to cover all pairings by a crew member, while satisfying minimum rest requirements between pairings. That is, the output of the reserve crew pairing problem serves as the input of the crew assignment problem. When mixed reserve pairings are assigned, a regular flight pairing has to be included as part of the reserve pairing. The flight that is assigned to a mixed reserve pairing is not determined in advance and can vary per instance of the reserve pairing.

Figure 3.3 illustrates a generic crew scheduling process with respect to a time horizon until the flights are operated. On a strategic level, the size of the workforce has to be determined based on the airline network and operations, where distinctions are made per posting (i.e. aircraft type) and operational rank. Part of this problem are the transitions that crew can make between postings and ranks. Given the workforce size for a specific crew type, a planning margin is required based on illness and operational disturbances. During this process, a reserve budget is determined, which represents the amount of days that should be included in the reserve patterns.

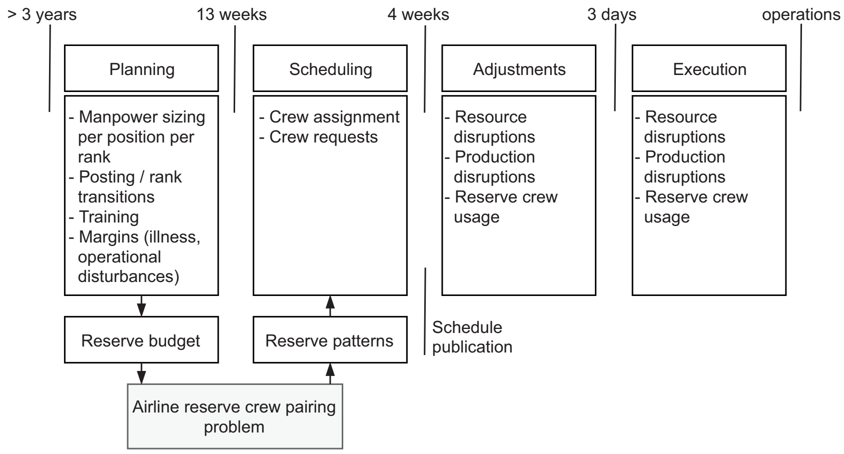

Figure 3.3: Position of reserve crew pairing problem in airline crew scheduling process.

With the reserve budget known, reserve patterns are constructed based on the flight schedule per posting and rank. The reserve patterns are taken into account in the scheduling process, where crew are assigned flight pairings, reserve duties, and other activities such as training or vacation. In other words, the crew assignment problem is solved in this phase. During this phase, crew are allowed to request specific flight pairings and leave periods. Four weeks in advance, the crew schedules are fixed and published. From this point on, disruptions that are inflicted on the schedule possibly require the usage of reserve crew to cover the disruption. In Figure 3.3, it can be seen that the airline reserve crew pairing problem is situated between the planning and scheduling process. This agrees with Figure 3.1, where the resulting set of reserve pairings is used as an input for the crew assignment problem. That is, the airline reserve crew pairing problem requires input from a strategic planning and delivers output for the assignment process.

The position of the airline reserve crew pairing problem within the generic crew scheduling process allows to set the model specifications. Given that the problem is positioned between planning and scheduling, the maximum run time of the model is not a critical factor, as opposed to for example recovery problems, where ad-hoc solutions have to be generated in small time periods. However, in order to execute required experiments in a practical amount of time, it is decided that the model should return reserve patterns for realistically sized problems in the order of a couple of hours or less.

Furthermore, it should be feasible to complete the project within the duration of a master thesis, which is a period of nine months. For this, the scope of the project should be limited. Therefore, it has been decided to isolate the airline reserve crew pairing problem from the overall crew scheduling problem. This implies that various assumptions are made to be able to obtain a workable project scope. The assumptions are detailed further in Section 3.3.

Finally, it is preferred that the reserve crew pairing model is developed so that it can be implemented in a more elaborate crew scheduling model in the future. For this, the inputs and outputs of the reserve crew pairing model should be generic, allowing additional models to be developed that can accurately determine or utilise the input and output of the reserve pairing model, respectively.

3.2. Design framework

The framework that serves as the basis of the research design is shown in Figure 3.4. The approach that has been chosen to achieve the research objective is to develop various optimisation methods that can be used to generate reserve patterns. Through a comparison of these methods the best method to create reserve patterns can be identified. Using this knowledge, recommendations can be made on how the gap between scheduled reserve capacity and expected reserve demand can be minimised.

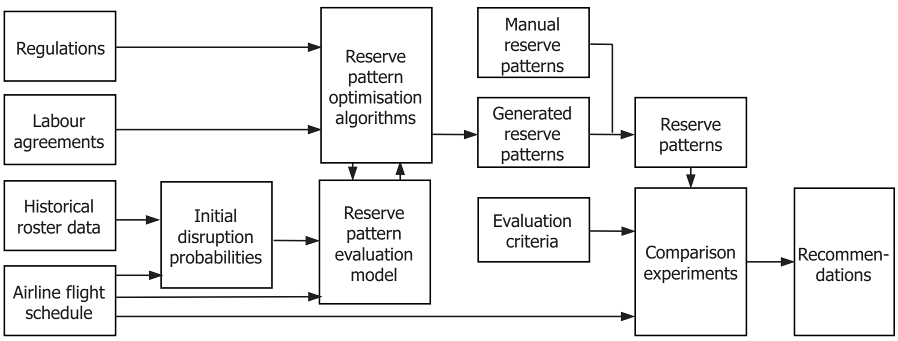

Figure 3.4: Design framework for the airline reserve crew pairing problem.

The core of the framework consists of the reserve pattern evaluation model and the reserve pattern optimisation algorithms. The function of the evaluation model is to determine specific performance measures of existing reserve patterns, which indicate the quality of the reserve pattern. The reserve patterns that are evaluated can either be manually created by a human scheduler or be computer generated. The evaluation model is used in combination with four optimisation algorithms. The optimisation algorithms iteratively generate reserve patterns and evaluate them using the evaluation model, aiming to return the best reserve patterns in terms of the performance measures after the optimisation process has finished. The four optimisation algorithms are introduced here:

1. **Random search**
   The random search algorithm generates reserve patterns as combinations of randomly selected reserve pairings. This algorithm has previously been applied to the airline reserve crew pairing

problem by Bayliss (2016), but it is the first time that it is applied to the airline reserve crew pairing problem for long-haul cockpit crew. This algorithm is important for the research project for two reasons:

(a) Random search is straightforward to implement, hence it can be used to demonstrate a proof of concept of solving the airline reserve crew pairing problem;

(b) The other three optimisation algorithms use more advanced optimisation procedures, but all incorporate some element of randomised search. Therefore, the random search algorithm serves as a benchmark method that demonstrates the added effect of the other optimisation methods over a pure random search.

2. **Learning Automata Search Technique (LAST)**
The LAST algorithm is a random search technique first introduced by Thathachar and Sastry (1987), that uses adaptive search procedures to improve the performance of the random search algorithm. The motivation to use LAST in this research project is that learning based optimisation techniques have never been applied to the airline reserve crew pairing problem before. Therefore, their performance is still unknown. From a methodological viewpoint, this is a novel element in the scientific domain.

   A description of the general LAST method found in Gosavi (2015) is used as the basis for this algorithm. The method starts like a pure random search, but starts adapting by updating the probabilities of selecting decision variable values based on the objective function values of previous solutions. Decision variable values that produce good solutions are rewarded through an increase in their selection probability, whereas bad decision variable values see their selection probabilities decrease. In this research project, the general method has been enhanced to deal with the airline reserve crew pairing problem more effectively, which is a novel element of this research. The way that the algorithm is made specific for the airline reserve crew pairing problem is that learning rate weights are introduced that increase or decrease the learning rate for specific reserve pairings, based on their performance in previously generated solutions.

3. **Greedy Randomised Adaptive Search Procedure (GRASP)**
The third optimisation algorithm is a construction based algorithm called GRASP, which is based on a general GRASP algorithm as devised by Feo and Resende (1995). The general GRASP algorithm starts from a zero solution, and adds one decision variable per iteration, depending on which decision variable improves the objective function the most during that iteration. In this research, the algorithm is adapted specifically for the airline reserve crew pairing problem, which is a novel point of this research.

   The GRASP algorithm is particularly interesting to the airline reserve crew pairing problem because the use of mixed reserve pairings influences which decision variable (i.e. reserve pairing) is optimally used in the next iteration. Similar to the LAST algorithm, the GRASP algorithm is newly enhanced, specifically for the problem considered in this research. To gain computational performance, the contribution of each reserve pairing when added to the incumbent solution is predicted (the pairing potential). The pairing potential is based on the number of flights that a reserve pairing covers and the disruption probabilities of those flights.

4. **Greedy Randomised Adaptive Search Procedure - Longest Flights (GRASP-LF)**
The GRASP-LF method is newly devised for this research project and is a variation to the regular GRASP method. It is based on rules taken from the current industry practice, where the longest flights on each day in the schedule are covered first. It is currently unknown if this approach yields high quality reserve patterns. Therefore it is included in this research.

The evaluation model and optimisation algorithms are considered in more detail in Chapters 4 and 5, respectively. The input for the optimisation process is an airline flight schedule which is used to measure the quality of reserve patterns against. That is, a reserve pattern should always be optimised for a corresponding flight schedule, meaning that the reserve pattern is only optimised for that specific flight schedule. In other words, the flight schedule serves as the input problem for the airline reserve crew pairing problem. Flight information of individual pairings should be included in the flight schedule. Part of this information are the lengths of the pairings, which are of influence to reserve crew pairing for long-haul crew. The pairings in long-haul flight schedules generally span multiple days and vary significantly in length from flight to flight. Given that a reserve pattern should be optimised for a flight schedule, the reserve pattern should take this high variability in flight pairings into account. This means that the algorithms should be able to create reserve patterns where the reserve pairings vary in length as well. Compared to the state of the art in the literature as currently set by Bayliss (2016), this is the main novelty from an application viewpoint.

The input scenario for the comparison experiments is based on the KLM flight schedule for the Airbus A330 posting for the first officer rank, and can be found in Appendix A. The flight schedule consists of 78 flights ranging in length from three to eight days, as shown in Table 3.1. It can be seen that the majority of pairings is either three or four days is length. The seven and eight day pairings are interesting from an experimental point of view, because covering these flights requires long reserve pairings, which are inefficient for the majority of the flights, given that they cause wasted reserve days when they are used for short flights. The primary disruption probabilities are derived from historical roster data, and are based on first officer absence rates. The planned and maximum flight duty periods and the reserve buffer period were available for use. The premium weight, which expresses the expected difficulty towards finding crew at premium cost for a flight, has been based on the disruption probabilities per flight, where flights with high disruption probabilities have high premium weights.

Table 3.1: Pairing length per day in the problem scenario flight schedule.

	Weekday							
Pairing length	**MON**	**TUE**	**WED**	**THU**	**FRI**	**SAT**	**SUN**	*Total*
3	6	4	6	5	3	4	6	*34*
4	5	6	4	7	7	5	4	*38*
5	-	-	-	-	-	1	1	*2*
6	-	-	-	-	-	-	-	-
7	-	-	-	-	1	-	1	*2*
8	-	1	-	-	-	-	1	*2*
Total	*10*	*11*	*10*	*12*	*11*	*10*	*13*	*78*

The manually constructed reserve pattern corresponding to the flight schedule is used to compare the reserve patterns from the optimisation models with. This pattern can be found in Appendix A as well. The reserve pattern consists of thirteen reserve pairings having a total reserve budget of 45 days, with at least one pairing starting per day. All reserve pairings except for four are mixed reserve pairings, aiming to cover almost all flights. The only flight that is not covered is flight 72. Even though the aim in constructing this manual reserve pattern was to cover all flights, it has been chosen to leave this flight uncovered because it is known as a popular flight with a high request rate, indicating that it is easy to find premium flying crew for this flight.

Part of the problem input are disruption probabilities per flight. These are derived from one year of historical cockpit crew roster data of KLM. This data has been analysed to determine the reserve crew demand for each flight flown in that year per operational rank (i.e. captain, first officer, second officer). Any differentiation between causes of reserve crew demand has not been made, so it does not matter if a reserve is required due to crew illness or an operational disturbance. Fact is that a reserve is required and when creating reserve patterns this should be accommodated.

The reserve demand data per flight has been aggregated to obtain crew absence (i.e. disruption) probabilities per destination, month, and day of the week. Exploiting this reserve demand data introduces opportunities compared to the current way of working in practice, where reserve patterns are manually constructed twice per year per rank and division, based on only the start time and length of the pairings in the flight schedule. Using automated and optimised reserve crew pairing, new reserve patterns can be made on a weekly or monthly basis where the disruption probabilities of the flights are not only different per destination, but also adjusted per week or month. This allows the user to include seasonality effects in the design of reserve crew patterns. Hence, instead of creating reserve patterns that are created towards covering an average week in a half year period, it will be possible to create short term reserve patterns for much smaller time periods, increasing the margin between the current solutions in practice and the potential of optimised reserve crew pairing.

The regulations that hold for the reserve crew pairing problem primarily follow from European rules concerning flight and duty time limitations, as defined in the publication by the Council of European Union (2014). This protects airline crew from having insufficient rest in between duties and pairings. The labour agreements that have been implemented in the model are derived from KLM practices, and are defined as follows:

- The maximum length of reserve pairings per posting per rank per day cannot exceed the maximum flight length on that day.
- There are two types of reserve duty, called reserve and standby. When crew are on reserve duty, they are available for a predefined time period (usually 12 hours) and can be called in for a flight duty between those hours. Standby crew can only be called in for duty between 9h and 10h on the day of operations or between 20h and 22h on the day before operations.
- When a reserve pairing is not utilised for three consecutive days, it cannot be utilised from the fourth day onward.

It is possible to define these labour agreements flexibly in the model, or to exclude them altogether. This makes that the model is generic for a wider variety of airlines.

The goal of this research is to make recommendations on how reserve patterns should be generated for long-haul cockpit crew. To make these recommendations, the reserve pattern optimisation algorithms will be used to generate reserve patterns for the KLM A330 first officer case study. The performance and the results of the model will be analysed and compared to each other and to manually constructed reserve patterns that are currently in use. Based on the performance of the models, recommendations are then made about how reserve patterns should be constructed.

3.3. Assumptions

This section lists and explains the main assumptions that have been made during the development of the reserve crew pairing models. These assumptions have to be made to isolate the problem from the overall scheduling process, which is required to define a project that is solvable in feasible time.

- Only long-haul intercontinental crew is considered.
- Flight pairings consist of single return flights from an airline hub. For long-haul crew, this is often the case in practice, and allows to represent a flight schedule as a list of departures from an airline hub.
- The airline uses a single base for its operations, implying that crew cannot be swapped between bases, and that all disruptions originate and are resolved at the single crew base.
- Reserve crew usage and premium flying are the only recovery measure from crew related disruptions. This allows to isolate the problem from the overall crew scheduling process. For long-haul crew, this assumption is acceptable to make, because the potential of other recovery measures such as resource swapping or flight delaying is limited.
- When a reserve pairing is utilised for recovery, it cannot be used for subsequent disruptions anymore.
- When premium flying is required for recovery, the entire flight has to be flown against a premium. A fixed number of required premium days per flight is known in advance.
- Disruptions are generated and resolved per flight, in the order of departure. It is assumed that all disruptions occur before departure, meaning that disruptions do not occur at outstations.
- Reserve holding is not allowed, meaning that a reserve has to be used if it that reserve pairing is feasible to cover a flight. In practice, this can occur.
- The reserve crew pairing problem is solved separately per posting and rank. This means that flying below rank (e.g. a captain acting as first officer) is not possible. Also, crew qualifications (e.g. a qualification to serve a specific destination) are neglected.
- Reserve crew absence and recovery of crew (after illness) is excluded from the model.

4

Pattern evaluation

This chapter details how airline reserve patterns are evaluated, with the aim to determine the quality of the reserve pattern. For this, the objective function of this project is considered first, in Section 4.1. Afterwards, the required input to evaluate reserve patterns is described in Section 4.2. Then, two models are explained that are considered to evaluate reserve patterns with. Section 4.3 details a simulation model, whereas Section 4.4 details an analytical model. The model output is presented in Section 4.5 and the model verification is presented in Section 4.6.

4.1. Objective function

Reserve patterns have to be evaluated to determine the quality of the patterns. The quality of a solution should represent how well it achieves the objectives of the decision maker. An objective function expresses these objectives in mathematical terms, so they can be measured in optimisation studies. The objective of this project is to minimise the gap between allocated reserve capacity and expected reserve demand. With this in mind, a number of objective functions have been considered, which are discussed below. The first alternative objective function is shown in Equation 4.1.

$$\min \quad \textit{unused reserve days} + w_p \cdot \textit{premium days} \tag{4.1}$$

In this function, w_p is the relative weight given to the number of premium days, with respect to the number of unused reserve days. This allows the decision maker to prioritise between both factors in the objective function. In practical terms, this objective function aims to minimise inefficiencies from airline operations: unused reserve days are days during which crew did not execute any flights and crew flying at premium cost is more expensive than regular flying. It is expected that an increase in one factor leads to a decrease in the other: increasing the reserve budget decreases the expected number of premium days, but is likely to increase the number of unused reserve days. The optimal objective function value is expected at the point where the unused reserve days and premium days are appropriately balanced according to their weights, as illustrated in Figure 4.1.

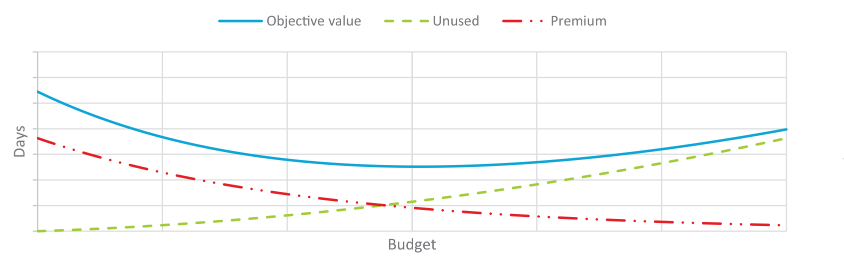

Figure 4.1: Expected shape (qualitative) of the objective function as a combination of unused reserve days and premium days.

The main disadvantage of this objective function is that it does not say anything about the costs that are required to operate all flights in the flight schedule. Unused reserve days are wasted resources compared to used reserve days, but an airline has equal expenses for both unused and used reserve days. In other words, this objective function has limited practical value. Another disadvantage is that this objective function can cause the model to be prejudiced towards using mixed reserve pairings. These pairings cause additional secondary disruptions which impose a high reserve usage rate, which would result in a lower number of unused reserve days and thus a lower objective value. However, causing additional disruptions for the sake of using more reserves is not desirable.

To address the downsides of the objective function in Equation 4.1, an alternative objective function is presented in Equation 4.2.

$$\min \quad \textit{reserve budget} + \textit{premium days} \tag{4.2}$$

The reserve budget in Equation 4.2 is expressed as the number of reserve duty days that are used to create the reserve pairings. When are reserve pairing is not used, the number of reserve days in the pairing are lost days. Following this, this objective function aims to minimise the number of days that are required to ensure that all disruptions are resolved, be it using reserve crew or using crew flying at premium cost. For this objective function it is also expected that an increase in reserve budget decreases the number of premium days, as shown in Figure 4.2. The objective functions in Equations 4.1 and 4.2 both allow to automatically determine a required reserve budget. However, in both objective functions, it is expected that resulting reserve budgets corresponding with optimal objective values will be too low. This can be explained by considering the practical meaning of reserve days and premium days. A reserve day should be considered as an insurance against propagating disruptive effects: it has a *chance* of being used, meaning that value is only added by reserve days a fraction of the time. On the other hand, premium days are (1) only required when a flight is disrupted, which is depending on chance, and (2) always generate value to the airline once required. Therefore, it is expected that the model will favour solutions with a relatively high number of premium days, because they are more efficient in resolving disruptions.

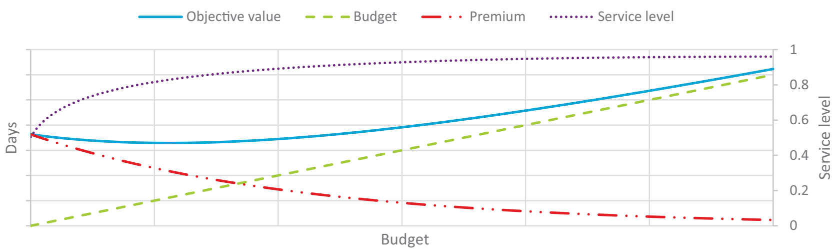

Figure 4.2: Expected shape (qualitative) of the objective function as a combination of reserve budget and premium days.

A risk in automatic sizing of the reserve budget, is that the model does not take into account a number of social effects that exist in practice. There is a natural aversion against flying against premium cost, due to reasons shown below:

- Crew dislike changes to their schedules;
- Finding crew for premium flying is uncertain and its success depends on time, destination, and crew absence within the division and rank;
- Finding crew for premium flying becomes harder when it occurs more often, also implying that premium costs depend on the operational 'social state';
- Premium flying imposes additional disruptive effects to crew schedules

Accurately modelling these effects is considered out of scope for this project. Therefore, two alternative objective functions are shown in Equations 4.3 and 4.4.

$$\begin{aligned} \min \quad & \textit{reserve budget} + \textit{premium days} \\ \text{subject to} \quad & \textit{budget} \approx \textit{budget requirement} \end{aligned} \tag{4.3}$$

$$\begin{aligned} \min \quad & \textit{reserve budget} + \textit{premium days} \\ \text{subject to} \quad & \textit{service level} \geq \textit{service level requirement} \end{aligned} \tag{4.4}$$

Both objective functions allow the decision maker to reflect the operational social state in a constraint that imposes a minimum value for the reserve budget in the optimal solution. In Equation 4.3, this is done directly via a budget constraint, which enforces a predetermined number of reserve days in the reserve pattern. A disadvantage of Equation 4.3 is that the social state of the system can not intuitively be summarised in a single reserve budget value. To counter this, the objective function in Equation 4.4 allows the decision maker to impose a minimum service level on the airline operations. The service level is expressed as a percentage of time during which the number of flights that is flown at premium costs is below a certain value. For example, in 95% of the flight schedule repetitions (e.g weeks) there can be no more than three flights flown at premium costs. It is expected that the service level increases as the reserve budget increases, as depicted in Figure 4.2. This is a much more intuitive method for the decision maker to express budget requirements for the reserve pattern. Summarising, the objective function in Equation 4.4 measures the number of days that are required to operate the flight schedule, given that crew disruptions occur and need to be covered by reserve crew. On top of this, social effects that are apparent in practice are intuitively represented in the form of a service level, which expresses the fraction of time during which a minimum number of flights is flown without using premium flying. Therefore, this objective function is chosen to optimise the reserve patterns with.

No closed expression for the number of premium days has been defined, because these values can only be determined by subjecting a reserve pattern to the execution of an airline flight schedule. Due to the interaction between the flight schedule and the reserve pattern, and between individual pairings in the reserve pattern, the expressions would be complex non-linear functions.

Instead, separate evaluation models have been investigated to obtain values for the number of premium days. In Section 4.3, a simulation evaluation model is detailed, whereas Section 4.4 presents an analytical evaluation model. But first, the model input is described in Section 4.2

4.2. Model input

The goal of the evaluation models is to determine how good a reserve pattern is when it is used in combination with an airline flight schedule. Therefore, the airline flight schedule and the proposed reserve pattern are the required inputs for the evaluation models.

Airline flight schedule

The airline flight schedule is a list of departures from the airline hub for a single aircraft type and crew rank. The airline schedule could span an arbitrary number of days, but in practice schedules often repeat on a weekly basis. It is assumed that the airline flight schedule is a standard schedule that repeats indefinitely. That is, after the last day in the flight schedule, the first day is operated again. For each flight, the following characteristics should be specified:

- **Flight ID** - The departure number of the flight.
- **Destination** - The destination of the flight.
- **Crew reporting** - The day and start time of the flight duty period.
- **Disruption probability** - The probability that a reserve crew is required, excluding secondary disruptions (i.e. disruption following from using mixed reserves).
- **Route length** - The number of flight days in the flight pairing.
- **Rest length** - The number of rest days in the flight pairing.
- **Planned flight duty period** - The planned duration that the crew has to be on duty to execute the flight.
- **Maximum flight duty period** - The maximum duration that the crew is allowed to be on duty to execute the flight. This number depends on the departure time, the number of cockpit crew, and the on-board rest facilities.
- **Reserve buffer period** - Extension of the maximum flight duty period for a reserve crew that is assigned to this flight. The number depends on the number of cockpit crew.

- **Premium weight** - A multiplier for the amount of premium days that should be counted when no reserve crew is available for the flight. This allows the decision maker to express particular ease or difficulty in finding premium crew for each flight.

Reserve pattern
The reserve pattern consists of a list of reserve pairings that are used to cover open crew positions on the flights in the flight schedule. The reserve pattern should have the same time span as the airline flight schedule. That is, reserve pairings can start on the same days as the days in the flight schedule. Similar to the flight schedule, it is assumed that the reserve pattern is repeated indefinitely. For each reserve pairing. the following characteristics should be specified:

- **Reserve ID** - The identification number of the reserve pairing.
- **Crew reporting** - The day and start time of the reserve duty period for the first or first two days.
- **Pairing length** - The total number of days in the reserve pairing, including flight days in mixed reserves.
- **Mixed flight length** - The number of days of the flight pairing that is part of a mixed reserve pairing. If no flight is part of the reserve pairing, this value equals 0
- **Mixed flights** - The flight IDs that can be assigned as flight in a mixed reserve pairing.
- **Assign probabilities** - The probability that a flight is assigned as the flight in a mixed reserve pairing for each flight of the mixed flights. These probabilities should sum to 1.

Besides the flight schedule and the reserve pattern, the service level requirements should be determined by the model user. This entails setting a maximum number of flights that can be flown against premium costs, and the fraction of time during which this number can be exceeded.

4.3. Simulation model

A simulation model has been developed to evaluate reserve patterns. This model estimates the objective function numerically, instead of deriving it exactly. The concept of the model is that of a basic disruption recovery model, where the operations of an airline flight schedule are simulated. Flights can get disrupted and recovered. All disruptions follow from crew absence and the only recovery measures are the use of reserve crew or premium payed crew overtime. The flight schedule is repeated N times (i.e. repetitions), and each departure d in the flight schedule is considered in each repetition. A flowchart of the simulation evaluation model is shown in Figure 4.3. The steps in the flowchart are further detailed below, where the numbers in the header correspond to their respective blocks in the flowchart.

Assign flights (EV.2)
When mixed reserve pairings are used to cover open positions, the flights originally assigned to the reserve pairing cannot be executed by the reserve crew anymore. In other words, using a mixed reserve to cover an open position, will always cause another open position to a later flight in the airline schedule. To include these effects, a unique flight from the flight schedule first has to be assigned to each mixed reserve pairing in the reserve pattern before the disruption and recovery process can be simulated.

First, all feasible flights from the flight schedule that can be assigned to each mixed reserve pairing in the reserve pattern are determined. There are two requirements for this: (1) the length of the flight in the flight schedule should be equal to the number of flight days in the reserve pairing; (2) the last day of the flight in the flight schedule should be equal to the last flight day of the mixed reserve pairing.

Figure 4.4 shows one mixed reserve pairing and a number of flight pairings. For each of the flight pairings, it is indicated whether it can be assigned as the flight in the mixed reserve pairing. The BAH and DCA flights fulfil the two requirements posed above. The CPT and LOS flights are both too long and the SFO flight is too short to be assigned as the mixed flight. In theory, the first requirement can be relaxed, so that shorter flights can also be assigned to mixed reserve pairings, given that the mixed flight ends on the same day as the reserve pairing. The ACC flight in Figure 4.4 is an example of such

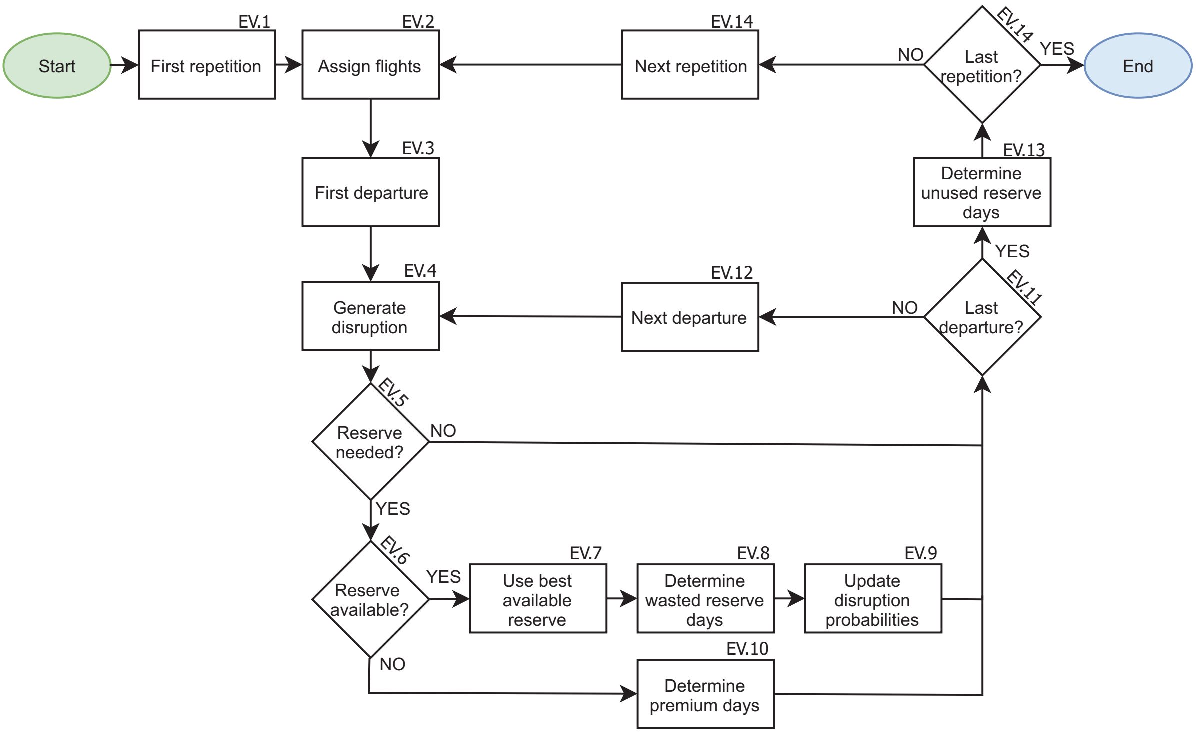

Figure 4.3: Flowchart of the simulation evaluation model.

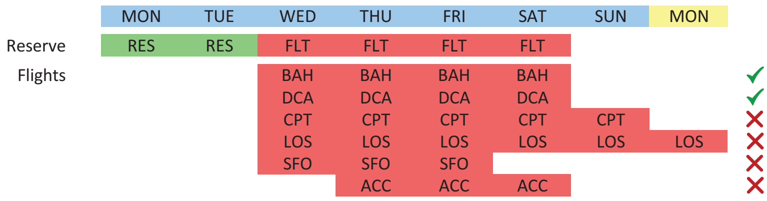

Figure 4.4: Example showing which flight pairings can be assigned as the flight in a mixed reserve pairing.

a flight. However, this is unwanted in practice, because it leads to an extra unused reserve day when the reserve pairing is not used.

Second, the model determines the probability that each of the feasible flights is assigned to a mixed reserve pairing. Two possible strategies (i.e. *assign policies*) have been implemented in the evaluation model: *equal probability* and *lowest disruption first*. In the equal probability policy, each feasible flight is assigned with equal probability. For example, if there are four feasible flights, each flight is assigned with a probability of 0.25. This policy is representative of current industry practice, where flights are randomly assigned to mixed reserve pairings. In the lowest disruption first policy, the flight with the lowest disruption probability that has not yet been assigned to a mixed reserve pairing, is assigned with a probability of 1. In Chapter 7, it is explained which policy is used for the experiments.

The feasible flights and assignment probabilities are determined for each mixed reserve pairing before the first repetition of the disruption simulator has started. However, the flights assigned to mixed reserve pairings can vary per repetition of the flight schedule. Therefore, for each repetition of the flight schedule (i.e. N times), a new flight should be selected based on the assignment probabilities of the reserve pairing. Because the feasible flights and assignment probabilities have already been determined, the flight realisation can be done fast for each repetition, by weighted random choices of the feasible flights. The assigned flights are tracked to avoid that the same flight is assigned multiple times in the same repetition.

Generate disruption (EV.4)
After flights are assigned to the mixed reserve pairings in the reserve pattern, each departure from the airline hub is considered in order of crew reporting time. Since the evaluation model is a disruption

and recovery simulator, disruptions should be generated for flights in the flight schedule. This is done using the disruption probability of the flight and a random number generator. When the random number is lower than or equal to the disruption probability of the flight, the flight is disrupted, meaning that a reserve crew is required to cover a vacant crew position.

Recovery (EV.6 / EV.7 / EV.8 / EV.10)
The recovery process is defined as checking for feasible and available reserve pairings and utilising them to cover an open crew position. To improve the computational performance of the recovery process, the feasible reserve pairings and the preferred reserve usage order for each flight in the flight schedule are determined before the disruption simulations are started. Figure 4.5 presents for a variety of reserve pairings whether they can be used to cover the BAH flight pairing. The following requirements should be satisfied for a reserve pairing to be feasible to cover an open crew position on a flight:

1. The reserve pairing days should cover the flight pairing days: the first day of the flight should be larger than or equal to the first day of the reserve pairing and the last day of the flight should be smaller than or equal to the last day of the reserve pairing. The second and last reserve pairings in Figure 4.5 violate this requirement.
2. The flight must depart within three days of the reserve pairing start day. The first reserve pairing in Figure 4.5 violates this requirement.
3. For mixed reserve pairings, the flight must depart before the start day of the flight assigned to the reserve pairing. The fourth reserve pairing in Figure 4.5 violates this requirement.
4. The flight must depart within twelve hours of the reserve pairing reporting time. The fifth and sixth reserve pairings in Figure 4.5 violate this requirement.
5. The maximum flight duty period for the reserve crew cannot be exceeded. This requirement is checked through Equation 4.5.

$$\textit{Flight reporting time} - \textit{reserve reporting time} - \textit{reserve buffer} \leq \textit{maximum FDP} - \textit{planned FDP} \quad (4.5)$$

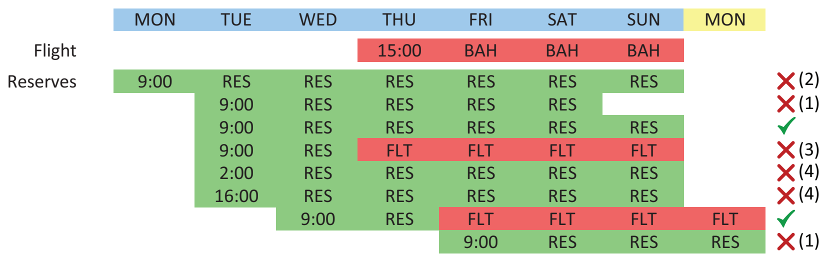

Figure 4.5: Example showing how feasible reserves to cover an open crew position on a flight are determined, including which requirements are violated.

For each flight, the order in which the feasible reserves are preferably used for recovery (i.e. *reserve usage policy*) has to be determined. Two reserve usage policies have been defined: *earliest start time* and *minimum waste days*. In the earliest start time policy, the feasible reserve pairings are sorted by ascending reporting day and time. That is, the reserve pairing with the earliest starting time is used for recovery first. The minimum waste days policy aims to minimise loss days in utilising reserves for recovery. This is done by sorting the feasible reserve pairings by increasing number of waste days, where waste days are defined as the number of days between the last day in the reserve pairing and the last day of the flight. The chosen reserve usage policy for the experiments is determined in the sensitivity analysis in Chapter 7.

The recovery process during the disruption simulations becomes straightforward when the feasible reserve pairings and preferred reserve usage order have already been determined. When a reserve crew is required for a flight, the feasible reserves are checked in the reserve usage order. When a feasible reserve has not yet been used to cover an open crew position of an earlier flight, it will be used to solve the current disruption. When the reserve has already been used, the next feasible reserve in the reserve usage order is considered, until the disruption is solved or all feasible reserves have been considered. It is not possible to refrain from using a reserve to save it for later disruptions.

When a reserve pairing is used, it will be labelled 'used' and the number of wasted reserve days (i.e. the inefficiency of using the reserve) are determined. This number is calculated with Equation 4.6. The wasted reserve days are incremented to the total number of unused reserve days over all simulations.

$$\textit{Wasted reserve days} = \textit{flight start day} - \textit{reserve start day} + \textit{reserve end day} - \textit{flight end day} \quad (4.6)$$

When no reserve pairing is available anymore, premium pay needs to be offered to other crew. It is assumed that the number of premium pay per flight is known in advance, and is specified as part of the flight schedule input data. This number is added to the total number of premium days over all simulations.

Update disruption probabilities (EV. 9)
If a mixed reserve pairing is used to cover an open crew position, the flight that was originally assigned to this pairing gets disrupted as well. This secondary disruptive effect is included by updating the disruption probability of the originally assigned flight to 1.0, meaning that it is certain that the flight is disrupted. Since the airline schedule is considered in order of departure, flights originally assigned to mixed reserves will always be considered later than all flights that can be covered by the mixed reserves. Because these mixed flights are considered after their disruption probability has been updated, the updated disruption probability will be used when the mixed reserve flight is considered.

Unused reserve days (EV. 13)
Next to wasted reserve days following from suboptimal utilisation of reserve pairings to cover open crew positions (i.e. the flight does not fit perfectly in the reserve pairing), unused reserve days can also follow from reserve pairings that have not been used entirely at the end of a repetition of the airline flight schedule. Therefore, at the end of each repetition, the number of unused reserve days following from all unused reserve pairings (i.e. reserve pairings without the 'used' label) has to be added to the incumbent number of wasted reserve days. For a single unused pairing, the number of unused reserve days is computed using Equation 4.7. It can be seen that for mixed reserve pairings, only the reserve days before the mixed flight begins are taken into account.

$$\textit{Unused reserve days} = \textit{reserve pairing length} - \textit{mixed flight length} \quad (4.7)$$

Simulation continuity (EV. 14)
Above, the steps in the simulation evaluation model flowchart that have to be taken each repetition are described. The simulation will obtain accurate performance measurements for a reserve pattern by performing these steps for a high number of repetitions. For the airline reserve crew pairing problem, these repetitions cannot be seen independently from each other: what happens in one repetition, can influence the results of previous and future repetitions. The reason for this is that the flight pairings in the airline schedule and the reserve pairings in the reserve pattern span multiple days. Possibly, these pairings can continue into future repetitions. For example, if a five day flight departs on Sunday (where the airline schedule is repeated every week), the flight does not finish until Thursday of the next repetition. Therefore, a reserve pairing that has started in a previous repetition, is able to cover an open position on a flight departing in the current repetition, as shown in Figure 4.6. In this figure, it is assumed that the flight schedule runs from Monday to Sunday. The reserve pairing starting on Saturday is able to cover the flight starting on Monday.

Similarly, a mixed reserve pairing that is used to cover an open position in the current repetition, can cause a secondary disruption in a later repetition. An example of this situation is shown in Figure 4.7. Again, it is assumed that the flight schedule runs from Monday to Sunday. Using the reserve pairing starting on Saturday, causes a secondary disruption on Monday in the next repetition.

To include these effects, the evaluation model does not simulate N independent repetitions, but simulates N repetitions *in sequence*. When determining which flights can be assigned to mixed reserve

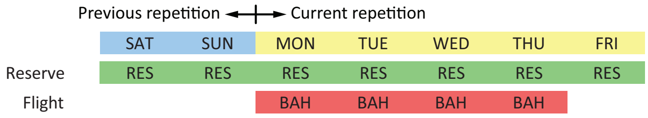

Figure 4.6: Reserve pairings that started in previous repetitions are able to cover flights in the current repetition.

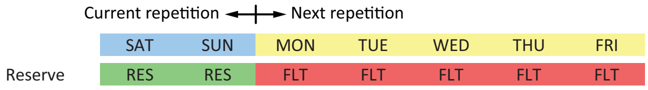

Figure 4.7: Mixed reserve pairings can cause secondary disruptions in future repetitions.

pairings and which reserve pairings are feasible for each flight, the simulation continuity is taken into account. Feasible flights to assign to mixed reserve pairings are not only distinct by their flight ID, but also by the number of airline schedule repetitions into the future. Similarly, the feasible reserves to cover an open crew position are defined both through the reserve ID and the number of flight schedule repetitions into the past.

Before the disruption and recovery simulations are started, the maximum required number of repetitions that must be looked ahead or back is derived from the airline schedule and the reserve pattern. During the simulations, a set of this number of repetitions is tracked simultaneously. This allows the model to use reserves from previous repetitions and to cause secondary disruptions to flights in future repetitions. When a repetition is finished, the oldest repetition in the set at that point is removed and a new repetition is appended to the set. For the new repetition, all the reserve pairings are unused and no secondary disruptions exist yet.

Because the evaluation model is able to simulate multiple repetitions in sequence, the model can handle airline schedules of arbitrary length. In practice, airline schedules commonly are repeated weekly, but in this model reserve patterns for one day long airline schedules can be evaluated as well.

To illustrate the simulation continuity, consider the example in Figure 4.8. The standard airline schedule consists of only one day of operations, with one departure to BAH. The corresponding reserve pattern consists of one six day reserve pairing. In the non-continuous situation, the only feasible reserve for the BAH flight is the reserve pairing starting on the same day as the departure. However, in the continuous situation, the same flight can be covered by three instances of the reserve pairing: from the current repetition and the previous two repetitions. Therefore, the model has to simultaneously keep track of three repetitions. For instance, if the third flight to BAH is disrupted, and the *earliest start time* reserve usage policy is used, the reserve pairing from two repetitions before would be considered first. If it is still available, then the reserve from this repetition will be labelled ‘used’.

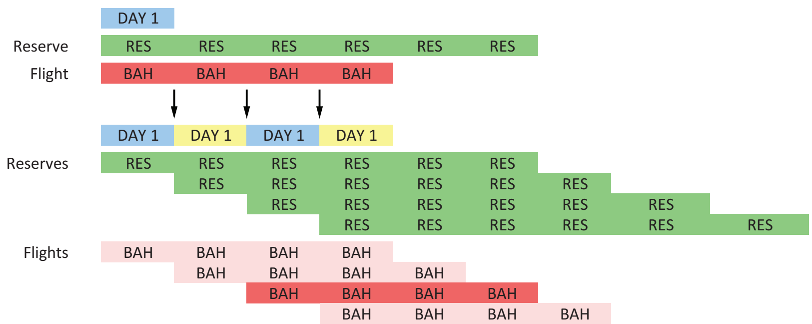

Figure 4.8: Translation of a standard airline schedule and reserve pattern to a continuous situation as required during simulation.

Finally, when starting the disruption and recovery simulations, the first few repetitions do not have information about used and available reserve pairings from previous repetitions, because these repetitions have not been simulated yet. To initiate the simulation process, for the first few repetitions it is assumed that reserve pairings from previous iterations are all unused. This can lead to inaccurate performance measurements. Therefore, the first N_c repetitions should not be included in calculating the performance measures, so that the simulation can converge to a steady state of used and unused reserves in previous repetitions. Preliminary experiments show that a fixed value of twenty repetitions is sufficient to converge towards a steady state.

4.4. Analytical model

The advantage of the simulation model is that it is a straightforward method to estimate the quality of a reserve pattern, even though the operational relations are complex. However, there are two notable drawbacks to using simulation evaluation: (1) the simulation model is stochastic, meaning that there is uncertainty in the model outcome, and (2) a large number of repetitions are required to mitigate the stochasticity of the model, which can be computationally expensive. Therefore, an analytical model is presented in this section with the aim to obtain exact expressions with reduced computational effort. The method is based on previous work by Paelinck (2001) and Bayliss (2016). The main adaptions with respect to the model by Bayliss (2016) are the inclusion of mixed reserve pairings and pairings of variable length. Below, the working principle of the analytical method for this project is presented.

Table 4.1 shows the notation that is used for the analytical model. A reserve combination can be considered to be a state of the reserve pattern during operations, indicating which reserves are still available and which cannot be used to cover disrupted flights anymore. For example, when there are only two reserve pairings in the reserve pattern, there are four reserve combinations:

1. Both pairings available;
2. Only the first pairing available;
3. Only the second pairing available;
4. No pairings available.

Table 4.1: Notation for analytical evaluation model.

Symbol	Meaning
p_f	Initial disruption probability of flight f
$p_{\text{eff},f}$	Disruption probability of flight f after reserve usage
r_i	Reserve usage probability for reserve i
a_c^f	Availability probability of reserve combination c before flight f
$a_{c,i}^f$	Availability probability of reserve combination c without reserve i before flight f
b_f	Cumulative availability probability of reserves for flight f

In the beginning, a_0^1, the probability of all reserves still being available, equals 1.0. Similarly, $r_i = 0$ for all $i \in I$ and $b_f = 0$ for all $f \in F$. The analytical model considers each flight sequentially, ordered by ascending departure day and time. For the first reserve in the reserve order of this flight, Equations 4.8 to 4.11 should be evaluated for each reserve combination in which the first reserve is still available.

$$\Delta a_c^f = -a_c^{f-1} p_f \tag{4.8}$$

$$\Delta a_{c,i}^f = +a_c^{f-1} p_f \tag{4.9}$$

$$r_i = r_i + a_c^{f-1} p_f \tag{4.10}$$

$$b_f = b_f + a_c^{f-1} \tag{4.11}$$

For the next reserves in the reserve order, the same equations should be evaluated, but only for the reserve combinations in which that reserve is still available and that have not been considered yet when earlier reserves in the reserve order were considered for this flight. When all reserves for a certain flight are considered, the effective disruption probability is determined via Equation 4.12. By propagating through all flights in the flight schedule, the number of premium days can be determined from the effective disruption probabilities.

$$p_{\text{eff},f} = p_f(1 - b_f) \tag{4.12}$$

There are two drawbacks to the analytical model. The first is that the effective disruption probabilities of flights depend on which flights are assigned as flights in mixed reserve pairings. This is due to the secondary disruptions that are caused when mixed reserve pairings are used. In this case, the primary disruption probability is 1.0 for reserve combinations in which the mixed reserve pairing is used. Because the pattern quality is dependent on which flights are assigned to the mixed reserve pairings, a separate evaluation should be done for each combination of assigned flights to mixed reserve pairings. This results to an exponential increase in required computational effort.

The second drawback is a consequence of the reserve usage order that is defined differently for each flight. Due to this, all reserve combinations have to be tracked. The number of reserve combinations increases exponentially with the number of reserves. The continuity effects that were described in Section 4.3 also have to be taken into account, increasing the number of reserve combinations even further.

These two effects make that for realistically sized problems, the analytical model is in fact slower than the simulation model is for a reasonable number of repetitions. Therefore, it is decided to continue with the simulation model as the method to determine the quality of reserve patterns.

4.5. Model output

The pattern evaluation model computes the performance of a reserve pattern when it is used to recover from disruptions given that an airline flight schedule is executed.

Objective function

The outputs related to the objective function that are computed by the evaluation model as shown in Table 4.2.

Table 4.2: Objective function outputs of the reserve pattern evaluation model.

Output
Objective function value
Reserve budget
Premium days
Unused reserve days
- From unused pairings
- From inefficient reserve usage

The overarching performance indicator is the objective function value, which follows from evaluating Equation 4.4. It is a single number that expresses the overall quality of the solution, where lower values indicate better solutions. To evaluate Equation 4.4, the number of premium days should be known. This number follows directly from the evaluation model, as explained in Section 4.3. By keeping track of these numbers separately, solutions can be assessed based on both elements of the objective function. This is important because of the inverse relationship between both elements in the objective function: an increase in reserve budget is likely to cause a decrease in premium days. Therefore, solutions can have similar objective function values, with different numbers of reserve budget and premium days. For the number of unused reserve days, a further differentiation is made between days following from unused pairings and days following from inefficient utilisation of reserves (i.e. when a flight is shorter than a reserve pairing).

Reserve pairing information

On top of general objective function information, the simulation evaluation model allows to extract detailed performance metrics of individual reserve pairings in the reserve pattern. For each reserve pairing, the information shown in Table 4.3 is tracked.

The reserve usage per flight gives insight in which reserve pairings are used to cover disruptions per flight in the airline schedule. The sum of the reserve usage per flight gives the total reserve usage

Table 4.3: Reserve pairing outputs of the reserve pattern evaluation model.

Output
Reserve usage probability per flight
Total reserve usage probability
Premium cost prevented
Unused reserve cost caused
Effective reserve pairing benefit

probability per reserve pairing. Naturally, high reserve usage probabilities are preferred for high quality reserve patterns, because the waste of resources is the lowest for these pairings. For each reserve pairing, the objective function decrease following from prevented premium days and the objective function increase following from caused unused reserve days are tracked as well. Subtracting these two numbers gives an indication of the effective benefit of a reserve pairing. The effect of individual pairings cannot be seen separate from the overall reserve pattern, because mixed reserve pairings can cause secondary disruptions that need to be covered elsewhere in the reserve pattern.

Flight information

Similar to the reserve pairings, detailed information can be derived from the evaluation model for each individual flight in the airline schedule. The outputs that are tracked are shown in Table 4.4.

Table 4.4: Flight outputs of the reserve pattern evaluation model.

Output
Total disruption probability
Secondary disruption probability
Effective disruption probability
Reserve IDs covering flight

Most outputs are concerned with tracking disruptions during the simulation process. Firstly, the total disruption probability is computed, which consists of primary and secondary disruptions (i.e. following from mixed reserve usage). Note that this number can be higher than the initial disruption probability value from the flight schedule input. The amount of secondary disruptions is tracked separately as well, to assess the effect of using mixed reserve pairings. The amount of unresolved disruptions is expressed through the effective disruption probability, which is the disruption probability of a flight given that reserve pairings can be used to cover open positions. Finally, the reserve pairings that can be used to cover open positions on the flight are extracted as well.

4.6. Verification

In this section, a system test for the evaluation model is done in which a small airline flight schedule will be disrupted and recovered using a reserve pattern. The reserve pattern will be evaluated both manually (using the analytical model) and numerically (using the simulation model). Through this, it will be shown that the results of both evaluations support each other, indicating a proper functioning of the simulation evaluation model as a whole.

The reserve pattern and flight schedule that are evaluated are shown in Figure 4.9. The first reserve is able to cover the flights to HAV, DAR, and KGL. Since the first reserve is a mixed reserve, it cannot be used to cover any flights after the mixed reserve has started. Hence, the MCT and LOS flights can be assigned as mixed flights to the first reserve, but cannot be covered by this reserve if they are disrupted. The second reserve pairing is able to cover all but the HAV flight.

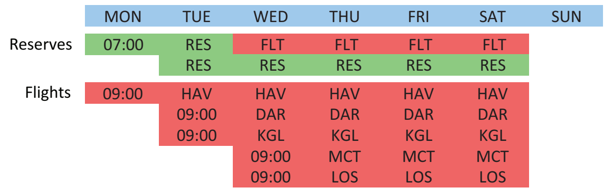

Figure 4.9: Reserve pattern and flight schedule for the evaluation model system verification.

The disruption probabilities, premium weights and the reserve usage order for each flight are shown in Table 4.5. For this derivation, arbitrary values have been chosen, but the order of magnitude corresponds to values encountered in industry practice.

Table 4.5: Flight schedule information for evaluation model system verification.

ID	**Destination**	**Disruption probability**	**Premium weight**	**Reserve order**
1	HAV	0.08	1.0	1
2	DAR	0.10	1.1	1, 2
3	KGL	0.12	1.3	1, 2
4	MCT	0.14	1.4	2
5	LOS	0.16	1.5	2

Below, the analytical evaluation of the reserve pattern is presented for the corresponding flight schedule. The evaluation considers each flight in the order of departure. For each flight, all reserves in the reserve order will be considered to reduce the effective disruption probability. Due to the reserve order that differs for each flight, one must keep track of all possible combinations of reserve pairings being still available and having already been used. This will be done using a vector representation, where an A indicates a reserve being available and an N indicates a reserve not being available:

$$\begin{array}{cccc} \text{AA} & \text{NA} & \text{AN} & \text{NN} \\ \end{array}$$
$$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Before the first flight is considered, both reserves are still available with a probability of 1.0: AA $= 1.0$. The symbol a_c^f will be used to represent the probability that reserve combination c is still available before the start of flight f. For example, $a_{\text{AA}}^1 = 1.0$ and $a_{\text{NA}}^1 = a_{\text{AN}}^1 = a_{\text{NN}}^1 = 0.0$. The probability that reserve i is used will be expressed by r_i. The initial and effective (i.e. after reserve consideration) disruption probabilities of flight f are defined as p_f and $p_{\text{eff},f}$, respectively. For the HAV flight, knowing that $a_{\text{AA}}^1 = 1.0$ and reserve 1 is the first reserve in the reserve order, the situation changes as follows:

$$\begin{array}{lllllll} \Delta a_{\text{AA}}^2 & = & -a_{\text{AA}}^1 p_1 & = & -1.0 \cdot 0.08 & = & -0.08 \\ \Delta a_{\text{NA}}^2 & = & +a_{\text{AA}}^1 p_1 & = & +1.0 \cdot 0.08 & = & +0.08 \\ r_1 & = & r_1 + a_{\text{AA}}^1 p_1 & = & 0.0 + 1.0 \cdot 0.08 & = & 0.08 \end{array}$$

Reserve 1 covers the HAV flight without causing any wasted reserve days, because the lengths of both pairings are equal. Therefore, the wasted reserve days do not need to be updated. Since the probability that no reserves are available equals 0.0, $p_{\text{eff},1} = 0.0$. Taking into account the changing reserve availability combinations, the reserve combination availability probabilities for flight 2 are shown below:

$$\begin{array}{cccc} \text{AA} & \text{NA} & \text{AN} & \text{NN} \\ \end{array}$$
$$\begin{bmatrix} 0.92 & 0.08 & 0.0 & 0.0 \end{bmatrix}$$

Both reserve 1 and reserve 2 can be used to cover disruptions to flight 2. For the situation where $a_{\text{AA}}^2 = 0.92$, reserve 1 will be used, since this reserve is higher in the reserve order for the DAR flight. Using reserve 1, the parameters change as shown below:

$$
\begin{array}{lclcl}
\Delta a^3_{\text{AA}} & = & -a^2_{\text{AA}} p_2 & = & -0.92 \cdot 0.10 & = & -0.092 \\
\Delta a^3_{\text{NA}} & = & +a^2_{\text{AA}} p_2 & = & +0.92 \cdot 0.10 & = & +0.092 \\
r_1 & = & r_1 + a^2_{\text{AA}} p_2 & = & 0.08 + 0.92 \cdot 0.10 & = & 0.172
\end{array}
$$

In 8% of the time, reserve 1 is not available anymore to cover flight 2. In that case, reserve 2 has to be used. If $a^2_{\text{NA}} = 0.08$, the following changes take place:

$$
\begin{array}{lclcl}
\Delta a^3_{\text{NA}} & = & -a^2_{\text{NA}} p_2 & = & -0.08 \cdot 0.10 & = & -0.008 \\
\Delta a^3_{\text{NN}} & = & +a^2_{\text{NA}} p_2 & = & +0.08 \cdot 0.10 & = & +0.008 \\
r_2 & = & r_2 + a^2_{\text{NA}} p_2 & = & 0.0 + 0.08 \cdot 0.10 & = & 0.008
\end{array}
$$

Reserve 2 covers the DAR flight without any wasted reserve days, but for reserve 1, one wasted reserve day has to be counted when it is used to cover the DAR flight, which happens in 9.2% of the time. Again, the probability that no reserves are available equals 0.0, so $p_{\text{eff},2} = 0.0$. When all deltas are summed and added to the reserve availability combinations, the situation at the start of flight 3 is as shown below:

$$
\begin{array}{cccc}
\text{AA} & \text{NA} & \text{AN} & \text{NN} \\
\end{array}
$$
$$
\begin{bmatrix} 0.828 & 0.164 & 0.0 & 0.008 \end{bmatrix}
$$

For the KGL flight, again reserve 1 and reserve 2 can be used, where reserve 1 is the preferred reserve to use. Given that $a^3_{\text{AA}} = 0.828$, the parameters get updated as shown below:

$$
\begin{array}{lclcl}
\Delta a^4_{\text{AA}} & = & -a^3_{\text{AA}} p_3 & = & -0.828 \cdot 0.12 & = & -0.0994 \\
\Delta a^4_{\text{NA}} & = & +a^3_{\text{AA}} p_3 & = & +0.828 \cdot 0.12 & = & +0.0994 \\
r_1 & = & r_1 + a^3_{\text{AA}} p_3 & = & 0.172 + 0.828 \cdot 0.12 & = & 0.271
\end{array}
$$

Similarly, when $a^3_{\text{NA}} = 0.164$, using reserve 2 causes the following adaptions:

$$
\begin{array}{lclcl}
\Delta a^4_{\text{NA}} & = & -a^3_{\text{NA}} p_3 & = & -0.164 \cdot 0.12 & = & -0.0197 \\
\Delta a^4_{\text{NN}} & = & +a^3_{\text{NA}} p_3 & = & +0.164 \cdot 0.12 & = & +0.0197 \\
r_2 & = & r_2 + a^3_{\text{NA}} p_3 & = & 0.008 + 0.164 \cdot 0.12 & = & 0.0277
\end{array}
$$

Similar to flight 3, in 9.9% of the time, one reserve day is wasted when reserve 1 is used to cover the KGL flight. In 0.8% of the time, both reserves have already been used before the start of the KGL flight. Therefore $p_{\text{eff},3} = 0.008 \cdot p_3 = 0.008 \cdot 0.12 = 0.00096$. At the start of flight 5, the reserve availability combinations are as follows:

$$
\begin{array}{cccc}
\text{AA} & \text{NA} & \text{AN} & \text{NN} \\
\end{array}
$$
$$
\begin{bmatrix} 0.729 & 0.244 & 0.0 & 0.0277 \end{bmatrix}
$$

When considering the MCT and LOS flights, it should be taken into account that the situation propagates differently depending on which flight is assigned to mixed reserve 1. The model assumes that each of the flights is assigned half of the time. First, assume that MCT is assigned to the first reserve as the mixed flight. Then, for the MCT flight, given that $a^4_{\text{AA}} = 0.729$ and only reserve 2 can be used:

$$
\begin{array}{lclcl}
\Delta a^5_{\text{AA}} & = & -a^4_{\text{AA}} p_4 & = & -0.729 \cdot 0.14 & = & -0.102 \\
\Delta a^5_{\text{AN}} & = & +a^4_{\text{AA}} p_4 & = & +0.729 \cdot 0.14 & = & +0.102 \\
r_2 & = & r_2 + a^4_{\text{AA}} p_4 & = & 0.0277 + 0.729 \cdot 0.14 & = & 0.130
\end{array}
$$

In 24.4% of the time, reserve 2 is still available while reserve 1 was used to cover one of the earlier flights. For this proportion, the disruption probability p_4 changes to 1.0, because using reserve 1 causes a secondary disruption to the mixed MCT flight. The probability parameters then change as shown below:

$$
\begin{array}{lclcl}
\Delta a^5_{\text{NA}} & = & -a^4_{\text{NA}} p_4 & = & -0.244 \cdot 1.0 & = & -0.244 \\
\Delta a^5_{\text{NN}} & = & +a^4_{\text{NA}} p_4 & = & +0.244 \cdot 1.0 & = & +0.244 \\
r_2 & = & r_2 + a^4_{\text{NA}} p_4 & = & 0.130 + 0.244 \cdot 1.0 & = & 0.373
\end{array}
$$

In all other cases, no reserves are available to cover the flight 4. This occurs in 2.77% of the time. Since reserve 1 has already been used in this situation, $p_4 = 1.0$ in calculating the effective disruption probability: $p_{\text{eff},4} = 0.0277 \cdot p_4 = 0.0277 \cdot 1.0 = 0.0277$. For the final flight, the starting situation is as follows (still assuming that the mixed flight of reserve 1 is flight 4):

$$\begin{array}{cccc} \text{AA} & \text{NA} & \text{AN} & \text{NN} \end{array}$$
$$\begin{bmatrix} 0.627 & 0.0 & 0.102 & 0.271 \end{bmatrix}$$

In this case, the only reserve combination that can still be used to cover a disrupted flight is AA, because only reserve 2 can be used to cover this flight, and $NA = 0.0$. Knowing that $a^5_{\text{AA}} = 0.627$, the situation changes as follows:

$$\begin{array}{lclcl} \Delta a^{\text{end}}_{\text{AA}} & = & -a^5_{\text{AA}} p_5 & = & -0.627 \cdot 0.16 & = & -0.100 \\ \Delta a^{\text{end}}_{\text{AN}} & = & +a^5_{\text{AA}} p_5 & = & +0.627 \cdot 0.16 & = & +0.100 \\ r_2 & = & r_2 + a^5_{\text{AA}} p_5 & = & 0.373 + 0.627 \cdot 0.16 & = & 0.474 \end{array}$$

In all other cases, or in 37.3% of the time, no reserve is available to cover the LOS flight. This leads to $p_{\text{eff},5} = 0.373 \cdot p_5 = 0.373 \cdot 0.16 = 0.0597$.

Using the same equations and logic, the same derivation can be done assuming that flight 5 is assigned as the mixed flight of reserve pairing 1. For this situation, the resulting parameters of interest are $r_2 = 0.474$, $p_{\text{eff},4} = 0.00388$, and $p_{\text{eff},5} = 0.0781$. Given that flights 4 and 5 are each assigned as mixed flights half the time, the resulting effective disruption probabilities for these flights become:

$$\begin{array}{lclcl} p_{\text{eff},4} & = & 0.5 \cdot 0.0277 + 0.5 \cdot 0.00388 & = & 0.0158 \\ p_{\text{eff},5} & = & 0.5 \cdot 0.0597 + 0.5 \cdot 0.0781 & = & 0.0689 \end{array}$$

The same problem has also been solved using the simulation evaluation model, using one million repetitions to estimate the parameter values. In Table 4.6, several parameter values for this problem are compared between the analytical evaluation and the numerical evaluation. It can be seen that the results between both methods are in support of each other, with maximum differences no larger than 0.5%. From this it can be concluded that the evaluation model functions correctly for the purpose of reserve pattern evaluation.

Table 4.6: Comparison of parameter values between analytical evaluation and numerical evaluation.

Parameter	**Analytical value**	**Numerical value**	**% difference**
$p_{\text{eff},1}$	0.0	0.0	0.0
$p_{\text{eff},2}$	0.0	0.0	0.0
$p_{\text{eff},3}$	0.00096	0.000946	0.001
$p_{\text{eff},4}$	0.015778	0.015681	0.010
$p_{\text{eff},5}$	0.068928	0.069285	0.036
r_1	0.27136	0.27131	0.005
r_2	0.47363	0.47458	0.095
Wasted reserve days	0.63731	0.63874	0.143
Unused reserve days	4.08912	4.08446	0.466
Premium days	0.50816	0.50967	0.151

5

Pattern optimisation

This chapter details the optimisation algorithms that have been developed for the airline reserve crew pairing problem. In total, four algorithms are considered, but the reserve pattern representation in these algorithms is first discussed in Section 5.1. The first two algorithms are based on random search procedures. The first algorithm is a pure random search and is explained in Section 5.2. The second algorithm is called the Learning Automata Search Procedure (LAST), and is detailed in Section 5.3. The final two algorithms are construction based algorithms, called Greedy Randomised Adaptive Search Procedure (GRASP) and GRASP-Longest Flights (GRASP-LF). These methods are considered in Sections 5.4 and 5.5, respectively.

5.1. Pattern representation

In this section, the method of defining reserve patterns is explained. This method consists of two parts: unique pairing generation and mathematical reserve pattern representation.

Unique pairings

Before the optimisation process is started, first all possible unique reserve pairings that can be used in a reserve pattern are generated. For this, standard pairing lengths, standard mixed flight lengths, and standard duty start times and durations have to be specified. Then, for each day, all feasible combinations of above variables are enumerated and appended to a list of reserve pairings. Infeasible pairings are pairings that either:

- are longer than the longest flight starting on the same day as the first day of the reserve pairing;
- do not have any feasible flights that can be assigned as flights, in the case of mixed reserve pairings;
- do not cover any flights.

For each reserve pairing, the same characteristics as in Chapter 4.2 should be specified. Reserve pairings that start with at least two reserve days after each other, can have different duty start times and durations per day. The duty times are only specified for the first two days, because during latter days reserves can only be utilised the day before. In case there is only one reserve day before a mixed reserve flight starts, there can only be one duty start time and duration.

Mathematical pattern representation

With all feasible reserve pairings known and available in a list, reserve patterns are mathematically easily denoted as a vector of integer numbers. Each position in the vector corresponds to the unique reserve pairing on the same location in the vector of feasible reserve pairings. A reserve pattern is defined as a vector of integer numbers, where the value of the integer indicates the number of times the related reserve pairing is used in the reserve pattern.

5.2. Random search

The first optimisation algorithm is a random search procedure, where reserve patterns are generated by randomly selecting a subset of reserve pairings from the set of unique reserve pairings. The steps that are taken in this optimisation method, are shown in the flowchart in Figure 5.1. Below, separate parts of the flowchart are discussed in more detail. The numbers in the headers correspond to the blocks in the flowchart.

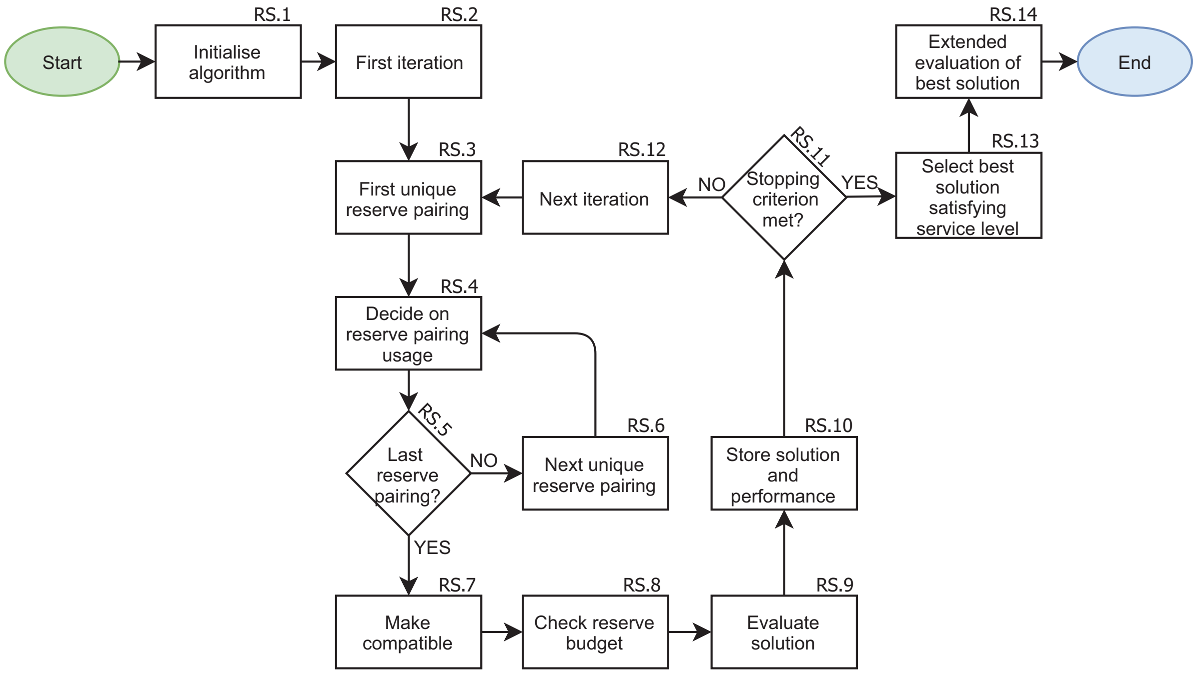

Figure 5.1: Flowchart of the random search optimisation method.

Initialisation (RS.1)
At the start of the optimisation process, the initial pattern density has to be set. This variable determines the selection probability of each pairing in the list of unique pairings, an thus determines how many reserve pairings are on average in the reserve pattern. For the random search algorithm, this value is constant and equal for each unique pairing. Its value is determined using a sensitivity analysis.

Pairing selection (RS.4 / RS.5 / RS.6)
With the initial pattern density known, individual reserve pairings can be selected based on this number. For each reserve pairing in the list of unique feasible pairings, its usage is decided as follows. A random number is generated using a random number generator. When the random number is lower than or equal to the pattern density (i.e. selection probability), then the pairing is included in the reserve pattern.

Pattern compatibility (RS. 7)
When reserve pairings are selected at random, it is possible that the resulting reserve pattern is not compatible with the flight schedule. This occurs when then are more mixed reserve pairings with mixed flights with a certain length and start day, than there are flights in the flight schedule with that length starting on that day. For example, assume there are two mixed reserves starting on day one with a length of four days, and a flight starting on day two, lasting three days. If there is only one three day flight starting on day two of the flight schedule, then one of the two mixed reserve pairings cannot be assigned a flight when the simulation evaluation model is run.

To avoid these incompatibilities, reserve pairings that are incompatible with the flight schedule are removed randomly until the schedule is compatible. For this, the lengths of the departing flights are counted and tracked for each day in the flight schedule. For each mixed reserve pairing that is included in the reserve pattern, it is checked if there is at least one flight starting on the same day and with the same length as the mixed flight. If a flight is available, the number of flights of that length on that day in the tracking variable is decreased by one. If no flight is available, the reserve pairing is removed from the reserve pattern. Following this method, all reserve patterns are always compatible with the flight schedule before they are evaluated.

Budget adaption (RS.8)
The service level constraint, introduced in Chapter 4, is used to improve the performance of the random search method. When a solution is generated that satisfies the service level, the reserve budget of this solution will be used as an upper limit for future solutions. A corresponding lower limit is enforced that is always five days lower than the upper limit, so that each generated solution lies within a range of budget values that have proven to generate solutions that satisfy the service level. The budget limits decrease the size of the solution space, and ensure that no time is wasted by searching for solutions with ineffective budget values. That is, when a pattern has been found that satisfies the service level, future solutions should try to satisfy the service level with a lower reserve budget.

The model also has the functionality to directly impose a budget constraint on reserve patterns. This means that the total number of reserve days in the reserve pattern that the solution should have, is specified in advance. The budget constraint works both for solutions that should be lower than or equal to a fixed budget value, or for solutions that should be equal to a budget value. Generated solutions that do not comply with the budget, are termed infeasible.

Even though the random search method uses the pattern density to ensure that the number of selected reserve pairings is accurate, most randomly generated reserve patterns do not satisfy the budget constraint. To avoid that a lot of time is wasted by evaluating infeasible solutions, infeasible reserve patterns are 'repaired' until the budget constraint is satisfied. If the reserve pattern has more reserve days than the budget allows, reserve pairings are randomly dropped one-by-one from the reserve pattern. On the other hand, when the reserve pattern is short on reserve days, reserve pairings that are not yet in the reserve pattern are added randomly one-by-one. When pairings are considered to be added to the reserve pattern, it is ensured that the reserve pattern budget does not exceed the constraint value afterwards. Otherwise, pairings have to be removed from the pattern again. After each pairing that is removed or added, the budget is recalculated, and a decision to remove or add additional pairings is made. Also, after each pairing that is added to the reserve pattern, the pattern compatibility has to be ensured again, as explained above. Using this method, it is ensured that each generated reserve pattern is always compatible with the flight schedule and always satisfies any budget constraints. Afterwards, the generated solution is evaluated by the simulation evaluation model.

Iterations (RS.10 / RS.11 / RS.12 / RS.13 / RS.14)
After the solution has been evaluated by the evaluation model, the integer reserve pattern and the corresponding objective function value are stored by the model. The previous steps that have been described, detail only one iteration of the random search algorithm. However, this process should be repeated for a large number of iterations for the algorithm to be effective. Over the course of the optimisation process, a large number of random reserve patterns are evaluated and stored. A stopping criterion is imposed on the algorithm by limiting the number of iterations that can be done. After the final iteration, the solution with the lowest objective function value in the memory, is selected as the best solution.

During the random search process, the computational effort invested in evaluating each randomly generated solution is limited, to obtain an appropriate balance between computational performance and solution accuracy. However, for the final solution, a more accurate quality estimate is obtained. This is done by evaluating the optimal solution with the evaluation model for 25000 consecutive weeks. This number is sufficient to obtain reliable performance measures for the generated reserve pattern.

5.3. LAST

The LAST algorithm is a random search technique first introduced by Thathachar and Sastry (1987), that uses adaptive search procedures to improve the solution quality of the random search algorithm. Gosavi (2015) describes that the method starts like a pure random search, but starts adapting by updating the probabilities of selecting decision variable values based on the objective function values of previous solutions. Decision variable values that produce good solutions are rewarded through an increase in their selection probability, whereas bad decision variable values see their selection probabilities decrease. Also when decision variables are not included in the solution, their selection probability changes, using the scheme in Figure 5.2. For the algorithm to converge quickly, accurate upper and lower solution bounds should be set. When these are too conservative, the convergence of the search process is slowed.

	Pairing used	Pairing not used
Good solution	+	-
Bad solution	-	+

Figure 5.2: Scheme indicating how selection probabilities are updated for different generated solutions, where a + indicates an increasing selection probability and a - indicates a decreasing selection probability.

The LAST optimisation procedure is comparable to the random search algorithm, and its corresponding flowchart is shown in Figure 5.3. In this figure, it can be seen that the only extra steps are the blocks LA.11 to LA.13: determining the reserve pairing benefit, updating the selection probabilities, and updating the best normalised objective values. On top of that, the algorithm initialisation (block LA.1) requires extra work. These adaptions from the random search algorithm to obtain the LAST algorithm are detailed in the following paragraphs.

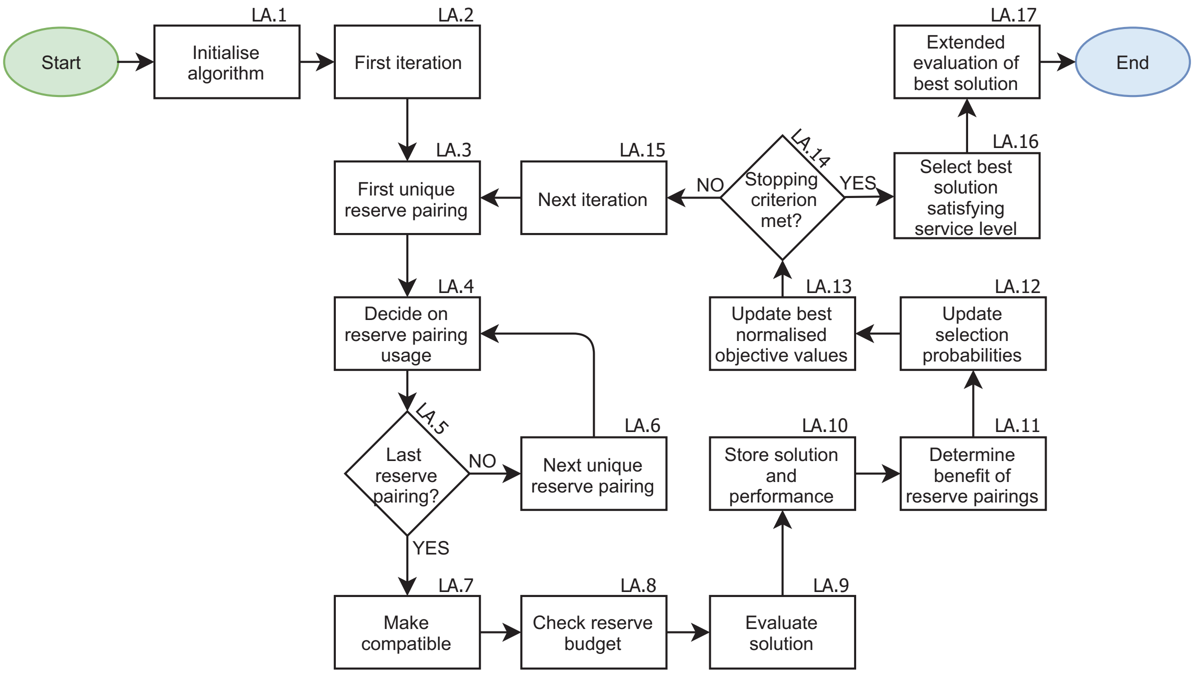

Figure 5.3: Flowchart of the LAST optimisation algorithm that has been adapted for the airline reserve crew pairing problem.

Initialisation (LA.1)
Equal to the random search algorithm, first the initial pattern density should be determined. This value determines the probability that each decision variable (i.e. reserve pairing) is included in the reserve pattern, and thus determines the number of reserve pairings in the reserve pattern. However, where the random search algorithm required only one single value for the selection probability, the LAST algorithm requires that a distinct selection probability is defined per decision variable. In fact, it requires a selection probability per decision variable value. For the airline reserve crew pairing problem, it is assumed that decision variable values can either be 0 or 1. That is, a pairing is either included in the reserve pattern, or it is not. Let $p^m(i,a)$ be the selection probability of value a for decision variable i during iteration number m. Then,

$$\begin{aligned} p^1(i,0) &= 1.0 - \textit{initial density} \qquad \forall i \in I \\ p^1(i,1) &= \textit{initial density} \qquad \forall i \in I \end{aligned} \tag{5.1}$$

Furthermore, an upper bound R_{max} and lower bound R_{min} should be defined, which will be used to determine the relative solution quality of generated reserve patterns. Since these values are not known in advance, they should be estimated. The estimates should be done such that the interval between R_{min} and R_{max} is as small as possible, without generated reserve patterns falling outside the interval. The model determines the bounds by generating and evaluating ten random solutions per bound. For the lower bound, the solution with the lowest objective function value is multiplied by a factor 0.5 to obtain the lower bound. For the upper bound, the solution with the largest objective function value is multiplied by a factor 1.15 to obtain the upper bound. Preliminary experiments show that these values provide a small interval without solutions lying outside the bounds.

Finally, the best normalised objective function value associated with each decision variable value should be initiated. Since no solutions have been generated yet, this value is 1.0 for each decision variable value. Let $B(i,a)$ be the best normalised objective function value for decision variable i with value a. Then,

$$B(i,a) = 1.0 \qquad \forall i \in I, a \in A(i) \tag{5.2}$$

Updating selection probabilities (LA.11 / LA.12)
After a reserve pattern has been generated and evaluated, the selection probabilities $p^{m+1}(i,a)$ have to be updated so that the random search process becomes adaptive. For each decision variable i, for each value a, it should be checked if $B(i,a)$ is larger or smaller than $B(i,x(i))$. Here, $x(i)$ is the value of decision variable i during the current iteration m. If $B(i,a)$ is larger than $B(i,x(i)$, there is an indication that a is a worse value for decision variable i than $x(i)$, because a better solution was found with $x(i)$ than with a. Therefore, the selection probability for a should be decreased and for $x(i)$ it should be increased. If $B(i,a)$ is larger than $B(i,x(i))$, the selection probability $p^{m+1}(i,a)$ should be updated using Equation 5.3. On the other hand, if $B(i,a)$ is smaller than $B(i,x(i))$, then the exact opposite is true and Equation 5.4 should be used to update the selection probability.

$$p^{m+1}(i,a) = p^m(i,a) - w_\mu(i)\mu[B(i,a) - B(i,x(i))]p^m(i,a) \tag{5.3}$$

$$p^{m+1}(i,a) = p^m(i,a) + w_\mu(i)\mu[B(i,x(i)) - B(i,a)]\frac{[1 - p^m(i,a)]p^m(i,x(i))}{|A(i)| - 1} \tag{5.4}$$

In these equations, μ is the learning rate and w_μ is the learning rate weight. In Chapter 4.5 the reserve pairing benefit to a solution was described, indicating the contribution a pairing has in terms of prevented premium days and the burden a pairing has in terms of caused unused reserve days. These benefits are used to determine the learning rate weight w_μ in Equations 5.3 and 5.4. In Equation 5.3, reserve pairings with a high benefit should have a higher learning rate, but in Equation 5.4, pairings with a high benefit should have a lower learning rate, as shown in Equation 5.5.

$$w_\mu(i) = \begin{cases} 1 - \frac{\text{benefit(i)}}{\sum_{i=1}^{I} \text{benefit(i)}} & \text{if} \quad B(i,a) < B(i,x(i)) \\ 1 + \frac{\text{benefit(i)}}{\sum_{i=1}^{I} \text{benefit(i)}} & \text{if} \quad B(i,a) > B(i,x(i)) \\ 1 & \text{if} \quad x(i) = 0 \end{cases} \tag{5.5}$$

When a equals $x(i)$, $B(i,a)$ will never be larger or smaller than $B(i,x(i))$, which means that the selection probability of $x(i)$ is never updated using Equations 5.3 and 5.4. To resolve this, $p^{m+1}(i,x(i))$ should be updated using Equation 5.6, after all other decision variable values have been updated.

$$p^{m+1}(i,x(i)) = 1 - \sum_{a \neq x(i); a=1}^{a=|A(i)|} p^{m+1}(i,a) \tag{5.6}$$

Updating normalised objective value (LA.13)
After the selection probabilities have been updated, the best normalised objective function values associated with each decision variable value $B(i,a)$ should be updated. First, the normalised objective function value for the current solution should be computed, using Equation 5.7. Here, F is the normalised objective function value and R is the regular objective function value.

$$F = \frac{R - R_{min}}{R_{max} - R_{min}} \tag{5.7}$$

Afterwards, for each decision variable, it should be checked if the current solution has a lower normalised objective function value for the selected decision variable value than the lowest normalised objective function value found so far. If this is the case, than Equation 5.8 should used to update the normalised objective function value.

$$B(i, x(i)) = F \tag{5.8}$$

Summarising, the LAST algorithm uses the same framework as the random search optimisation algorithm: solutions are generated by randomly selecting reserve pairings based on a selection probability. However, instead of a constant selection probability, the LAST algorithm defines a separate selection probability per pairing, which is updated based on the benefit of the pairing to the resulting reserve pattern.

5.4. GRASP

The third optimisation algorithm is a construction based algorithm adapted from the generic Greedy Randomised Adaptive Search Procedure (GRASP) from Feo and Resende (1995). The algorithm is specifically adapted for the airline reserve crew pairing problem, which has never been done in the literature before. The algorithm starts from a zero solution (i.e. an empty reserve pattern) and adds reserve pairings one by one to construct feasible reserve patterns. At each construction iteration, the reserve pairing that should be added is determined by ordering all unique pairings in a candidate list based on their potential, which is a measure of the amount of premium days that can be prevented by the reserve pairing. After each reserve pairing that is added to the reserve pattern, the potential of each unique pairing is recalculated. Due to this, the algorithm is adaptive, as the potential reflects the impact of selecting the previous element. The GRASP procedure that has been designed for the airline reserve crew pairing problem is displayed in the flowcharts of Figures 5.4 and 5.5. In the remainder of this section, the steps in the flowchart are detailed further, where the numbers in the headers indicate the corresponding blocks in the flowcharts.

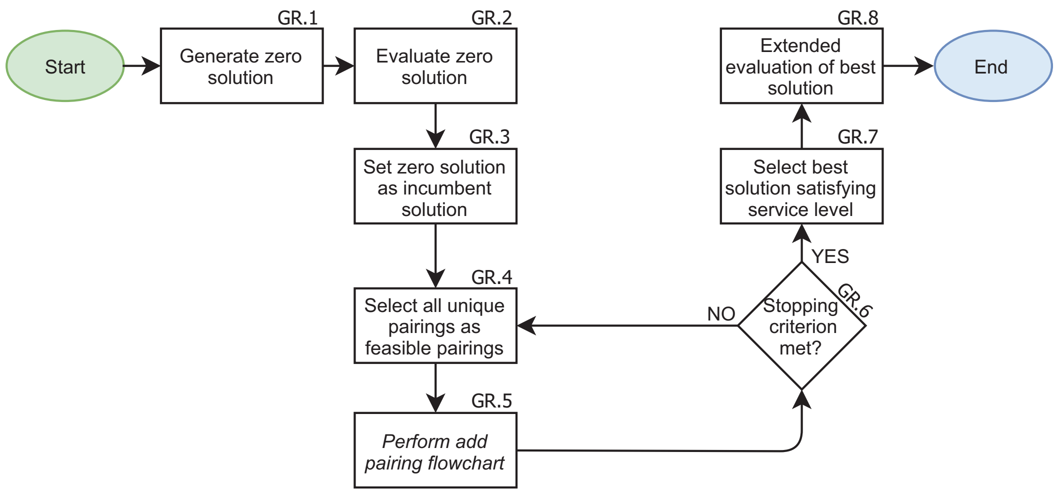

Figure 5.4: Flowchart of one repetition of the GRASP optimisation method that has been developed for the airline reserve crew pairing problem.

Zero solution (GR.1 / GR.2 / GR.3)

The GRASP algorithm adds one reserve pairing to the reserve pattern per iteration. However, before pairings can be added, first an initial solution has to be generated and evaluated. The initial solution is always the zero solution: the mathematical reserve pattern representation consists solely of zeros. The performance of the empty reserve pattern given an airline flight schedule is required to determine the potential of all unique reserve pairings, defined as the feasible pairings.

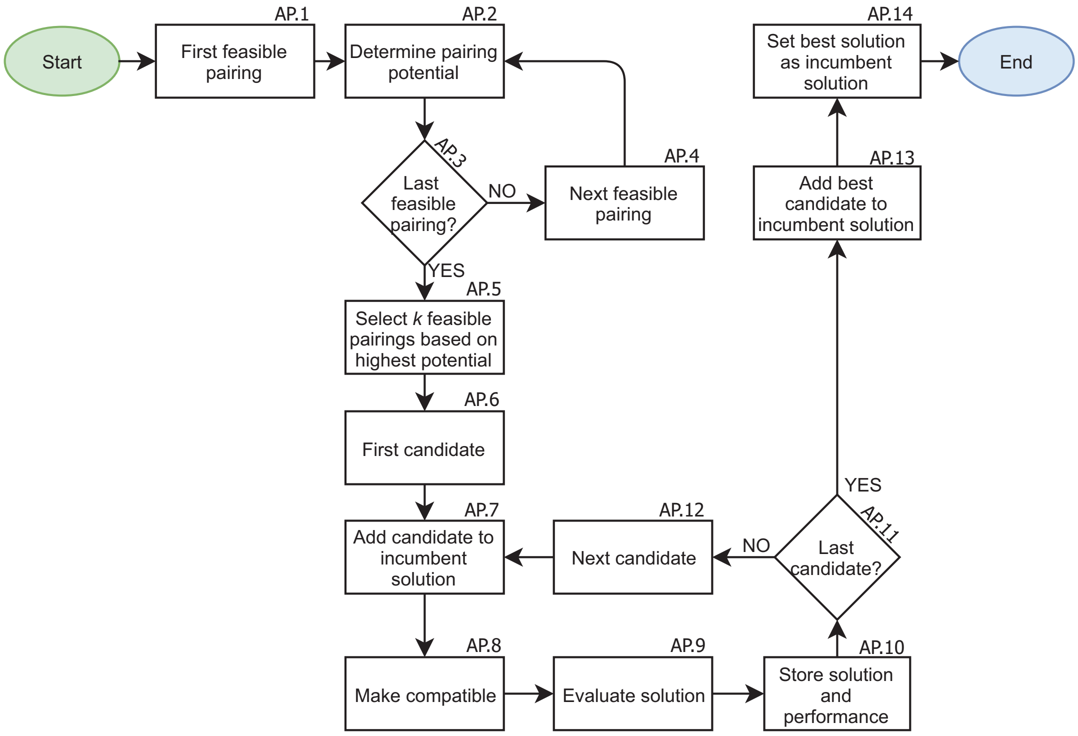

Figure 5.5: Flowchart showing how pairings are added in the construction based optimisation methods that have been developed in this research for the airline reserve crew pairing problem.

Pairing potential (GR.4 / AP.1 / AP.2 / AP.3 / AP.4)
The next step after the zero solution has been generated is to determine which reserve pairing should be added to the incumbent zero solution during the current iteration.

Firstly, all feasible pairings should be defined. For the GRASP method, the feasible pairings are the same as all unique reserve pairings as described in Section 5.1, because any infeasible pairings are already excluded from the list of unique pairings and there are no restrictions on which pairings can be added to the reserve pattern.

Secondly, the potential of each of the feasible pairings is determined. The potential is a measure of the benefit to the objective function that a reserve pairing can have when it is added to the incumbent solution. The potential of a reserve pairing is determined using a methodology similar to the analytical evaluation model, where the reserve availability decreases when a reserve pairing increasingly covers flights. For each reserve pairing, the initial potential P should be set to 0.0 and the initial reserve availability r should be set to 1.0. Then, for each flight f in the flight schedule that the reserve pairing is able to cover, the following steps should be taken.

1. Determine the decrease in disruption probability Δp caused by the reserve pairing using Equation 5.9.

$$\Delta p = p_{\text{eff},f} \cdot r \tag{5.9}$$

   where p_{eff} is the effective disruption probability of the flight, before the reserve pairing is added and r is the reserve availability.

2. Update the reserve availability using Equation 5.10.

$$r = r(1.0 - p_{\text{eff},f}) \tag{5.10}$$

3. Increase the reserve pairing potential using Equation 5.11.

$$P = P + \frac{\Delta p w_{p,f} l_f}{l_r} \tag{5.11}$$

where $w_{p,f}$ is the premium weight and l_f is the number of route days of flight f, and l_r is the length of the reserve pairing r.

Restricted candidate list (AP.5)
With the potential of each pairing known, a restricted candidate list (RCL) is constructed. The RCL includes a number of reserve pairings with the highest potential. The number of reserve pairings in the RCL depends on the number of candidate pairings k that will be evaluated and a multiplier c_k that can be adjusted by the user. For example, if $k = 5$ and $c_k = 3$, then the 15 pairings with the highest potential are included in the RCL.

From the RCL, k reserve pairings are selected as candidate pairings. This is done using a weighted random selection without replacement where the candidates with the highest potential have the highest probability of being selected. The selection probability q_i of reserve pairing i in the RCL is obtained through Equation 5.12, which enforces an exponentially decreasing selection probability, assuming that reserve pairings in the RCL are ordered by decreasing potential. The parameter values in this function have been determined from preliminary experiments, ensuring a proper balance between biasing the top pairings and not neglecting the reserve pairings lower on the RCL.

$$q_i = \frac{20^{0.9-\frac{0.8(i-1)}{kc_k-1}}}{\sum_{i=1}^{I} 20^{0.9-\frac{0.8(i-1)}{kc_k-1}}} \tag{5.12}$$

Candidate evaluation and addition (AP.6 - AP.14)
Each of the selected candidate pairings is added to a separate instance of the incumbent solution, after which the solutions are evaluated by the simulation model. To determine which of the candidate pairings should be added to the incumbent solution of the next iteration, a selection policy is followed. The user of the model is able to choose one out of three selection policies. The selection policies are:

(a) Select the solution with the lowest number of unused reserve days. This enforces that solutions arise that have high reserve utilisation rates.

(b) Select the solution with the lowest objective value. This results in the steepest possible descent in objective value, possibly at the cost of low reserve utilisation rates.

(c) Select the solution with the lowest combined value of unused reserve days and objective value, where the unused reserve days have a weighing factor of 0.75 and the objective value has a weighing factor of 0.25. This policy aims to combine the two policies above, choosing the solution that decreases the objective function the most when multiple solutions have a comparably good number of unused reserve days. The weights have been chosen so that the number of unused reserve days is the leading parameter, but only when the objective function is much better for alternatives, the objective function value is dominant.

Iterations (GR.6 / GR.7 / GR.8)
The algorithm keeps adding reserve pairings to the reserve pattern until the budget exceeds the value of the budget constraint. If no budget constraint value is specified, pairings will keep being added until the ratio between unused reserve days and premium days exceeds 12.0. This value is chosen so that the algorithm keeps adding pairings until the optimal budget has been surpassed by a safe margin.

From all solutions that were visited during the reserve pattern construction, the solution with the lowest objective function value, that satisfies the budget constraint, is selected as the optimal solution. Since this optimisation algorithm uses random elements in the solution procedure, there is a chance that a poor combination of reserve pairings is selected, leading to a bad optimal solution. To prevent this from happening, the model is able to repeat the entire optimisation process an arbitrary number of times. The reserve pattern with the lowest objective function value from all repetitions is then selected as the optimal solution.

5.5. GRASP-LF

The fourth optimisation algorithm is an adaption of the GRASP method and is called the Greedy Randomised Adaptive Search Procedure - Longest Flights (GRASP-LF) method. The GRASP-LF method first adds one reserve pairing per day in the flight schedule, which covers the longest flight pairing starting on that day. In case there are multiple longest flights, the earliest longest flight of that day is chosen, because there is a chance that subsequent longest flights are covered as well. After the longest flight on each day is covered, the regular GRASP method is started with the incumbent solution, as shown in Figure 5.6.

The idea of first covering the longest flight per day is taken from reserve pattern design principles that are currently used in practice when reserve patterns are created manually. In these patterns, it is aimed to cover a maximum number of flight pairings with one reserve pairing, which results in covering the longest flight on each day. Using this policy in an automated model has two advantages: (1) generated reserve patterns are more likely to cover a high number of flights in the flight schedule, which decreases the expected number of premium days, and (2) the generated reserve patterns will reflect the current reserve patterns, which is helpful for validation purposes.

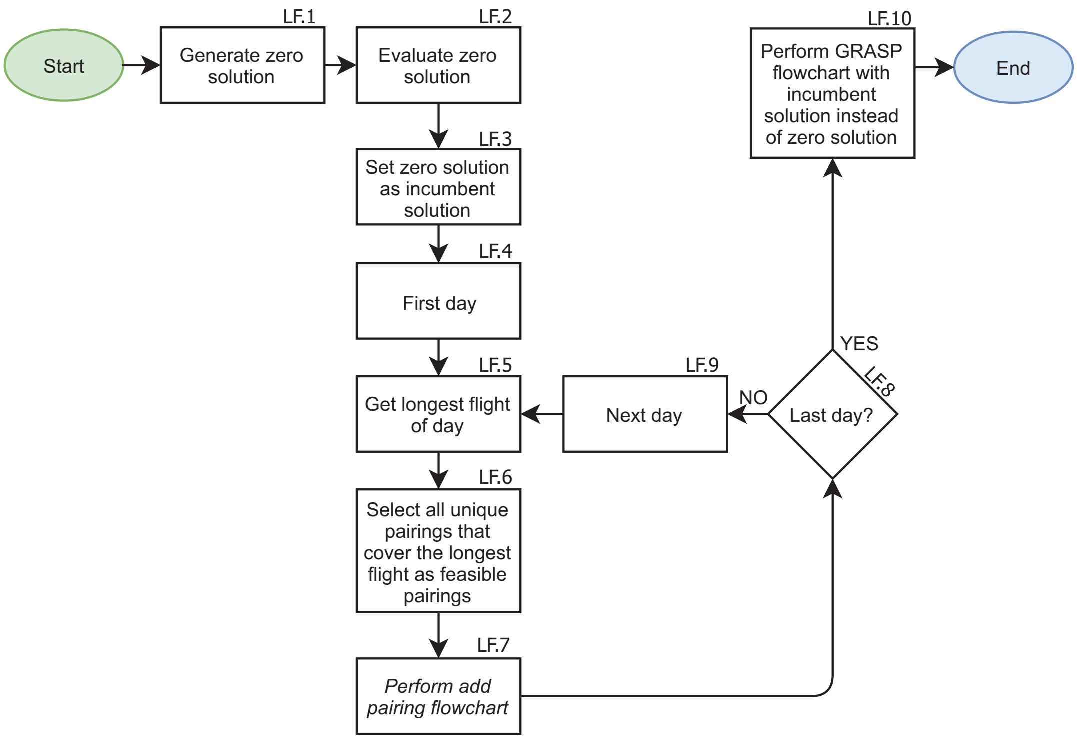

Figure 5.6: Flowchart of one repetition of the GRASP-LF optimisation method that has been developed for the airline reserve crew pairing problem.

6 Comparison experiments

This chapter considers the experiments that are performed to compare the reserve pattern optimisation algorithms. Firstly, the comparisons that are done between the algorithms are described in Section 6.1. Secondly, the results of the experiments are presented in Section 6.2. Finally, the results are validated in Section 6.3 and discussed in Section 6.4.

6.1. Experiment descriptions

A number of experiments will be done to compare the optimisation algorithms with each other and with the manually constructed reserve pattern:

1. **Objective value**
   For the first experiment, the optimisation algorithms will be run and the solutions from the algorithms and the manual solution will be compared by objective function value. This can be considered to be the primary experiment that is done, since it analyses which algorithm is able to create solutions with the lowest cost, while satisfying the required service level. In this experiment the solution process of the algorithms is also analysed and discussed.
2. **Premium days**
   Since both factors in the objective function are different in magnitude, the second experiment aims to determine which of the algorithms is able to minimise the number of premium days, given that a fixed budget is imposed. Equal to the first experiment, comparisons with the manual solution are done. On top of this, the best algorithms for the first and second experiments will be compared to each other, to see if the best optimisation method differs per objective.
3. **Dynamic reserve pairing**
   In the third experiment, the potential of automated reserve crew pairing will be demonstrated by considering seasonality effects in crew absence data. Instead of solving the problem scenario using average crew absence data, the problem will be solved separately for different time periods, with adjusted crew absence data. By comparing this against the static solutions of the first experiment, the difference in objective value, and thus the advantage of dynamic reserve crew pairing can be analysed.

Model settings
Following from a pre-experiment sensitivity analysis, that is described separately in Chapter 7, the settings in Appendix B are used for the experiments. Furthermore, the required service level is derived from the manual solution, requiring less than three flights being flown at premium cost in 97.1% of the time.

6.2. Results

This section presents and analyses the results of the experiments that were done to compare the optimisation algorithms. The results are discussed per experiment.

Experiment 1: Objective value

The first experiment aims to compare the optimisation algorithms by their optimal objective function values. The experiment has been performed three times for each optimisation algorithm, and the best result per algorithm is used for the comparison. The main performance measures for the results from this experiment are shown in Table 6.1, and the full reserve patterns per method can be found in Appendix C. To evaluate the generated reserve patterns, the simulation model required 24.4 seconds on average to simulate 25000 weeks. Given that the reserve patterns are complex to evaluate because of the combinations of pure and mixed reserve pairings with varying lengths, the simulation model is fast enough based on the project specifications.

It can be seen that the GRASP optimisation method achieves the best result with respect to the objective function, with a value of 41.44. Compared to the manually constructed reserve pattern, a 12.4% improvement is observed. The percentage improvement translates to a saving of 5.9 working days on a weekly basis. The corresponding size of the total workforce in this example equals 104 FTEs. Given that one FTE amounts to five working days per week, a total decrease of 1.1% of the total workforce costs can be projected. The LAST and GRASP-LF methods also manage to yield lower objective function values than the manual solution, with 2.7% and 5.7% improvements, respectively. The random search method is the only algorithm that is unable to improve on the existing solution.

Table 6.1: Performance measures for experiment 1 for the manual solution and all optimisation methods.

Performance measure	**Manual**	**Random**	**LAST**	**GRASP**	**GRASP-LF**
Objective function value	**47.31**	**52.77**	**46.03**	**41.44**	**44.61**
Reserve budget	45	49	43	38	42
Premium days	2.31	3.77	3.03	3.44	2.61
Service level	0.9714	0.9773	0.9785	0.9734	0.9706
Optimisation time [s]	-	4343	5078	498	568
Reserve pairings (pure)	13 (4)	11 (5)	10 (4)	10 (4)	12 (3)
Unused reserve days	21.97	23.93	18.82	14.92	19.20
Flights covered	77	70	75	74	76
Disruption improvement	81.7%	81.1%	84.9%	82.5%	82.7%

It can be seen that the service level constraint functions as it is supposed to, since the service level is comparable across the manual solution and the computer generated solutions. The improvements in objective function value are obtained by a decrease in the reserve budget that is needed to obtain the required service level. This result could be expected, since the reserve budget is a larger component in the objective function than the number of premium days, as was explained in Chapter 4.1. In other words, it is more efficient to decrease the number of reserve days than the number of premium days. For example, the solution generated by the GRASP method allocates seven reserve days per week less than the manual solution, but this generates on average only 1.13 premium days per week extra.

The disruption improvement is defined as the average percentage improvement for all flights between the initial and effective disruption probabilities. The results show that high values for this metric do not guarantee a low number of premium days too. For example, the LAST algorithm achieves the highest percentage disruption improvement, but the manual solution yields the lowest number of premium days. This can be explained by considering the flights that are not covered for each reserve pattern. It can be seen that higher values in premium days are related to longer flights not being covered, which lead to a relative high number of premium days. This is supported when looking at the number of flights covered: solutions with a large number of covered flights in general lead to low numbers of premium days. Similarly, the percentage disruption improvement and the service level are also correlated, because both of these metrics are indications for the number of disruptions not being covered. The disruption improvement percentage also depends on the ratio between pure and mixed reserve pairings. Mixed reserve pairings cause secondary disruptions, which influence the effective disruptions probabilities of certain flights. This is why the LAST algorithm scores better on the disruption improvement percentage compared to the GRASP and GRASP-LF methods. The reserve patterns generated by the GRASP and GRASP-LF methods both feature some flights of which the effective disruption probabilities are not reduced as a consequence of secondary disruptions, even though they are covered by reserve pairings.

When considering the required optimisation time of the solution methods, large differences are found. The GRASP method is the fastest with a required optimisation time of 498 seconds, whereas the GRASP-LF method needed 568 seconds to generate the solution. This marginal difference can be explained by the total number of iterations that were performed over all repeats by each optimisation algorithm until the stopping criterion was met, which is 71 for GRASP and 74 for GRASP-LF. This value is higher for the GRASP-LF method because it requires to use long reserve pairings in the first iterations to cover the longest flight on each day. The results show that these long pairings lead to a high number of unused reserve days (following from a low reserve usage rate) and only prevent a low number of premium days (i.e. they are inefficient). Therefore, the stopping criterion is met at a higher number of iterations. This effect is enlarged due to the fact that solutions with more reserve pairings take longer to evaluate. The optimisation time of the random search and LAST methods are a order of magnitude larger than for the other two methods, which is also explained by the higher number of iterations (4000) that is performed for these methods. The LAST algorithm is slower than the random search algorithm for two reasons: (1) the selection probabilities and best normalised objective values have to be updated each iteration, and (2) more solutions are evaluated because the bounds are recalculated during the optimisation process.

Figure 6.1 shows the progression of the optimal objective function value for the random search and the LAST optimisation methods. The combination of the initial reserve pattern density and the budget adaptions that have been described in Chapter 5.2 cause a steep descent in optimal objective function value in the initial optimisation phase. The initial density value results in reserve patterns with high budgets. As soon as a solution is found that satisfies the service level constraint (as indicated by a encircled solution in the graph), the random search and LAST optimisation methods continue searching for reserve patterns with lower but comparable reserve budgets. In time, this decreases the reserve budget and thus the objective function value.

It can be seen that the random search method stops improving sooner than the LAST method. This can be credited to the adaptive search procedure that is used in the LAST method. Due to this, reserve patterns with favourable reserve pairings are generated more often, which increases the chance of improving the objective function value.

In Figures 6.2 and 6.3 the optimisation process of the GRASP and GRASP-LF methods, respectively, is visualised. The solid line shows the path of objective function values that were selected at each iteration. At each iteration, a number of dashed side paths are shown that show the objective values of the other candidate solutions that were evaluated. In Figure 6.2, going from iteration two to three, the effect of the selection policy can be seen. Instead of selecting the candidate with the lowest objective function value, the candidate with the lowest weighted combination of unused reserve days and objective function is selected.

In these figures the nature of the objective function and the effect of the service level constraint can be seen. Due to the larger factor of reserve budget compared to premium days, the objective function almost exclusively increases at each iteration. That is, each reserve pairing that is added results in a sharper increase in reserve budget than the decrease in premium days. Therefore, to avoid the optimal solution from having an impractically low reserve budget, the service level constraint implicitly enforces a minimum budget. The visited solutions (i.e. feasible solutions) that satisfy the service level constraint are encircled in the graph. Of all feasible solutions, the one with the lowest objective function value is chosen as the optimal solution. This solution is indicated with a filled circle in the graph. As can be seen, the optimal solution does not have to come from the selected candidates. Instead, any solution that has been visited by during the optimisation process is eligible to be the optimal solution.

The difference between the GRASP and the GRASP-LF is notable in the first seven iterations (during which they are different). Because the GRASP-LF has to cover the longest flight on each day, inefficient long reserve pairings, as explained above, have to be used. Therefore, the followed path in the GRASP-LF algorithm increases sharper during the first seven iterations. However, it can be seen that the objective function increase after the first seven iterations is sharper for the GRASP method than for the GRASP-LF method. This is a result of the high potential pairings that the GRASP algorithm has already added in the first seven iterations. These can still be added by the GRASP-LF algorithm after the longest flights have been covered.

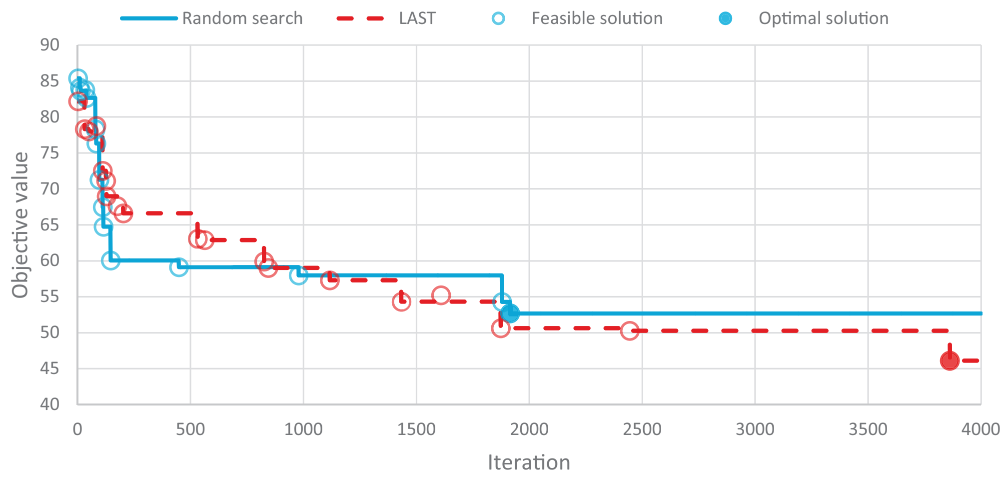

Figure 6.1: Objective function progression of experiment 1 for the random search and LAST methods.

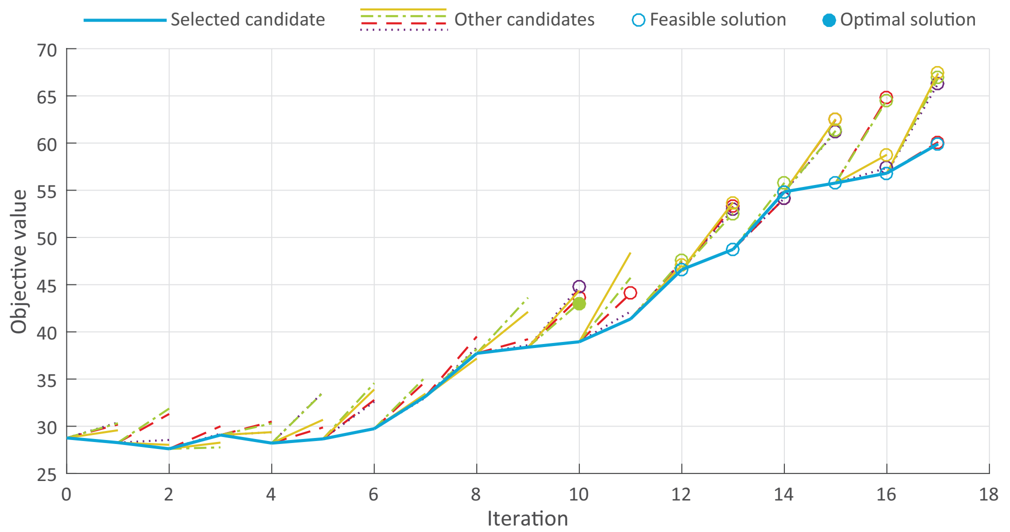

Figure 6.2: Objective function progression during the best repetition of experiment 1 for the GRASP method.

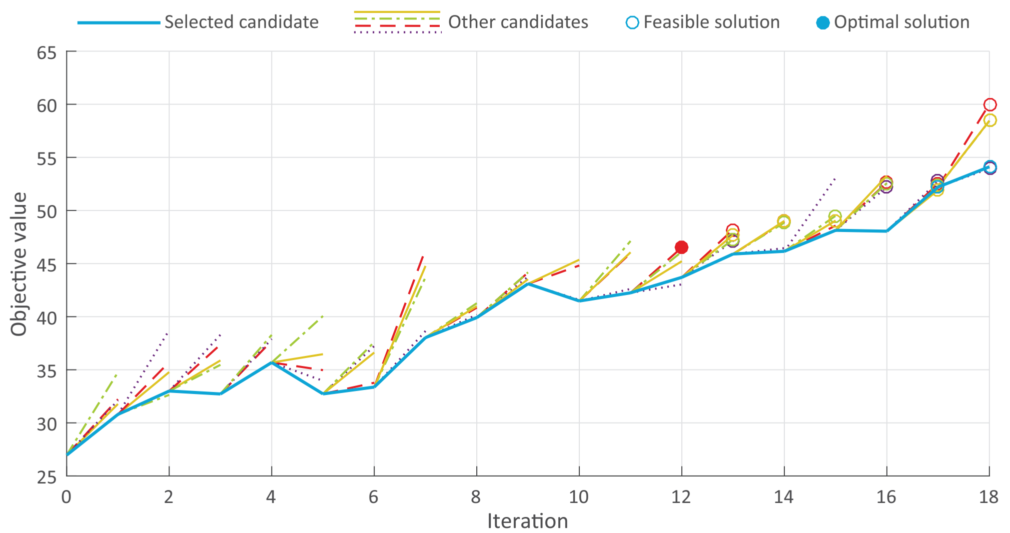

Figure 6.3: Objective function progression during the best repetition of experiment 1 for the GRASP-LF method.

Experiment 2: Premium days
In the second experiment, the reserve budget, which is the main contributor to the objective function value, is constrained. Due to this, the other contributor to the objective function value, the amount of premium days, becomes the main performance measure of a reserve pattern. Therefore, this essentially allows a comparison of the optimisation algorithms with respect to the amount of premium days that can be prevented. In this experiment a comparison between the best algorithms for experiments 1 and 2 is also made.

In Table 6.2 the main performance indicators for this experiment are shown. First of all, it can be seen that the objective function value for the manual solution is the lowest, even though the manual solution yields more premium days than all but the random search optimisation algorithms. This is a result of how the budget constraint is implemented in practice, which allows a margin of one reserve day more or less than the required budget value. This is done to increase the number of feasible solutions, with the aim of finding a feasible solution with a low number of premium days. This is especially relevant to the construction based optimisation algorithms, which visit relatively few solutions during the optimisation process. Therefore, there is a chance that during the optimisation process not a single solution is visited that exactly satisfies the budget constraint. By increasing the margin of the budget value by one, this chance is decreased.

Table 6.2: Performance measures for experiment 2 for the manual solution and all optimisation methods.

Performance measure	Manual	Random	LAST	GRASP	GRASP-LF
Objective function value	47.31	49.76	48.31	47.78	47.36
Reserve budget	45	46	46	46	46
Premium days	**2.31**	**3.76**	**2.31**	**1.78**	**1.36**
Service level	0.9714	0.9640	0.9877	0.9847	0.9806
Optimisation time [s]	-	4426	5331	509	600
Reserve pairings (pure)	13 (4)	14 (3)	12 (4)	17 (3)	13 (3)
Unused reserve days	21.97	23.68	20.58	20.58	20.37
Flights covered	77	74	75	76	78
Disruption improvement	81.7%	77.8%	89.0%	88.9%	88.0%

Secondly, compared to the first experiment, where the GRASP algorithm proved best, the GRASP-LF algorithm is more effective for the second experiment. With 1.36 expected premium days per week, it scores 41.1% better compared to the manual solution. The GRASP method manages to yield a 22.5% improved solution compared to the manual solution. The reason that the GRASP-LF method outperforms the GRASP method is because the number of flights covered with the GRASP-LF method is larger. Since the longest flight per day is covered first, even though covering these flights is ineffective considering the reserve usage rate, there are no premium days resulting from these flights. The GRASP method does not cover two of the longest flights, which results in an increase in premium days compared to the GRASP-LF method.

The random search based algorithms were unable to derive solutions that were better than the manual solution. However, when considering the service level and the disruption improvement percentage, the LAST method and the GRASP and GRASP-LF methods all outperform the manual solution. Again, these metrics are an indication of the number of uncovered flights, whereas the number of premium days also takes into account the lengths of the flight pairings. The reserve pattern of the LAST algorithm (see Appendix C) supports this statement: three of the longest flights are not covered by the reserve pattern.

In Figure 6.4 the optimal objective function value progression is shown for the random search and LAST optimisation methods, where the vertical axis represents the number of premium days in the reserve pattern. Compared to Figure 6.1, for experiment 1, the feasible solutions have not been indicated in the graph. This is because the random search based optimisation methods use the budget constraint value to make the budget of the reserve pattern compatible with the constraint value before the pattern is evaluated. That is, every solution that is considered for the random search and LAST algorithms is a feasible solution when a budget constraint is present. In theory, the methods should be more effective in finding reserve patterns with few premium days when a budget constraint is present, compared to experiment 1, which leaves the budget open. This is because the size of the feasible solution search space is limited due to the budget constraint. However, the relative performance of the algorithms between experiment 1 and experiment 2 cannot be compared, because the objective functions are of a different nature.

It can be seen that, similar to experiment 1, the LAST method outperforms the random search method. Compared to experiment 1, the optimal objective function value decreases less steeply. This can be explained by the presence of the budget constraint. Since no budget constraint is present in experiment 1, the optimisation process begins with conservative budgets. As a result, the budget, and thus the objective function value, decrease fast in the beginning. In experiment 2, the required budget is already known, so all improvement in objective function value has to come from decreasing the number of premium days. As can be expected, the improvement in the objective function value is still the fastest in the first couple of hundred iterations. The LAST algorithm manages to keep improving in objective function value for a larger number of iterations compared to the random search algorithm.

In Figures 6.5 and 6.6 the objective function progression for the GRASP and GRASP-LF methods is depicted. Compared to experiment 1, a number of differences can be noted. Firstly, the objective function (i.e. number of premium days) decreases as the number of iterations increases. This is an effect of isolating the premium days component from the objective function. Secondly, it can be seen that the selection policy is critical for this experiment. Currently, the candidate with the lowest weighted sum of unused reserve days and objective function value is selected as the optimal candidate. If instead the solution is selected that decreases the objective function the most, a much steeper descending path would be followed. In the figures, this can be seen from the dashed lines (i.e. other candidates) that are below the path followed by the selected candidates. Even though the descent in objective function value would be steeper, from preliminary tests it followed that the resulting number of premium days corresponding to the required budget is in fact larger. This is because a steepest descent by premium days causes only pure reserve pairings to be used. Pure reserve pairings do not cause secondary disruptions and thus decrease the number premium days by a larger amount. Using only pure reserve pairings results in a poor coverage over all flights, because pure reserve pairings require more budget than mixed reserve pairings. Thirdly, the number of feasible solutions in experiment 2 (i.e. solutions that satisfy the budget constraint) is smaller than the number of feasible solutions in experiment 1 (i.e. solutions that satisfy the service level requirement). For example, in Figure 6.5 only three feasible solutions were visited. This illustrates the need for the margin in the reserve budget constraint value.

Experiment 3: Dynamic reserve pairing

In the third experiment, the effects of including seasonality in the reserve pairing problem are investigated. For this, the problem scenario is solved separately for each month in the summer season for which the flight schedule is valid. For each month, the disruption probabilities per flight are adjusted for the relative crew absence rate per month. Thus when the average crew absence rate during a month was low, the disruption probabilities are decreased correspondingly. The adaption weights that are used have been derived from historical roster data and are presented in Table 6.3. For the third experiment, only the GRASP method is investigated, because this algorithm yielded the best results in first experiment.

Table 6.3: Absence rate adjustments per month for the period during which the flight schedule is valid.

	Average	**April**	**May**	**June**	**July**	**August**	**September**	**October**
Absence rate %	5.22	3.34	5.19	4.02	3.14	4.43	6.43	6.22
Weight	1.0	0.64	0.99	0.77	0.60	0.85	1.23	1.19

In Figure 6.7, the optimal objective function values per month are shown for the manually constructed reserve pattern, and the statically and dynamically generated reserve patterns from the GRASP method. The statically generated reserve pattern uses the average crew absence rate, corresponding to a weight of 1.0. That is, the static reserve pattern is based on the problem scenario of experiment 1, and the same pattern is used for each month. The dynamically generated reserve patterns are made separately for each month, where the flight schedule used different disruption probabilities per month.

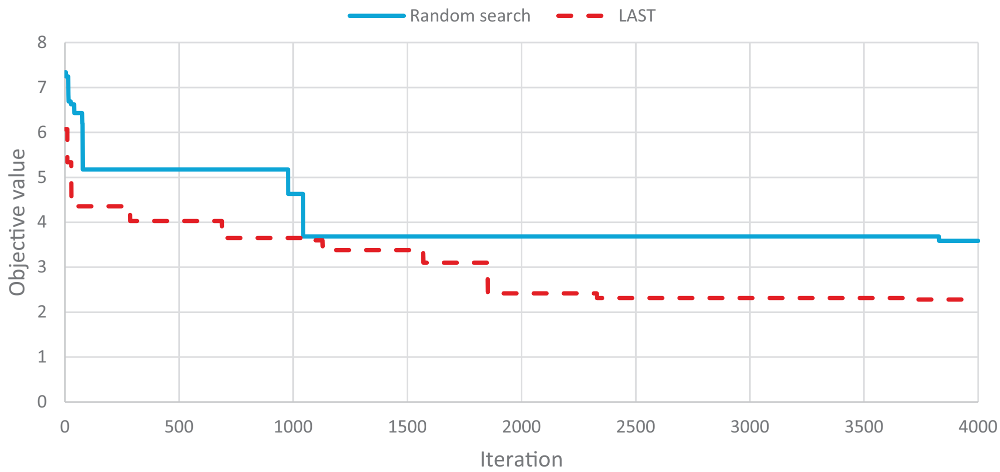

Figure 6.4: Objective function progression of experiment 2 for the random search and LAST methods.

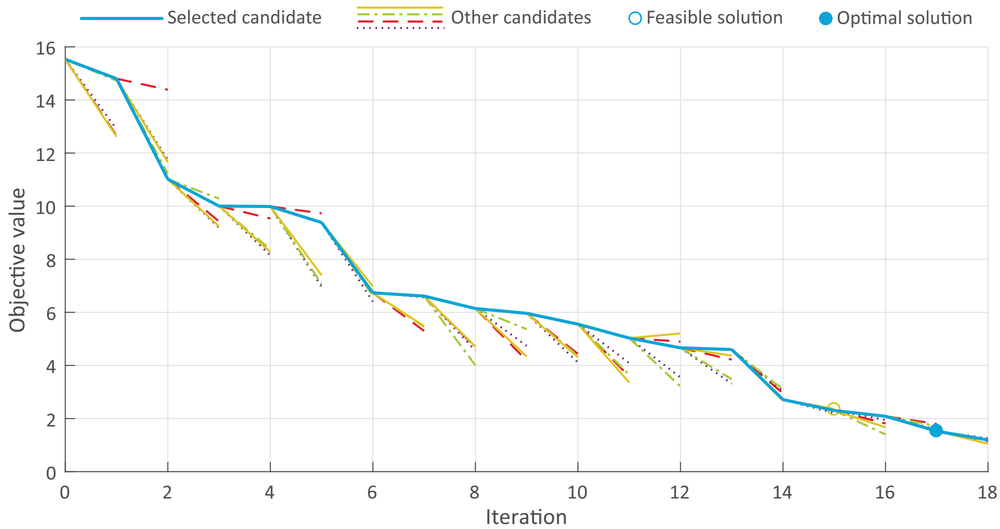

Figure 6.5: Objective function progression during the best repetition of experiment 2 for the GRASP method.

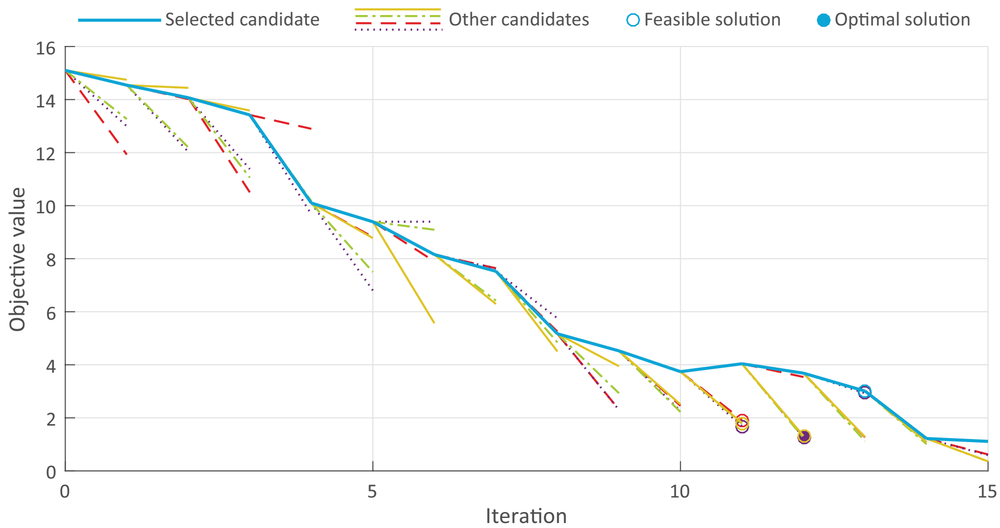

Figure 6.6: Objective function progression during the best repetition of experiment 2 for the GRASP-LF method.

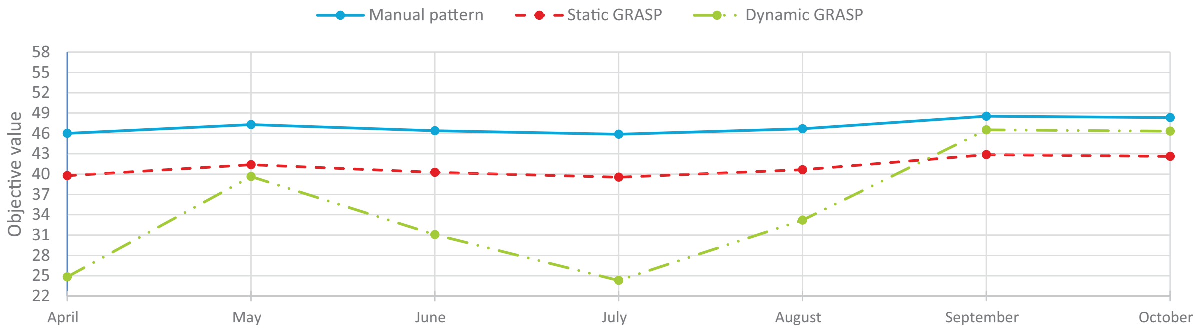

Figure 6.7: Optimal objective function value per month for the manual reserve pattern and statically and dynamically GRASP generated reserve patterns.

When the same reserve pattern is used for each month, the only variation in objective function value comes from the number of premium days, which is the minor factor in the objective function. In months where the crew absence rate is above the average value, the objective function value is larger than in experiment 1, with the opposite being true for months with lower crew absence rates. However, when different reserve patterns are created per month, large differences in the objective function values can be noticed from month to month. In months with low crew absence rates, the model is able to allocate lower reserve budgets in the optimal solutions. On average, the improvement in objective function value using monthly generated reserve patterns is 11.9 days per week (25.3%) compared to the manual solution and 5.9 days per week (14.3%) compared to the static solution.

In Figure 6.7 it can be seen that in September and October the dynamic solution is worse than the static solution. This can be explained by considering the service level, which is depicted in Figure 6.8. In September and October, where the crew absence rates are above average, the average reserve pattern is not sufficient to reach the required service level of 0.971. When looking at the dynamic solution, it can be seen that generated reserve patterns are constant in their service level over the months. This can be considered another advantage, next to a more cost effective solution. In summary, dynamic reserve crew pairing allows an airline to provide a constant and minimum level of service at minimum costs, which is a large improvement over the current situation.

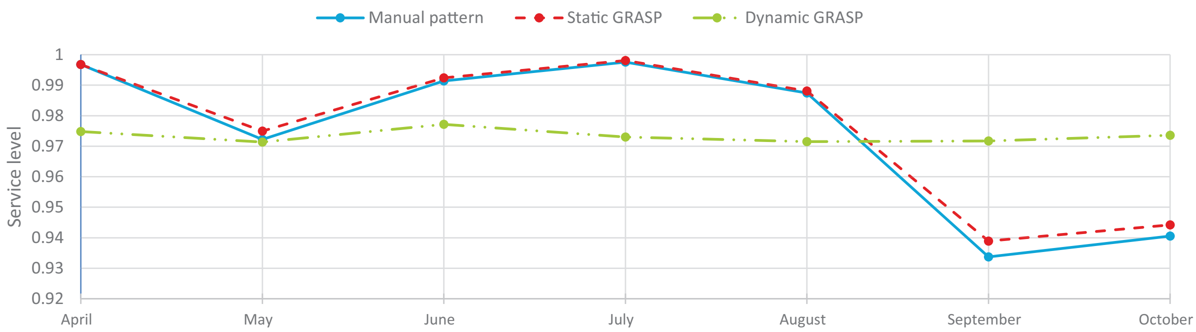

Figure 6.8: Service level per month corresponding to the manual reserve pattern and statically and dynamically GRASP generated reserve patterns.

6.3. Validation

This section presents the findings of the results validation that has been done for this research project. The aim of the validation process is to ensure that the generated results (i.e. the reserve patterns) are sufficiently accurate for their intended purpose, which is use of the reserve patterns in practice. The validation process for this project consisted of three parts:

- A comparison of the generated reserve patterns with the existing manually constructed reserve patterns;
- Face validation with the KLM crew scheduler who created the existing manually constructed reserve patterns, but is now employed as a crew controller, who resolve crew disruptions and thus

use the reserve pairings;

- Qualitative comparison of the performance measures with data from practice of the solutions after evaluation using the simulation evaluation model.

The conclusions from the results validation process are the following:

- **The results are valid within the project scope** Following from the comparison of the manually constructed reserve patterns and the consultation of the crew scheduler, the reserve patterns have been shown to be usable as reserve patterns in practical situations. Within the scope of the model, the generated reserve patterns are valid to cover the corresponding input flight schedules. From this it follows that the relevant labour agreements and regulations have been implemented correctly in the model, concluding that the model results are valid.
- **Factors that limit the validity** Given that the model scope has been defined such that the airline reserve crew pairing problem can be isolated from the overall crew scheduling problem, a number of points were brought up during the expert consultation that can improve the validity of the results for a wider scope.
  - **Recovery measures** The model assumes that the only recovery measures for a crew disruption are reserve usage and premium flying. Moreover, when reserves are still available for recovery, they have to be used. In practice, reserve pairings can sometimes be saved for specific flights, for which experience has shown that these flights are difficult to cover using premium flying. Instead of using a reserve pairing, it is chosen to divert to premium flying. This is supported by the qualitative data comparison that has been done. The number of premium days in practice was found to be larger than in the model. However, it is considered outside the scope of this project to implement the decision making process of the crew controller in the model.
  - **Flight coverage** It was found that it is preferred from the perspective of the crew controller that all flights in the flight schedule are covered by at least one reserve pairing, preferably starting on the same day as the flight pairing. The model does not currently support this, but a model extension through the implementation of an additional constraint can provide this option in the results.
  - **Reserve pairings** Another preference from crew controllers is that a minimum number of pure reserve pairings is included in the reserve pattern. Even though mixed reserve pairings are useful to cover a wide spread of flights with a limited reserve budget, primary disruptions eventually have to be covered by a pure reserve pairing in order to stop the snowball disruptive effect. By enforcing a minimum number of pure reserve pairings, it can be ensured that a minimum number of primary disruptions following from crew absence can be covered during operations.

Summarising, the model provides sufficiently accurate results given the present regulations and labour agreements. However, for the model to be more relevant to practical operations, some of the model assumptions should be lifted so that the validity of the results is improved.

6.4. Discussion of results

Having completed all tests for the comparison experiments, the results can be discussed. The reserve pattern simulation evaluation model is considered first. This model has been shown to be effective in the evaluation of reserve patterns, which can be defined in a more complex manner compared to the existing scientific state of the art. This follows from the use of mixed reserve pairings and pairings of variable length. Despite the increased complexity, no impractical computational times are experienced. A realistically sized flight schedule can be simulated for a thousand weeks in approximately one second. It was found that a similar analytical model became too complex for the reserve pattern flexibility required in this research, yielding impractical computational times. Another advantage of the simulation model is that it is straightforward to include extra functions or constraints in the model. The main drawback of the simulation model, the stochasticity in the model outcome, can be mitigated by increasing the number of simulation repetitions. Because of the computational efficiency of the model, this can be done without loss of practical use.

Secondly, the optimisation algorithms are considered. In total, four algorithms have been developed and compared. The LAST, GRASP, and GRASP-LF methods are novel methods in solving the airline reserve crew pairing problem. Across the first two tests, the random search and LAST optimisation methods are outperformed by the construction based methods (GRASP and GRASP-LF), both in terms of objective function values and required optimisation times. The best result in the first test was generated by the GRASP algorithm, which yielded a 12.4% improvement over the manually constructed reserve pattern. This improvement was primarily achieved by a reduction in reserve budget. However, for the second test, where the reserve budget was constrained, the GRASP-LF method was shown to be most effective. This method yielded an improvement of one premium day per week (a 41.4% improvement) compared to the manual solution.

The reason that the random based optimisation techniques are not effective for this problem is due to the size of the solution space. A high number of iterations is required for the random search and LAST methods to converge towards high quality solutions, resulting in impractically high computational times. Combining this with a simulation evaluation method results in optimisation methods that do not yield satisfying results within the specified run time. The LAST method does manage to converge at a faster rate than the random search algorithm, but due to the large number of decision variables it is still difficult to randomly generate patterns that consist of a set of pairings which work well together.

The novel GRASP and GRASP-LF methods are able to generate reserve patterns much faster because only one reserve pattern is generated per repetition, even though only one pairing is added per iteration. Due to this, fewer reserve patterns have to be evaluated, leading to lower computational times. The fact that reserve pairings are added one-by-one is also beneficial for the solution quality. Since the potential of pairings is re-evaluated after each pairing that is added, the GRASP and GRASP-LF algorithms are able to adapt to the effect of reserve pairings that were added before. Due to this, these methods are able to create reserve patterns that effectively cover the reserve demand following from the flight pairings in the airline schedule.

Depending on the goal of the decision maker, either the GRASP or GRASP-LF methods can be used. When it is aimed to automatically size the reserve budget that should be allocated to achieve a minimum service level, which is a novel application with respect to airline reserve crew pairing, the GRASP method should be used. However, when a reserve budget has already been decided on, the GRASP-LF method is better able to allocate these reserve days so that a lower number of premium days is obtained.

Finally, the effects of dynamic reserve crew pairing have been considered in the third test. From this test it follows that creating reserve patterns for smaller time spans is beneficial compared to static reserve crew pairing. Dynamic reserve crew pairing allows to include seasonality effects in the determination of reserve budgets, which prevents oversized reserve patterns in time periods with low crew absence rates and undersized reserve patterns in time periods with high crew absence rates. In combination with the novel service level constraint, the reserve patterns can be regularly adapted to provide constant service levels over time.

7

Sensitivity analysis

This chapter presents the results of the pre-test sensitivity analysis that has been done to solve the reserve crew pairing problem. The aim of the sensitivity analysis is twofold: (1) to determine the extent to which the model output changes when the input parameters of the model are changed, and (2) to determine which input parameter values yield the best results. In Section 7.1, the analysis of parameters relevant to the evaluation model are discussed. Thereafter, Sections 7.2 and 7.3 present the sensitivity analysis for the random search based and construction based methods, respectively.

7.1. Evaluation model

For the evaluation model, the reserve use policy, the mixed flights assignment policy and the input schedule size have been investigated. For each experiment in this section the manual reserve pattern has been used for the comparisons.

Use policy

The reserve use policy defines which reserve pairing is used to cover a disrupted flight pairing. There are two options: (1) by earliest start time or (2) by minimum number of wasted reserve days. In Table 7.1, the results for this analysis are shown. The evaluation model yields better objective function values for the minimum waste days policy with slightly improved run times. The difference in objective value is the result of a lower number of premium days. This shows that using reserve pairings are the best fit for a disrupted flight is better than using the reserve that has started earliest. An explanation for the higher number of premium days for the earliest start time policy is that long reserve pairings are likely to be used for short flights, leaving long flights vulnerable to premium flying. Therefore, the minimum waste days usage policy is chosen for the experiments.

Table 7.1: Comparison between reserve usage policies in the evaluation model.

Parameter value	**Average objective value**	**Average run time** [s]
Minimum waste days	47.31	25.2
Earliest start time	48.29	26.5

Assignment policy

The assignment policy determines which flight is assigned to mixed reserve pairings in the reserve pattern. Again, two possible settings can be chosen: (1) equal probability for each possible flight that can be assigned or (2) the flight with the lowest initial disruption probability. In Table 7.2 it can be seen that the difference between both settings is negligible. Hence, the equal probability setting is chosen to be used for the experiments, since it is a better reflection of the current way of working in practice. An explanation for limited impact of this setting is that absence of reserve crew is neglected in this research. Otherwise, reserve crew absence can be related to the disruption probability of the mixed flight, where flights with low disruption probabilities indicate increased reserve crew presence. Following this, more notable differences between the mixed flight assignment policies can be noted.

Table 7.2: Comparison between mixed flight assignment policies in the evaluation model.

Parameter value	**Average objective value**	**Average run time** [s]
Equal probability	47.34	25.1
Lowest disruption first	47.31	25.2

Schedule size

To test the sensitivity of the evaluation model to the size of the input schedules, a range of tests is conducted in which the number of flights and reserves in the input schedules is multiplied by different factors. For each test, the flight schedule and reserve pattern are copied a number of times. For example, if the multiplication factor is 3, then there are 234 flights and 39 reserve pairings, instead of 78 flights and 13 reserves. For each departure, three identical departures take place, and for each reserve pairing, three identical reserve pairings start at the same time. Figure 7.1 shows the primary assessment criteria for the different multiplication factors. The figure shows that the objective function value increases linearly with increasing schedule size. When decomposing both factors of the objective function, linear trends are seen for both the reserve budget and number of premium days. In contrast, the required evaluation time increases exponentially, at a slightly faster rate than the linear trend. This is explained by considering that besides the increasing number of flights, the number of feasible reserves for recovery per flight also increases. Resulting, more reserve pairings should be considered in recovery, which takes longer per flight.

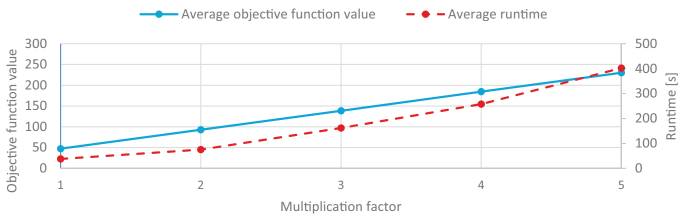

Figure 7.1: Objective function value comparisons between different input schedule sizes in the evaluation model.

7.2. Random search based methods

This section describes the sensitivity analyses that have been done for the parameters of the random search based methods. The pure random search and LAST methods are evaluated for the number of iterations, the initial schedule density, and the number of simulation repetitions. For the LAST method, the sensitivity of the learning policy, learning rate and random fraction are also evaluated.

Iterations

To investigate the sensitivity of the number of iterations, the random search and LAST methods were executed with 10000 iterations. The best objective function value visited up to each iteration was stored and plotted on the horizontal axis in Figure 7.2. The run time has also been plotted in the figure, but has been calculated as a fraction of the total run time at 10000 iterations. Therefore, it should be used as an indication for the run time, instead as an exact prediction. It can be seen that the largest improvements in objective function values are made in the first 3000 iterations, after which the rate of convergence decreases. The LAST algorithm is able to decrease the optimal objective function value at a faster rate, which can be accredited to the adaptive search procedure. For the experiments, an iteration number of 4000 is chosen, because a further increase would lead to run times that are practically out of scope for this research.

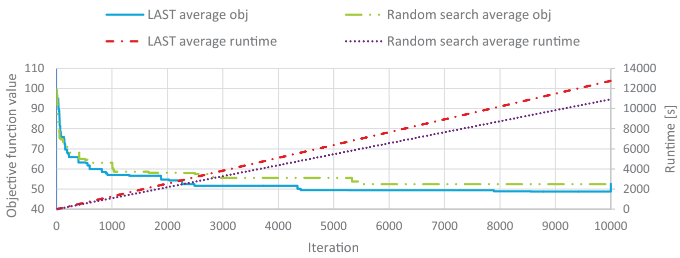

Figure 7.2: Objective function value comparisons between different iteration numbers for the random search and LAST methods.

Initial density

The initial density determines the probability that each reserve pairing in the list of unique reserve pairings is included in the reserve pattern during the first iteration. Thereafter, the selection probabilities are adapted for the LAST algorithm, but remain constant in the random search algorithm. Resulting from this, it can be seen in Figure 7.3 that the initial density has a larger effect on the objective function value for the LAST algorithm than for pure random search. The LAST algorithm performs better for initial density values of 0.025 and 0.05 than for 0.1 and 0.2. This is because the number of reserve pairings that are selected in the reserve pattern are closer to the optimal solution for the lower initial density values than for the larger values. For the pure random search, there is no difference between the initial density values, because the service level constraint enforces a budget after the pairings have been selected using the initial density.

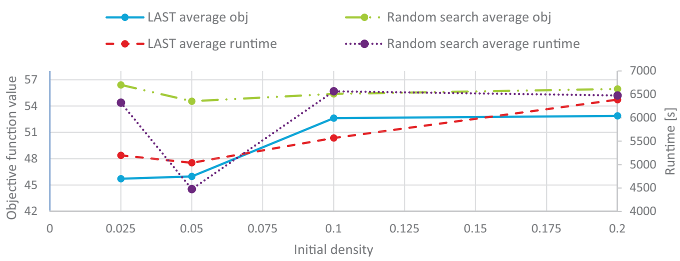

Figure 7.3: Objective function value comparisons between different initial density values for the random search and LAST methods.

Simulation repetitions

Figure 7.4 shows the average optimal objective function value for a number of simulation repetition values. Correspondingly, in Figure 7.5 the acquired service level of the optimal solution is plotted for the same simulation repetition values. The general trend that is apparent is that when the number of simulation repetitions increases, the objective function value also increases. This is because for low numbers of simulation repetitions, the generated solutions are evaluated less carefully, leading to increased bias in the service levels. When the optimal solutions are evaluated further, the service level turns out to be lower than initially estimated, which can be seen in Figure 7.5: for low values of N, the service level is lower than required in the experiments. Therefore, to ensure that the generated optimal solutions from the pure random search and LAST methods have limited bias for solutions with too low service levels, a simulation repetitions value of 1500 has been chosen. This does come at a computational cost, but a higher priority to accurate service level estimation is given.

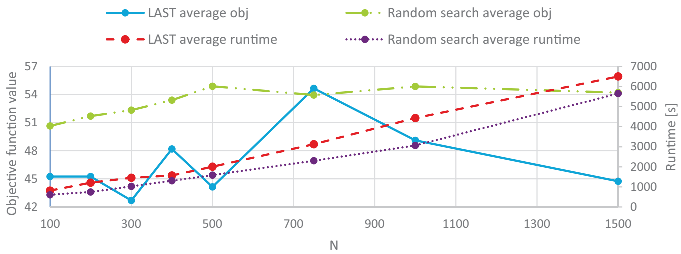

Figure 7.4: Objective function value comparisons between different amount of repetitions for the random search and LAST methods.

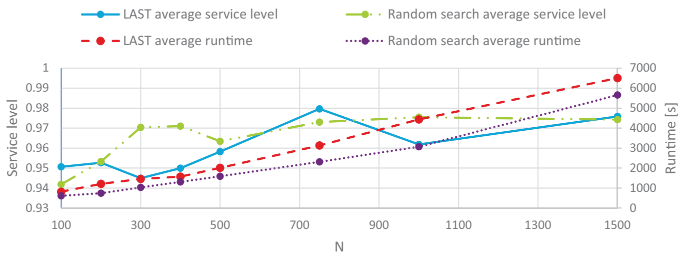

Figure 7.5: Service level comparisons between different amount of repetitions for the random search and LAST methods.

Learning policy

The learning policy determines which metric is used to adapt the selection probabilities in the LAST algorithm. The learning policy parameter basically serves as a surrogate objective function that is only used for the learning aspect of the LAST algorithm. In Table 7.3 the results for three learning policies are shown. If the sum of premium days and reserve budget is used for the learning policy, the budget days would be dominant in this metric, and the algorithm would be unable to learn which reserve pairings are effective at covering flights and thus preventing premium days. Therefore, this metric yields high objective values. On the other hand, only using premium days for learning will create a bias for pure reserve pairings, since the number of reserve days is irrelevant in this metric. A combination of premium days and unused reserve days results in the best performance. This metric incorporates the fact that reserve pairings should use their reserve days at a high rate, but also aims to minimise the number of premium days. Therefore, the sum of premium days and unused reserve days is used to base the learning process of the LAST algorithm on.

Table 7.3: Comparison between learning policies for the LAST method.

Parameter value	**Average objective value**	**Average run time** [s]
Premium + unused	48.10	5622.7
Premium	51.21	5678.0
Premium + budget	53.69	5681.0

Learning rate

The learning rate determines the extent to which the selection probabilities of the decision variables are adapted after each iteration, with the aim of increasing the probability that high quality solutions are identified. A balance between convergence towards high quality pairings and maintaining randomness in the search procedure should be aimed for in determining the learning rate. If the learning rate is too high, there is a chance that the LAST algorithm visits the same solutions over and over again. This is what happens for the learning rates of 0.02 and 0.04 in Figure 7.7. It can be seen that roughly the first ten decision variables have high selection probabilities, but after that there is a steep decrease towards a selection probability of zero for all other decision variables. Due to this, the diversity in reserve patterns generated is low. For the learning rates of 0.005 and 0.01, the selection probability of the first ten decision variables is lower, but decreases towards zero for other decision variables much slower. In Figure 7.6 it can be seen that the performance of the higher learning rates is inferior to that of the lower learning rates. On the other hand, if the learning rate is too low, the learning process can be too slow and high quality reserve pairings do not arise. Therefore, a learning rate of 0.01 is chosen for the experiments.

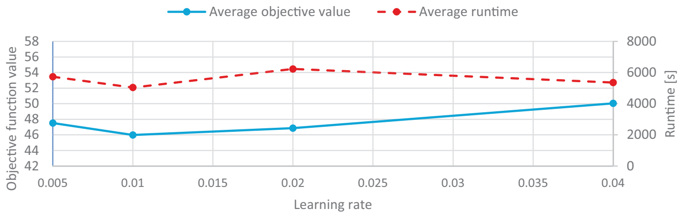

Figure 7.6: Objective function value comparisons between different learning rate values for the LAST method.

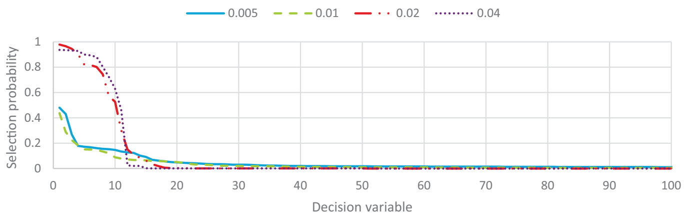

Figure 7.7: Selection probability of decision variables after the final iteration, in descending order, for different learning rate values for the LAST method.

Random fraction

The random fraction determines how many reserve patterns are generated using the initial density selection probability values and how many are generated using the adapted selection probability values following from the LAST algorithm. The aim of the random fraction is to ensure that some solution diversity is always maintained, even if the adapted selection probabilities have fully converged. In Figure 7.8, the average objective value and the run time for different random fractions have been plotted. For a random fraction of two (i.e. one in two generated solution uses the initial density) the objective values are much higher than for the other evaluated parameter values. An explanation for this is that the effect of the LAST algorithm is too small for it to have a positive effect on the solution value. Over all evaluated values, no clear trend is identified, but for a random fraction of three low objective function values were found. Therefore, this value was used in the experiments.

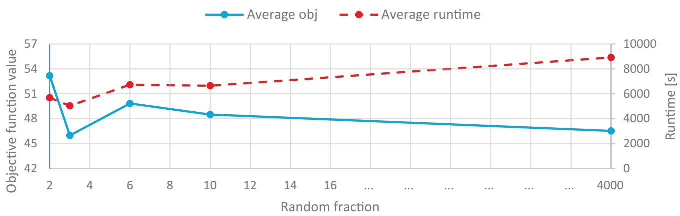

Figure 7.8: Objective function value comparisons between different random fraction values for the LAST method.

7.3. Construction based methods

The sensitivity of five parameters has been investigated for the construction based methods. These are the number of repeats, the population size, the candidate multiplier, the number of simulation repetitions, and the selection policy. Both the GRASP and GRASP-LF methods are considered simultaneously. At the end of this section, the sensitivity of the GRASP method to the service level constraint is considered.

Repeats

The number of repeats defines how many times the optimisation algorithm is executed, before the optimal solution is selected from the repetition with the lowest objective value. It is expected that a higher number of repeats leads to lower objective values but an increased run time: the probability of encountering a repeat with a low objective function value increases when more time is invested in generating solutions. Figure 7.9 supports this hypothesis, although the trend is strongest for lower number of repeats. The objective value stops improving when the number of repeats surpasses four. This indicates that doing four repeats already yields a sufficiently large chance of finding high quality solutions. Therefore, this value has been chosen for the experiments. It can also be seen that the trend is stronger for the GRASP-LF method than for the GRASP method, meaning that the GRASP method is more consistent in generating high quality solutions.

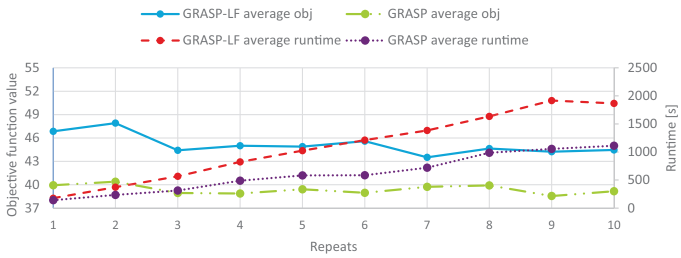

Figure 7.9: Objective function value comparisons between different numbers of repeats for the construction based methods.

Population size

The population size determines the number of candidate pairings that are evaluated at each construction iteration. From all evaluated pairings, the best one according to the selection policy, is selected for the next iteration. In Figure 7.10, an interesting trend can be noticed: the algorithm performance improves for increasing population sizes, with a minimum at a population size of four to five. The improvement is explained by realising that by evaluating more reserve pairings, the chance of evaluating a favourable reserve pairing increases. However, when the population size is increased further, the algorithm performance deteriorates again. Paired with this, a non-linear increase in computational time can be noticed. An explanation for this is that it is more likely that incompatible reserve pairings are added as candidates when the population size increases. When this occurs, a reserve pairing also has to be removed from the reserve pattern to make the pattern compatible again. No pairings are effectively added to some candidate solutions, which is beneficial considering the selection policy. In other words, the construction algorithms struggle to efficiently add reserve pairings to the patterns at each construction iteration, leading to deteriorated algorithm performance.

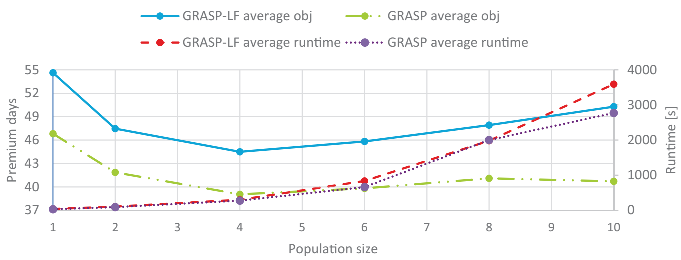

Figure 7.10: Objective function value comparisons between different population sizes for the construction based methods.

Candidate multiplier

The candidate multiplier determines the size of the pool of unique reserve pairings from which the candidate pairings in the population are selected, ranked by pairing potential. This implies that the minimum value of this parameter should be one, so that at least one unique reserve pairing is available per candidate in the population. Figure 7.11 shows the performance of the GRASP and GRASP-LF algorithms for various values of the candidate multiplier parameter. The performance of both algorithms is constant for lower values of the candidate multiplier, but a decrease in performance can be seen when all unique pairings are included in the pool from which candidate pairings are selected. For the experiments, a value of 2.5 has been chosen, which provides a balance between (weighted) randomised selection procedures and limiting the size of pool.

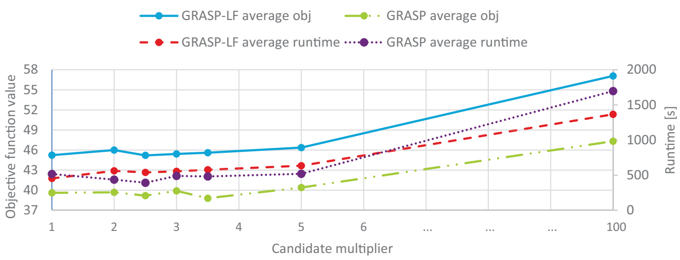

Figure 7.11: Objective function value comparisons between different candidate multiplier numbers for the construction based methods.

Simulation repetitions

With respect to the number of simulation repetitions, similar trends to the random search based methods can be noticed: an increase in the number of repetitions results in a more accurate estimation of the quality of generated solutions. This leads to higher service levels and objective function values for higher numbers of simulation repetitions. To enforce an equal bias between all four optimisation algorithms, the same number of repetitions is chosen: 1500. At this value, the estimated service levels of the optimal solutions approach the required levels, and further increasing the number of repetitions only increases required run times.

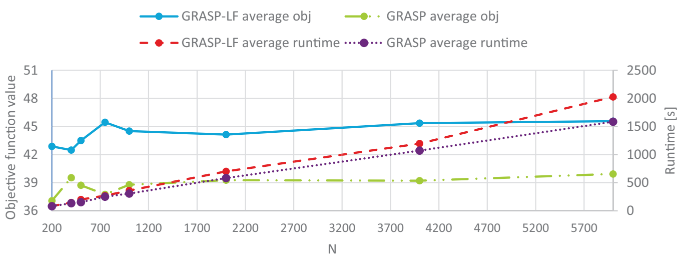

Figure 7.12: Objective function value comparisons between different amount of repetitions for the construction based methods.

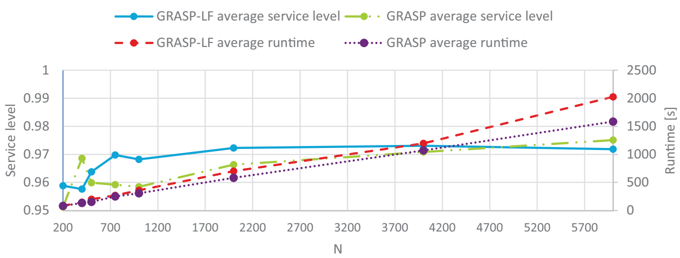

Figure 7.13: Service level comparisons between different amount of repetitions for the construction based methods.

Selection policy

The selection policy determines which candidate pairing is added to the incumbent solution during each construction iteration. Table 7.4 presents the algorithm performance for the three selection policies that have been investigated: minimum number of unused reserve days, minimum objective value, or a weighted combination of those factors. It can be seen that there are only minor variations in the performance of the selection policies, which can be accredited to the limited number of repetitions of the sensitivity analysis. This is supported by the results from the experiments in Chapter 6: in Figures 6.2 and 6.3 the followed path is the path using the weighted combination selection policy, which is almost identical to the minimum objective value.

For the second test the selection policy is more relevant. In Figures 6.5 and 6.6 it can be seen that the path followed by the weighted combination selection differs strongly from the path of the minimum objective value (i.e. steepest descent). If the minimum objective value was chosen as the selection policy, only pure reserve pairings would be added, since the number of unused reserve days (i.e. the usage rate of reserve pairings) is irrelevant. Therefore, the combination selection policy is chosen for the experiments.

Table 7.4: Comparison between mixed flight assignment policies in the evaluation model.

	GRASP		**GRASP-LF**	
Parameter value	**Objective**	**Run time**	**Objective**	**Run time**
min Unused reserve days	38.47	526.4	45.01	541.7
min $0.75 \cdot$ unused $+ 0.25 \cdot$ objective	39.77	415.6	44.29	459.6
min Objective value	39.56	455.2	45.11	410.1

Service level

In Figure 7.14 the optimal objective function value (7.14-a) and the corresponding reserve budget (7.14-b) and number of premium days (7.14-c) are shown for a range of service level constraint values. Each graph in the figure represents a value for the maximum number of flights that can be flown at premium cost per repetition of the flight schedule. On the horizontal axis the fraction of flight schedule repetitions during which this number of flights cannot be exceeded, has been plotted. Most notable from these graphs is the exponential increase in objective function value and reserve budget, and the exponential decrease in premium days, when the service level approaches values of 1.0. This is the result of the increasingly demanding smoothness in the operation of flights. When the service level is required to be 1.0, the maximum number of flights flown at premium cannot be exceeded in a single flight schedule repetition. This requires a large number of reserve days. It can also be seen that the exponential increase is sharper when the maximum number of premium flights is lower, which is also a result of the increasingly demanding operational requirements in terms of premium flights.

When the maximum number of flights equals four or five, and the service level requirements are low, the optimal solutions show that no reserves should be used. This means that the required service levels can be obtained, even if all disrupted flights are flown at a premium. Correspondingly, the entire objective function value follows from the number of premium days.

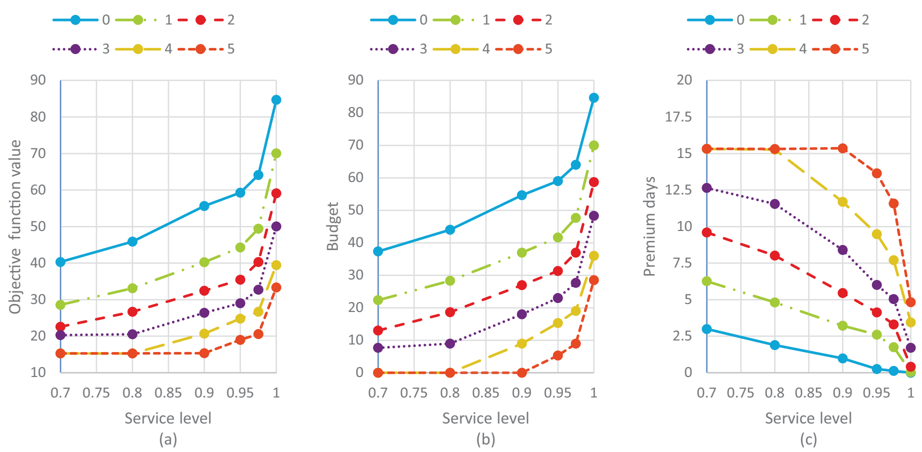

Figure 7.14: Objective function value comparisons between different values for the service level constraint for the GRASP method.

8
Conclusions and recommendations

This final chapter presents the conclusions of the research in Section 8.1, lists the research contributions of the project in Section 8.2, and offers a number of recommendations for further research in Section 8.3.

8.1. Conclusions

This thesis has considered the airline reserve crew pairing problem for long-range cockpit crew. The central research question that was defined is presented below:

How should airline long-haul cockpit reserve crew patterns be constructed in order to minimise the gap between scheduled reserve capacity and expected reserve demand?

Following a literature review, it was found that airline reserve crew has only been considered sparsely in research. The state of the art in reserve crew pairing provides limited means to create efficient reserve crew patterns for airlines in practice. It is assumed that reserve crew patterns consist solely of reserve pairings of constant length and that the reserve pairings consist only of reserve duty days. When these assumptions are used to create reserve patterns for long-haul crew in practice, large numbers of unused reserve days are the result. This is a consequence of the large variability in flight pairing length that is apparent in long-haul crew schedules, in combination with the constant length of reserve pairings. A more realistic approach is to use reserve pairings with variable lengths and reserve pairings that can be combinations of reserve duty days and regular flights. This creates the opportunity to create cost efficient reserve patterns for the extensive range of flights in long-haul airline schedules. This is most relevant to cockpit crew schedules, who are the most expensive human resources for an airline.

To address the research question, a reserve pattern evaluation model and four optimisation algorithms to generate reserve patterns were developed. The scheduled reserve capacity, which is reflected in a reserve pattern, is subjected to flight schedule simulations in the evaluation model. This results in a prediction of the expected reserve demand. Hence, using the simulation evaluation model, the gap between the scheduled reserve capacity and the expected reserve demand can be determined. The evaluation model is used iteratively in combination with one of the optimisation algorithms, which aim to generate reserve patterns that minimise the gap. The objective of the optimisation was defined to minimise the sum of the reserve budget (in days) and the number of premium days. A corresponding service level requirement was introduced to implicitly enforce a minimum reserve budget (i.e. capacity) in an intuitive manner.

For the evaluation model, both analytical evaluation and simulation evaluation were considered. It was found that for the analytical evaluation model the complexity of the operations encountered by long-haul cockpit crew was too complex to yield practically applicable computational performance. Therefore, the simulation evaluation model was used to determine the effectiveness of generated reserve patterns, being able to evaluate approximately a thousand weeks of operations in one second, for realistically sized problem sizes.

The four optimisation algorithms were a pure random search technique, an adaptive random search technique called LAST, and two adaptions to a greedy randomised adaptive construction based algorithm, called GRASP and GRASP-LF. The latter three algorithms are novel adaptions from existing standard techniques. All algorithms use characteristics specific to the airline reserve crew pairing problem to improve algorithm performance. Three comparison experiments between these algorithms allowed to make recommendations on how reserve crew patterns for long-haul cockpit crew should be scheduled. The first comparison experiment aimed to compare objective function values between the algorithms. It revealed that the construction based algorithms outperformed the random search based methods, while also yielding lower computational times. Compared to a manually constructed reserve pattern, the GRASP method yielded the best result with a 12.4% improvement requiring a computational time of 498 seconds. The percentage improvement translates to a saving of 5.9 working days on a weekly basis. The corresponding size of the workforce in this example equals 104 FTEs. Given that one FTE amounts to five working days, a total decrease of 1.1% of the total workforce costs can be projected. The main improvement in the first experiment came from decreasing the reserve budget, which is the main component in the objective function. The second comparison experiment, where the reserve budget was fixed and the number of premium days was minimised, showed that the GRASP-LF method yielded the best results. A qualitative relation between the number of flights covered and the number of premium days was found. A final comparison investigated the effect of dynamic reserve crew pairing, where reserve patterns are made for shorter time periods, using corresponding flight disruption data. This approach allows to provide constant service levels over time, compared to static solutions. The results from this experiment indicate that dynamic reserve crew pairing can be utilised to obtain 14.3% more efficient reserve patterns compared to the static solution. This amounts to an additional saving of 5.9 working days per week over the static solutions.

In summary, the GRASP and GRASP-LF methods are both viable methods to create reserve crew patterns that minimise the gap between scheduled reserve capacity and expected reserve demand. Depending on the objectives of the decision maker, either the GRASP method or the GRASP-LF method is most cost efficient. When the reserve budget in unconstrained, the GRASP method yields the best results, but if a minimisation of premium days with a fixed reserve budget is required, then the GRASP-LF method performs best. Additionally, dynamic reserve crew pairing shows potential to further optimise the reserve crew pairing process compared to current practice. Preliminary experiments indicate improvements of 14.3%, but further research should be done to obtain more accurate estimates.

8.2. Research contributions

Conclusions about the contributions of this research towards the scientific body are divided into applicational and methodological novelties.

Problem definition

The airline reserve crew pairing problem has never been considered for long-haul cockpit crew before. Compared to the state of the art, this problem requires the use of reserve pairings of variable length which allow regular flights after reserve duty days. These characteristics are especially relevant to long-range cockpit crew, since (1) these are the most expensive type of human resources for an airline (2) long-haul crew generally operate longer flight pairings of varying lengths compared to short-haul crew.

Simulation optimisation

Reserve pattern optimisation via simulation evaluation is a novel method that is applied in this project. The complex operational characteristics of long-range cockpit crew result in impractically high computational requirements when analytical evaluation models are used. A fast simulation evaluation model allows to measure the quality of reserve patterns in complex operations with acceptable computational requirements.

Development of optimisation algorithms specifically for this problem

The novel LAST, GRASP, and GRASP-LF algorithms were created from existing default optimisation algorithms. Each of the algorithms uses problem specific characteristics to increase its optimisation performance. The LAST algorithm uses the benefit (positive contribution to the objective value) of a reserve pairing to adapt the learning rate for the respective decision variables. Similarly, the GRASP and GRASP-LF algorithms use the potential (a forecast of the amount of premium days prevented) of all reserve pairings to determine which pairings should be added at each construction iteration. Furthermore, the GRASP-LF algorithm is a novel adaption of the GRASP method that mimics the manual way of constructing reserve patterns: by covering the longest flight on each day in the flight schedule.

Comparison of algorithms
A comparison between the novel optimisation algorithms has never been done before. Therefore, the comparison between the optimisation algorithms is also a research contribution. With the pure random search algorithm as a benchmark method, the other algorithms show how alterations of the default random search method can improve the resulting solution quality. Given that the tests compared different objectives, different outcomes for each of these objectives were found.

8.3. Recommendations

For further research, a variety of directions is offered that can be pursued. The recommendations are both aimed at further theoretical research or at the implementation of this project in practice.

Crew scheduling modelling
The reserve crew pairing problem that is considered in this research has been isolated from the overall crew scheduling problem, in order to obtain a workable project scope. Following from this, the model can be used independently from the overall crew scheduling process, given that the required input is defined externally and the generated output is manually processed for further use in the crew scheduling process. An alternative to this approach is to incorporate airline reserve crew pairing in the overall crew scheduling process. For example, the current airline flight schedules can automatically be adapted for each time period that reserve patterns are made. Corresponding flight disruption data can be estimated automatically and adapted when required. The output from the reserve crew pairing process can automatically be included in the assignment process, where pairings are assigned to individual crew. The values for the service level constraints can be automatically adapted based on the social state of the overall crew scheduling model. Through a system of feedback, various parts of the overall model can iteratively be used to obtain balanced and supportive model inputs and outputs.

Disruption timing
The current model assumes that flight disruptions come in the exact order of the flights in the flight schedule. A more accurate representation would be to approximate the disruption timing following representative distributions that vary over time.

Extended model functions
To increase the practical value of the model, a number of additional functions can be implemented which represent operations as encountered in practice. An example is to include a constraint that enforces that specific flights in the flight schedule are always covered by a reserve pairing, or a constraint that enforces a minimum number of pure reserve pairings. Another example is to include other recovery measures in the model, such as type swapping or flight delaying. On top of this, flight cancellations can be introduced when crew readiness towards flying at premium costs cannot be expected.

Bibliography

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

Abdelghany, K. F., Abdelghany, A. F. 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.

Alfares, H. K. (2007), 'Operator staffing and scheduling for an it-help call centre', *European Journal of Industrial Engineering* **1**(4), 414–430.

Amaran, S., Sahinidis, N. V., Sharda, B. and Bury, S. J. (2014), 'Simulation optimization: a review of algorithms and applications', *Annals of Operations Research* **12**(4), 301–333.

Andradóttir, S. and Prudius, A. A. (2009), 'Balanced explorative and exploitative search with estimation for simulation optimization', *INFORMS Journal on Computing* **21**(2), 193–208.

Avramidis, A. N., Chan, W., Gendreau, M., L'ecuyer, P. and Pisacane, O. (2010), 'Optimizing daily agent scheduling in a multiskill call center', *European Journal of Operational Research* **200**(3), 822–832.

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

Barnhart, C. and Cohn, A. (2004), 'Airline schedule planning: Accomplishments and opportunities', *Manufacturing & service operations management* **6**(1), 3–22.

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

Bayliss, C., De Maere, G., Atkin, J. A. 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.

Bayliss, C., De Maere, G., Atkin, J. and Paelinck, M. (2012), Probabilistic airline reserve crew scheduling model, *in* 'OASIcs-OpenAccess Series in Informatics', Vol. 25, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

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, *in* '2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)', IEEE, pp. 1944–1950.

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

Bertsimas, D. and Demir, R. (2002), 'An approximate dynamic programming approach to multidimensional knapsack problems', *Management Science* **48**(4), 550–565.

Bianchi, L., Dorigo, M., Gambardella, L. M. and Gutjahr, W. J. (2009), 'A survey on metaheuristics for stochastic combinatorial optimization', *Natural Computing* **8**(2), 239–287.

Bijvank, M., Byrka, J., van Heijster, P., Gnedin, A., Olejniczak, T., Swist, T., Zyprych, J., Bisseling, R., Mulder, J., Paelinck, M. et al. (2007), 'Cabin crew rostering at KLM: optimization of reserves', *European Study Group Mathematics with Industry* p. 27.

Blum, C., Puchinger, J., Raidl, G., Roli, A. et al. (2010), 'A brief survey on hybrid metaheuristics', *Proceedings of BIOMA* pp. 3–18.

Blum, C. and Roli, A. (2003), ‘Metaheuristics in combinatorial optimization: Overview and conceptual comparison’, *ACM Computing Surveys* **35**(3), 268–308.

Brusco, M. J. and Jacobs, L. W. (1998), ‘Personnel tour scheduling when starting-time restrictions are present’, *Management Science* **44**(4), 534–547.

Burke, E., De Causmaecker, P., Petrovic, S. and Berghe, G. V. (2003), Variable neighborhood search for nurse rostering problems, *in* ‘Metaheuristics: computer decision-making’, Springer, pp. 153–172.

Campbell, G. M. (2012), ‘On-call overtime for service workforce scheduling when demand is uncertain’, *Decision Sciences* **43**(5), 817–850.

Cezik, M. T. and L’Ecuyer, P. (2008), ‘Staffing multiskill call centers via linear programming and simulation’, *Management Science* **54**(2), 310–323.

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

Council of European Union (2014), ‘Council regulation (EU) no 83/2014’. `https://www.easa.europa.eu/document-library/regulations/commission-regulation-eu-no-832014#group-easa-downloads`.

Davis, S. G. and Reutzel, E. T. (1981), ‘A dynamic programming approach to work force scheduling with time-dependent performance measures’, *Journal of Operations Management* **1**(3), 165–171.

Dillon, J. E. and Kontogiorgis, S. (1999), ‘US Airways optimizes the scheduling of reserve flight crews’, *Interfaces* **29**(5), 123–131.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), ‘Ant system: optimization by a colony of cooperating agents’, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **26**(1), 29–41.

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. F. (2011), ‘Cross-training performance in flexible labor scheduling environments’, *IIE Transactions* **43**(8), 589–603.

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

Easton, F. F. and Goodale, J. C. (2005), ‘Schedule recovery: Unplanned absences in service operations’, *Decision Sciences* **36**(3), 459–488.

Easton, F. F. and Mansour, N. (1999), ‘A distributed genetic algorithm for deterministic and stochastic labor scheduling problems’, *European Journal of Operational Research* **118**(3), 505–523.

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

Elshafei, M. and Alfares, H. K. (2008), ‘A dynamic programming algorithm for days-off scheduling with sequence dependent labor costs’, *Journal of Scheduling* **11**(2), 85–93.

Ertogral, K. and Bamuqabel, B. (2008), ‘Developing staff schedules for a bilingual telecommunication call center with flexible workers’, *Computers & Industrial Engineering* **54**(1), 118–127.

Feo, T. A. and Resende, M. G. (1995), ‘Greedy randomized adaptive search procedures’, *Journal of global optimization* **6**(2), 109–133.

Gaballa, A. (1979), ‘Planning callout reserves for aircraft delays’, *Interfaces* **9**(2), 78–86.

Gao, C., Johnson, E. and Smith, B. (2009), 'Integrated airline fleet and crew robust planning', *Transportation Science* **43**(1), 2–16.

Glover, F. (1989), 'Tabu search—part I', *ORSA Journal on computing* **1**(3), 190–206.

Glover, F. and Laguna, M. (2013), Tabu search, *in* 'Handbook of Combinatorial Optimization', Springer, pp. 3261–3362.

Gnanlet, A. and Gilland, W. G. (2009), 'Sequential and simultaneous decision making for optimizing health care resource flexibilities', *Decision Sciences* **40**(2), 295–326.

Gong, W.-B., Ho, Y.-C. and Zhai, W. (2000), 'Stochastic comparison algorithm for discrete optimization with estimation', *SIAM Journal on Optimization* **10**(2), 384–404.

Gosavi, A. (2015), 'Simulation-based optimization', *parametric optimization techniques and reinforcement learning* .

Gurvich, I., Luedtke, J. and Tezcan, T. (2010), 'Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach', *Management Science* **56**(7), 1093–1115.

Hillier, F. S. (2012), *Introduction to operations research*, Tata McGraw-Hill Education.

Hong, L. J. and Nelson, B. L. (2006), 'Discrete optimization via simulation using compass', *Operations Research* **54**(1), 115–129.

Hu, J., Fu, M. C. and Marcus, S. I. (2007), 'A model reference adaptive search method for global optimization', *Operations Research* **55**(3), 549–568.

Ingolfsson, A., Campello, F., Wu, X. and Cabral, E. (2010), 'Combining integer programming and the randomization method to schedule employees', *European Journal of Operational Research* **202**(1), 153–163.

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

Jönsson, H. (1987), 'Dimensioning of bus driver buffers subject to variations in the traffic load', *Engineering Costs and Production Economics* **12**(1-4), 29–38.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. et al. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.

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.

Koole, G. and Pot, A. (2005), Approximate dynamic programming in multi-skill call centers, *in* 'Proceedings of the 37th conference on winter simulation', Winter Simulation Conference, pp. 576–583.

Le, M.-l. and Wu, C.-c. (2013), 'Solving airlines disruption by considering aircraft and crew recovery simultaneously', *Journal of Shanghai Jiaotong University (Science)* **18**(2), 243–252.

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

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

Maher, S. J. (2015*b*), 'Solving the integrated airline recovery problem using column-and-row generation', *Transportation Science* **50**(1), 216–239.

Martí, R. and Reinelt, G. (2011), *The linear ordering problem: exact and heuristic methods in combinatorial optimization*, Vol. 175, Springer Science & Business Media.

Mladenović, N. and Hansen, P. (1997), 'Variable neighborhood search', *Computers & operations research* **24**(11), 1097–1100.

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

Paelinck, M. (2001), KLM cabin crew reserve duty optimisation, *in* ‘Agifors proceedings’.

Parisio, A. and Jones, C. N. (2015), ‘A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand’, *Omega* **53**, 97–103.

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

Pinker, E. J. and Larson, R. C. (2003), ‘Optimizing the use of contingent labor when demand is uncertain’, *European Journal of Operational Research* **144**(1), 39–55.

Pot, A., Bhulai, S. and Koole, G. (2008), ‘A simple staffing method for multiskill call centers’, *Manufacturing & Service Operations Management* **10**(3), 421–428.

Powell, W. B. (2009), ‘What you should know about approximate dynamic programming’, *Naval Research Logistics (NRL)* **56**(3), 239–249.

Schaefer, A. J., Johnson, E. L., Kleywegt, A. J. and Nemhauser, G. L. (2005), ‘Airline crew scheduling under uncertainty’, *Transportation science* **39**(3), 340–348.

Shapiro, A., Dentcheva, D. and Ruszczyński, A. (2009), *Lectures on stochastic programming: modeling and theory*, SIAM.

Shapiro, A. and Philpott, A. (2007), ‘A tutorial on stochastic programming’, *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html* **17**.

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

Shi, L. and Ólafsson, S. (2000), ‘Nested partitions method for global optimization’, *Operations research* **48**(3), 390–407.

Shi, L. et al. (2000), ‘Nested partitions method for stochastic optimization’, *Methodology and Computing in Applied probability* **2**(3), 271–291.

Sohoni, M. G., Johnson, E. L. and Bailey, T. G. (2004), ‘Long-range reserve crew manpower planning’, *Management Science* **50**(6), 724–739.

Sohoni, M. G., Johnson, E. L. and Bailey, T. G. (2006), ‘Operational airline reserve crew planning’, *Journal of Scheduling* **9**(3), 203–221.

Soykan, B. and Erol, S. (2014), ‘A branch-and-price algorithm for the robust airline crew pairing problem’, *Journal of Defense Sciences* **13**(1), 37–74.

Stojković, M. (2005), ‘The operational flight and multi-crew scheduling problem’, *Yugoslav Journal of Operations Research* **15**(1), 25–48.

Stojković, M. and Soumis, F. (2001), ‘An optimization model for the simultaneous operational flight and pilot scheduling problem’, *Management Science* **47**(9), 1290–1305.

Stojković, M., Soumis, F. and Desrosiers, J. (1998), ‘The operational airline crew scheduling problem’, *Transportation Science* **32**(3), 232–245.

Thathachar, M. A. and Sastry, P. S. (1987), ‘Learning optimal discriminant functions through a cooperative game of automata’, *IEEE Transactions on Systems, Man, and Cybernetics* **17**(1), 73–85.

Thompson, G. M. and Goodale, J. C. (2006), ‘Variable employee productivity in workforce scheduling’, *European Journal of Operational Research* **170**(2), 376–390.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E. and De Boeck, L. (2013), ‘Personnel scheduling: A literature review’, *European Journal of Operational Research* **226**(3), 367–385.

Vassilacopoulos, G. (1985), ‘Allocating doctors to shifts in an accident and emergency department’, *Journal of the Operational Research Society* **36**(6), 517–523.

Wei, G., Yu, G. and Song, M. (1997), ‘Optimization model and algorithm for crew management during airline irregular operations’, *Journal of Combinatorial Optimization* **1**(3), 305–321.

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

Xu, J., Nelson, B. L. and Hong, J. (2010), ‘Industrial strength compass: A comprehensive algorithm and software for optimization via simulation’, *ACM Transactions on Modeling and Computer Simulation* **20**(1), 3.

Xu, J., Nelson, B. L. and Hong, L. J. (2013), ‘An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems’, *INFORMS Journal on Computing* **25**(1), 133–146.

Xu, Y., Wang, L., Wang, S.-y. and Liu, M. (2015), ‘An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time’, *Neurocomputing* **148**, 260–268.

Yan, S., Chen, C.-H. and Chen, C.-K. (2008), ‘Short-term shift setting and manpower supplying under stochastic demands for air cargo terminals’, *Transportation* **35**(3), 425–444.

Yang, S., Jiang, Y. and Nguyen, T. T. (2012), ‘Metaheuristics for dynamic combinatorial optimization problems’, *IMA Journal of Management Mathematics* **24**(4), 451–480.

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

Yu, G., Argüello, M., Song, G., McCowan, S. M. and White, A. (2003), ‘A new era for crew recovery at Continental Airlines’, *Interfaces* **33**(1), 5–22.

Zhang, D., Lau, H. H. and Yu, C. (2015), ‘A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems’, *Computers & Industrial Engineering* **87**, 436–453.

Zhu, X. and Sherali, H. D. (2009), ‘Two-stage workforce planning under demand fluctuations and uncertainty’, *Journal of the Operational Research Society* **60**(1), 94–103.

A
Problem scenario

Table A.1: Problem scenario flight schedule. All times expressed in fraction of days. Destinations have been stripped from the flight schedule for confidentiality reasons.

Flight ID	Dest	Dep time	Disrupt prob	Route days	Rest days	Planned FDP	Max FDP	Buffer period	Premium weight
1		0.38	0.048	4	3	0.497	0.667	0.333	1.411
2		0.38	0.038	4	3	0.399	0.667	0.250	1.286
3		0.4	0.055	4	3	0.399	0.542	0.250	1.500
4		0.41	0.038	3	3	0.417	0.667	0.333	1.286
5		0.47	0.015	3	4	0.427	0.542	0.250	1.000
6		0.47	0.026	4	3	0.566	0.667	0.333	1.140
7		0.5	0.047	4	3	0.403	0.490	0.250	1.392
8		0.53	0.043	3	2	0.333	0.542	0.250	1.348
9		0.53	0.031	3	3	0.486	0.521	0.250	1.198
10		0.56	0.050	3	4	0.427	0.542	0.250	1.431
11		0.59	0.022	3	4	0.375	0.521	0.250	1.091
12		1.35	0.026	4	3	0.333	0.542	0.250	1.136
13		1.4	0.052	4	3	0.399	0.542	0.250	1.460
14		1.41	0.038	4	3	0.417	0.667	0.333	1.286
15		1.42	0.055	3	2	0.337	0.542	0.250	1.498
16		1.47	0.015	4	4	0.427	0.542	0.250	1.000
17		1.47	0.026	4	3	0.566	0.667	0.333	1.140
18		1.53	0.043	3	2	0.333	0.542	0.250	1.348
19		1.54	0.031	3	3	0.486	0.521	0.250	1.198
20		1.56	0.050	4	4	0.427	0.542	0.250	1.431
21		1.59	0.022	3	4	0.375	0.521	0.250	1.091
22		1.9	0.031	8	3	0.406	0.458	0.250	2.000
23		2.38	0.048	3	4	0.497	0.667	0.333	1.411
24		2.38	0.038	3	4	0.399	0.667	0.333	1.286
25		2.4	0.055	4	3	0.399	0.542	0.250	1.500
26		2.42	0.055	3	2	0.337	0.542	0.250	1.498
27		2.47	0.026	4	3	0.566	0.667	0.333	1.140
28		2.5	0.047	4	3	0.403	0.490	0.250	1.392
29		2.53	0.043	3	2	0.333	0.542	0.250	1.348
30		2.54	0.031	3	3	0.486	0.521	0.250	1.198
31		2.56	0.038	4	2	0.434	0.542	0.250	1.286
32		2.59	0.022	3	4	0.375	0.521	0.250	1.091
33		3.35	0.026	4	3	0.333	0.542	0.250	1.136
34		3.38	0.048	4	3	0.497	0.667	0.333	1.411

Flight ID	Dest	Dep time	Disrupt prob	Route days	Rest days	Planned FDP	Max FDP	Buffer period	Premium weight
35		3.38	0.038	3	4	0.399	0.667	0.333	1.286
36		3.4	0.052	4	3	0.399	0.542	0.250	1.460
37		3.41	0.038	4	3	0.417	0.667	0.333	1.286
38		3.42	0.055	3	2	0.337	0.542	0.250	1.498
39		3.47	0.015	4	4	0.427	0.542	0.250	1.000
40		3.47	0.026	4	3	0.566	0.667	0.333	1.140
41		3.53	0.043	3	2	0.333	0.542	0.250	1.348
42		3.54	0.031	3	3	0.486	0.521	0.250	1.198
43		3.56	0.050	4	4	0.427	0.542	0.250	1.431
44		3.59	0.022	3	4	0.375	0.521	0.250	1.091
45		4.38	0.038	4	3	0.399	0.667	0.333	1.286
46		4.4	0.052	4	3	0.399	0.542	0.250	1.460
47		4.42	0.055	4	2	0.337	0.542	0.250	1.498
48		4.47	0.026	4	3	0.566	0.667	0.333	1.140
49		4.47	0.049	4	3	0.566	0.708	0.333	1.421
50		4.5	0.047	4	3	0.403	0.490	0.250	1.392
51		4.53	0.043	3	2	0.333	0.542	0.250	1.348
52		4.54	0.031	3	3	0.486	0.521	0.250	1.198
53		4.56	0.038	4	2	0.434	0.542	0.250	1.286
54		4.59	0.022	3	4	0.375	0.521	0.250	1.091
55		4.9	0.031	7	3	0.406	0.458	0.250	2.000
56		5.35	0.026	5	3	0.333	0.542	0.250	1.136
57		5.38	0.048	3	4	0.497	0.667	0.333	1.411
58		5.4	0.052	4	3	0.399	0.542	0.250	1.460
59		5.41	0.038	3	3	0.417	0.667	0.333	1.286
60		5.47	0.015	4	4	0.427	0.542	0.250	1.000
61		5.47	0.026	4	3	0.566	0.667	0.333	1.140
62		5.53	0.043	3	2	0.333	0.542	0.250	1.348
63		5.54	0.031	4	2	0.486	0.521	0.250	1.198
64		5.56	0.050	4	4	0.427	0.542	0.250	1.431
65		5.59	0.022	3	4	0.375	0.521	0.250	1.091
66		6.38	0.048	3	4	0.497	0.667	0.333	1.411
67		6.38	0.038	3	4	0.399	0.667	0.333	1.286
68		6.4	0.052	4	3	0.399	0.542	0.250	1.460
69		6.41	0.038	3	3	0.417	0.667	0.333	1.286
70		6.42	0.055	4	2	0.337	0.542	0.250	1.498
71		6.47	0.026	4	3	0.566	0.667	0.333	1.140
72		6.42	0.049	7	4	0.566	0.708	0.333	1.421
73		6.5	0.047	3	4	0.403	0.490	0.250	1.392
74		6.53	0.043	3	2	0.333	0.542	0.250	1.348
75		6.54	0.031	4	2	0.486	0.521	0.250	1.198
76		6.56	0.038	5	2	0.434	0.542	0.250	1.286
77		6.59	0.022	3	4	0.375	0.521	0.250	1.091
78		6.9	0.031	8	3	0.406	0.458	0.250	2.000

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 07:00	RD	RD	RD										
ID 2		RT 07:00	RD	RD	RD									
ID 3		RT 16:00	RT 10:59	SB	SB	RD	RD	RD	RD					
ID 4			RT 07:00	RT 07:00	SB	SB								
ID 5			RT 07:00	RD	RD	RD								
ID 6				RT 07:00	RT 07:00	SB	SB							
ID 7				RT 07:00	RD	RD	RD							
ID 8					RT 07:00	RT 07:00	SB	SB						
ID 9					RT 16:00	RT 10:59	RD	RD	RD	RD	RD			
ID 10						RT 07:00	RT 07:00	RD	RD	RD				
ID 11						RT 07:00	RT 07:00	SB	SB					
ID 12							RT 07:00	RD	RD	RD	RD			
ID 13							RT 16:00	RT 10:59	SB	SB	RD	RD	RD	RD

Figure A.1: Manually constructed reserve pattern for the problem scenario.

B

Model settings

Table B.1: Model settings for the evaluation model.

Setting	Value
Use policy	Minimum waste days
Assign policy	Equal probability
N	25000
Day costs	1.0

Table B.2: Model settings for the random search and LAST optimisation methods.

Setting	Value
Iterations	4000
N	1500
Initial density	0.05
Learning policy	min unused + premium
Random fraction	3
Learning rate	0.01

Table B.3: Model settings for the GRASP and GRASP-LF methods.

Setting	Value
Repeats	4
Population size	5
Candidates multiplier	2.5
N	1500
Selection policy	min $0.75 \cdot$ unused $+ 0.25 \cdot$ premium

C
Test results

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 07:00	RT 07:00	SB	SB										
ID 2	RT 16:00	RT 07:00	SB	SB										
ID 3		RT 07:00	RT 11:00	RD	RD	RD	RD							
ID 4		RT 07:00	RT 07:00	SB	RD	RD	RD	RD						
ID 5		RT 11:00	RT 11:00	RD	RD	RD								
ID 6			RT 07:00	RT 07:00	SB	SB								
ID 7				RT 07:00	RT 16:00	SB	SB							
ID 8					RT 07:00	RT 11:00	SB	SB						
ID 9					RT 11:00	RT 11:00	SB	RD	RD	RD				
ID 10							RT 07:00	RT 07:00	RD	RD	RD			
ID 11							RT 11:00	RT 16:00	RD	RD	RD			

Figure C.1: Reserve pattern for test 1 for the random search method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 11:00	RT 11:00	SB	SB										
ID 2		RT 07:00	RT 07:00	RD	RD	RD	RD							
ID 3		RT 11:00	RT 16:00	RD	RD	RD								
ID 4			RT 07:00	RT 07:00	SB	SB								
ID 5				RT 07:00	RT 11:00	SB	SB							
ID 6				RT 11:00	RD	RD	RD							
ID 7					RT 07:00	RT 16:00	RD	RD	RD					
ID 8					RT 07:00	RT 07:00	SB	SB	RD	RD	RD			
ID 9							RT 07:00	RT 07:00	SB	SB	SB			
ID 10							RT 07:00	RT 07:00	SB	RD	RD	RD	RD	

Figure C.2: Reserve pattern for test 1 for the LAST method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 11:00	RT 07:00	SB	SB										
ID 2		RT 07:00	RT 07:00	RD	RD	RD								
ID 3		RT 07:00	RT 07:00	RD	RD	RD								
ID 4			RT 07:00	RT 07:00	SB	SB								
ID 5				RT 07:00	RD	RD	RD							
ID 6				RT 11:00	RT 07:00	SB	SB							
ID 7					RT 07:00	RD	RD	RD						
ID 8					RT 16:00	RT 07:00	SB	SB						
ID 9						RT 07:00	RT 07:00	RD	RD	RD				
ID 10							RT 07:00	RT 07:00	RD	RD	RD	RD		

Figure C.3: Reserve pattern for test 1 for the GRASP method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 07:00	RD	RD	RD										
ID 2		RT 07:00	RT 11:00	SB	SB									
ID 3		RT 07:00	RT 07:00	SB	SB	SB								
ID 4		RT 07:00	RD	RD	RD									
ID 5		RT 16:00	RT 07:00	SB	SB	RD	RD	RD	RD					
ID 6				RT 07:00	RD	RD	RD							
ID 7					RT 07:00	RT 07:00	RD	RD	RD					
ID 8					RT 07:00	RT 07:00	RD	RD	RD	RD	RD			
ID 9						RT 07:00	RT 07:00	SB	SB	SB				
ID 10							RT 07:00	RT 07:00	RD	RD	RD			
ID 11							RT 07:00	RT 07:00	RD	RD	RD			
ID 12							RT 16:00	RT 11:00	SB	SB	RD	RD	RD	RD

Figure C.4: Reserve pattern for test 1 for the GRASP-LF.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1		RT 07:00	RT 07:00	SB	SB									
ID 2		RT 07:00	RD	RD	RD	RD								
ID 3		RT 16:00	RT 07:00	SB	RD	RD	RD							
ID 4		RT 16:00	RT 07:00	SB	RD	RD	RD	RD						
ID 5			RT 07:00	RT 16:00	SB	SB								
ID 6				RT 07:00	RD	RD	RD							
ID 7					RT 07:00	RT 07:00	SB	SB	RD	RD	RD			
ID 8					RT 11:00	RT 16:00	RD	RD	RD					
ID 9							RT 07:00	RT 07:00	SB	SB	SB			
ID 10							RT 07:00	RD	RD	RD				
ID 11							RT 07:00	RT 11:00	SB	RD	RD	RD		
ID 12							RT 11:00	RT 07:00	RD	RD	RD			
ID 13							RT 11:00	RT 07:00	RD	RD	RD	RD		
ID 14							RT 11:00	RT 11:00	RD	RD	RD	RD		

Figure C.5: Reserve pattern for test 2 for the random search method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 07:00	RT 16:00	SB	SB										
ID 2	RT 07:00	RD	RD	RD										
ID 3		RT 07:00	RT 11:00	SB	RD	RD	RD							
ID 4		RT 07:00	RT 07:00	RD	RD	RD	RD							
ID 5				RT 07:00	RT 07:00	SB	SB							
ID 6				RT 07:00	RT 11:00	SB	SB							
ID 7					RT 07:00	RT 07:00	SB	RD	RD	RD	RD			
ID 8					RT 11:00	RD	RD	RD	RD					
ID 9						RT 07:00	RT 16:00	RD	RD	RD				
ID 10							RT 07:00	RT 07:00	SB	SB	SB			
ID 11							RT 07:00	RT 07:00	SB	SB	RD	RD	RD	
ID 12							RT 11:00	RD	RD	RD	RD			

Figure C.6: Reserve pattern for test 2 for the LAST method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1	RT 07:00	RD	RD	RD										
ID 2		RT 07:00	RT 07:00	SB	SB	SB								
ID 3		RT 07:00	RD	RD	RD									
ID 4		RT 07:00	RT 11:00	RD	RD	RD								
ID 5		RT 11:00	RT 07:00	SB	SB									
ID 6			RT 07:00	RD	RD	RD								
ID 7			RT 07:00	RD	RD	RD								
ID 8				RT 07:00	RD	RD	RD							
ID 9				RT 07:00	RD	RD	RD							
ID 10				RT 11:00	RT 11:00	SB	SB							
ID 11					RT 07:00	RT 07:00	RD	RD	RD					
ID 12					RT 07:00	RT 07:00	RD	RD	RD	RD				
ID 13					RT 16:00	RT 07:00	SB	RD	RD	RD	RD			
ID 14						RT 07:00	RT 07:00	RD	RD	RD				
ID 15							RT 07:00	RT 07:00	RD	RD	RD			
ID 16							RT 07:00	RT 07:00	RD	RD	RD			
ID 17							RT 07:00	RT 07:00	SB	RD	RD	RD	RD	

Figure C.7: Reserve pattern for test 2 for the GRASP method.

	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
ID 1		RT 07:00	RT 07:00	RD	RD	RD								
ID 2		RT 07:00	RT 11:00	RD	RD	RD								
ID 3		RT 16:00	RT 07:00	SB	SB	RD	RD	RD	RD					
ID 4				RT 07:00	RT 11:00	SB	SB							
ID 5				RT 07:00	RT 11:00	SB	SB							
ID 6				RT 07:00	RD	RD	RD							
ID 7					RT 07:00	RT 07:00	RD	RD	RD					
ID 8					RT 07:00	RT 07:00	SB	RD	RD	RD				
ID 9					RT 16:00	RT 07:00	RD	RD	RD	RD	RD			
ID 10							RT 07:00	RT 07:00	SB	SB				
ID 11							RT 07:00	RT 07:00	SB	RD	RD	RD	RD	
ID 12							RT 11:00	RT 07:00	RD	RD	RD			
ID 13							RT 16:00	RT 07:00	SB	SB	RD	RD	RD	RD

Figure C.8: Reserve pattern for test 2 for the GRASP-LF.