Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization: Transport Engineering and Logistics

Report number: 2016.TEL.8034

Title: **A decision support system for airline disruption management at KLM Cityhopper**

Author: K. Aslan

Title (in Dutch) Een beslissingsondersteunend systeem voor het oplossen van operationele verstoringen bij KLM Cityhopper

Assignment: Master's thesis

Confidential: yes (until Jul 2021)

Initiator (university): prof. dr. ir. G. Lodewijks

Initiator (company): J.A.C. van der Veer (KLM Cityhopper, Schiphol)

Supervisor (university): dr. ir. H. P. M. Veeke

Supervisor (company): M.C.E. Paelinck (KLM, Amstelveen)

Date: August 29, 2016

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	K. Aslan	Assignment type:	Master thesis
Supervisor (TUD):	Dr. ir. H.P.M. Veeke	Creditpoints (EC):	35
Initiator (Company):	J.A.C. van der Veer	Specialization:	TEL
Supervisor (Company)	M.C.E. Paelinck	Report number:	2016.TEL.8034
Professor	Prof. Dr. ir. G Lodewijks	Confidential:	Yes, until Jul 2021

Subject: **A decision support system for airline disruption management at KLM Cityhopper**

Introduction

Airline operations have grown tremendously complex over the years. This has resulted in increasingly tighter schedules with more resource utilization to maximise profits and minimise costs. However, increased resource utilisation makes the carefully planned fleet and crew schedules more susceptible to unforeseen events such as bad weather or technical failures. Some events may even propagate through an entire network.

To manage operations, KLC has an entity named Operations Control where human experts control the safety of operations and deal with complications that have a negative effect on the flight schedule. Operators can choose from different actions to mitigate the impact of a disruption on the flight schedule. Such actions include swapping resources, delaying flights until other resources become available or even cancelling flights.

Problem definition

Currently, there is little insight in the decision-making process. Operators use different information systems to assess the necessary steps to mitigate a problem. However, it is difficult to evaluate if all options have been analyzed, and if all the required information is consulted. Furthermore, the required information may not even be readily available or accessible.

Therefore, there is need to support the operator during the decision-making process.

Research goal

Design a decision support system that provides insight on the quality of possible solutions to a disruption by informing on the consequences thereof.

Execution

- Analyze the processes of airline disruption management (ADM) at KLM Cityhopper according to the Delft System Approach
- Analyze the control process at ADM according to which one reacts to occurring disruptions.
- Define performance criteria for a decision support system
- Develop a prototype of a decision support system for operational disruptions occurring
- Verify the tool and validate as far as possible
- Study relevant literature

b.a

Prof. dr. ir. G. Lodewijks

Summary

KLM Cityhopper (KLC) is a regional subsidiary of KLM Royal Dutch Airlines. It functions as an operator for KLM and conducts a large part of its European network. Depending on the season, it operates around 250 flights a day to around 55 destinations. KLC does not sell any tickets for these flights. Instead, its financial basis is based on the number of performed flight legs.

Each season, KLM places an order with KLC in the form of a timetable. KLC is then responsible for the execution of each flight leg on this schedule. To do so, KLC manages its own fleet that exists out of three aircraft types with a total of 47 aircraft, and its own flight crew, which is a total of about 1000 FTEs. These resources are carefully planned to maximise utilisation whilst keeping cost as low as possible. However, during the day of operation the carefully constructed flight and crew schedules can become infeasible due to unforeseen events, such as technical failures, bad weather or late incoming passengers. Such events typically disrupt passenger itineraries and lead to one or more resources to become unavailable, requiring an alternative to continue planned operations. Thereto, KLC has an entity named Operations Control where human experts monitor and coordinate flight operations and deal with complications that adversely impact the planned schedules to ensure each passenger arrives to their destination in a safe and sound manner. In the wake of disruptions, KLC operations controllers can initiate a recovery process for the so-called airline recovery problem in the domains of fleet or crew assignment. Given an initial situation and a list of known disruptions, the recovery processes consists out of reallocation of aircraft or crew members to resume regular operations and minimise impact on passengers as far as possible. Operations controllers can swap, delay or cancel flights to make aircraft or crew members available again and resume operations.

Currently, the recovery process is a manual process where operation controllers rely on limited information, wit and experience to achieve the best possible result. There is little to no transparency in the decision-making process. After careful analysis it is concluded that this is because not all the required information is readily available or accessible. To address this problem the following research goal is formulated: *"Design a decision support system that provides insight on the quality of possible solutions to a disruption by informing on the consequences thereof."*

To achieve this, first design requirements and performance criteria for the decision support system (DSS) are selected. To keep the airline recovery problem manageable it is decided to first address the aircraft recovery domain. A literature survey is conducted to analyse the state of the art for the airline recovery process. From the literature survey, a metaheuristic based on a tabu search is found to be able to deliver the desired results for the airline recovery process. This metaheuristic is then adapted with some minor but functional improvements and is implemented into a software package that is able to deliver the required solutions. Computational tests show that the algorithm is able to find a pool of solution to problems that are realistic and acceptable in a reasonable amount of time. To help the operations controllers to identify the best course of actions, solutions are then presented using

characteristics that provide insight in the actions needed to achieve that outcome and consequences thereof. Finally, recommendations are made to further improve upon the DSS and algorithm.

Samenvatting

KLM Cityhopper (KLC) is een 100% dochteronderneming van KLM. KLC fungeert als een operator voor KLM en realiseert een groot deel van het Europees netwerk van KLM. Afhankelijk van het seizoen voert KLC ongeveer 250 vluchten per dag uit op 55 bestemmingen. KLC verkoopt zelf geen tickets voor deze vluchten. In plaats hiervan berust KLC's financiële basis op het realiseren van de geplande vluchten.

Elk seizoen plaats KLM een bestelling bij KLC in de vorm van een dienstregeling. Het is dan KLC's verantwoordelijkheid om de vluchten op deze dienstregeling te realiseren. Hiervoor beschikt KLC over haar eigen vloot, welk bestaat uit 47 vliegtuigen, en haar eigen vliegpersoneel, welk bestaat uit ongeveer 1000 FTE. Deze middelen worden zo ingepland dat zij maximaal kunnen ingezet worden tegen minimale kosten. Desondanks kunnen de zorgvuldig opgestelde schema's en roosters verstoord worden door onvoorziene voorvallen zoals technische defecten, slecht weer of te late passagiers. Zulke gebeurtenissen hebben als gevolg dat de reisplannen van (andere) passagiers verstoord raken en dat een of meer type middelen niet beschikbaar raken; met als gevolg dat een alternatief gevonden moet worden om aan de planning te kunnen voldoen. Om operationele verstoringen te kunnen opvangen heeft KLC een orgaan genaamd Operations Control waar de besturing plaats vindt. Hier zitten experts die de operatie bewaken, coördineren en operationele verstoringen afhandelen om te zorgen dat de passagiers zo min mogelijk hinder ondervinden. Gedurende verstoringen kunnen de KLC Operation Controllers een herstel proces aangaan in de domeinen van vloot en bemanning. In dit proces is het de bedoeling om in een bepaalde tijd vanuit een gegeven initiële verstoorde situatie terug te keren naar de originele planning en om de impact op de passagiers zo min mogelijk te houden. Hiervoor kunnen Operation Controllers vluchten wisselen van vliegtuig, doorvertragen of annuleren van vluchten om zo vliegtuigen of bemanningspersoneel beschikbaar te stellen en de geplande operatie te kunnen hervatten.

Momenteel werkt men dit herstelproces manueel uit waarbij het Operations Control personeel is toegewezen tot hen ervaringen, inzichten en de beschikbare informatie. Er is weinig tot geen inzicht in het huidige beslissingsproces. Na een zorgvuldige analyse is gebleken dat dit komt doordat de informatie niet of beperkt toegankelijk of beschikbaar is. Voor dit probleem is onderzoeksdoel voorgenomen om een "*Beslissingsondersteunend systeem te ontwikkelen dat inzicht biedt in de kwaliteiten en consequenties van de mogelijke oplossingen tot een opgetreden probleem"*.

Hiervoor worden er eerst een aantal design criteria opgesteld. Om het probleem behapbaar te houden wordt de keuze gemaakt om in deze stadium eerst een prototype te ontwikkelen voor de *aircraft recovery problem*. Vervolgens wordt er een literatuuronderzoek gedaan om te onderzoeken wat er al geproduceerd is op dit vlak. Vanuit dit onderzoek wordt een metaheuristiek gebaseerd op de *tabu search* geselecteerd die in staat is om de gewenste resultaat te leveren. Deze metaheuristiek wordt dan met wat aanpassingen en verbeteringen geïmplementeerd tot een computermodel welk in staat is de gewenste oplossingen te vinden. Tests tonen aan dat het ontwikkelde algoritme in staat is om in een redelijke tijd een scala aan oplossingen te vinden die realistisch en acceptabel zijn. Om de gebruiker te

helpen bij het vinden van de juiste handelswijze worden de oplossingen gepresenteerd aan de hand van een aantal karakteristieken die inzicht bieden in de handelingen die nodig zijn en ook de gevolgen daarvan. Tot slot worden een aantal aanbevelingen gedaan om het opgeleverde model en algoritme te verbeteren en tot verdere onderzoek.

Voorwoord

Voor u ligt het verslag mijn afstudeeronderzoek dat is geschreven als afronding van de master Transport Engineering and Logistics aan de Technische Universiteit Delft.

Toen ik in september 2008 begon aan de TU Delft had ik nooit gedacht dat ik mijn studieperiode zou afsluiten bij de KLM – iets met wat mijn voorliefde voor reizen altijd toch een soort droom is geweest. Na een jaar mee te hebben gedraaid kan ik niets anders concluderen dan dat het een geweldige ervaring is geweest! Ik heb er ontzettend van genoten om naast mijn afstudeeronderzoek ook aan andere projecten binnen het bedrijf een steentje te kunnen bijdragen. Het afronden van dit afstudeeronderzoek is soms samengegaan met wat frustratie, maar ik geloof dat het wel geleid heeft tot een mooi eindproduct. Stiekem vind ik het wel een beetje lastig om afstand te doen van wat is opgeleverd, maar aan de andere kant vind ik ook wel dat het tijd is geworden voor wat nieuws. Hoe dan ook, ik heb er met veel plezier aan gewerkt en ik hoop dat het straks ook echt een bijdrage kan leveren aan de bedrijfsvoering.

Deze gelegenheid wil ik dan ook gebruiken om een aantal mensen te bedanken. Ten eerste wil ik Arjan Willems bedanken voor het onderwerp, anders was dit onderzoek nooit tot stand gekomen. Tevens bedankt voor de constructieve feedback, het beantwoorden van eindeloze stroom van vragen en het delen van je expertise als manager Operations Control. Jan van der Veer wil ik hartelijk bedanken voor de uitstekende project management skills, de constructieve feedback en een gezonde dosis positiviteit en humor. Wanneer ik trachtte af te dwalen heb je mij altijd goed kunnen bijsturen. Marc Paelinck wil ik bedanken voor het begeleiden en meedenken in het modelleer en programmeerwerk. Anders was ik waarschijnlijk nog steeds aan het debuggen…. Many thanks to Filzah for your endless support. The experience wouldn't have been as enjoyable without you by my side. I could always vent and discuss subjects with you in which you have no knowledge in, but it always helped me to understand things better myself. Ik had mij vanuit het bedrijf in ieder geval geen beter begeleiding kunnen voorstellen.

Daarnaast gaat mijn dank natuurlijk ook uit naar Hans Veeke voor de kritiek, het brainstormen en richting geven aan mijn onderzoek. De enigszins regelmatige bijeenkomsten hebben mij geholpen om vaak een stap terug te nemen en het grotere plaatje niet uit het oog te verliezen. Ook wil ik Gabriel Lodewijks bedanken voor de kritische maar zeker ook opbouwende feedback tijdens de bijeenkomsten die we hebben gehad.

Zonder alle steun vanuit KLM, TU Delft en familie en vrienden zou dit resultaat niet voor u liggen.

List of abbreviations

AOG	Aircraft on ground
CC	Crew Control
DMPS	Duty Manager Passenger Services
DMOC	Duty Manager Operations Control
DoO	Day of operation
DSA	Delft Systems Approach
DSS	Decision support system
FC	Fleet controller
FS	Fleet scheduler
FTE	Full time equivalent
GUI	Graphical user interface
jr. OC	Junior Operation Controller
KLC	KLM Cityhopper
OC	Operations Control
OCC	Operations Control Center
OP&C	Operations Planning & Control
TFM	Traffic flow manager

Table of contents

1 Introduction

Over the past few decades, the airline industry has grown tremendously. Accordingly, airline operations have grown to be more complex with increasingly tighter schedules and more resource utilization than ever before. However, increased resource utilisation makes the carefully planned flight and crew schedules more susceptible to unforeseen events such as bad weather, technical failures or ill crew members. Some events may even propagate through an entire network.

Disruptions typically lead to resource restrictions in aircraft, crew members or airport facilities that can cause delays or flight cancellations, hereby causing inconvenience for the passenger and increasing expenses for the airline. It is estimated that irregularities can cost as much as 3% of an airline's annual revenue and that a better recovery process can lead to at least 20% savings therein [1]. Thus, for airlines, it is cost-effective to maintain on-time performance by reducing the adverse impact of disruptions on the flight schedule as much as possible. Evidently, because of the nature of many disruptions, it is impossible to eliminate them. While eliminating is impossible, effective recovery programs can mitigate the impact and improve the operational efficiency of the airline.

To manage flight operations, airlines have an entity named operations control (OC) where human experts control the safety of operations and deal with complications that have a negative effect on the flight schedule [2]. Operations controllers can choose from different actions to minimize the impact of a disruption on the flight schedule, such as delaying flights until resources become available or using reserve resources to manage a disruption. It is crucial that operations controllers have all the relevant and necessary information for effective decision-making during recovery programs. Information such as the immediate effect and subsequent impact of a disruption on operations, the time that is required to solve one, the relevant restrictions that are applicable to the situation and the best possible actions that can reduce the impact of a disruption need to be readily available.

However, too often it is difficult if not impossible for the operations controller to collect all the necessary information in a timely manner. Furthermore, with larger disruptions it can be extremely complex to evaluate what the best intervention is to manage the problem. Therefore, in the past two decades, much research has been devoted to developing and evaluating airline recovery tools. The most recent developments show promising results in terms of solution quality and speed, often with less financial impact [2]–[5].

It has proven difficult to implement such recovery tools. Experienced end-users generally reject a recovery tool if they feel that the provided support is inadequate or incomplete, if the search algorithm is unclear, if it makes their work more cumbersome or if they simply do not trust the tool and think they can do better [6].

1.1 Aim and scope

This study aims to develop a decision support system (DSS) that can solve both minor and major disruptions and is also accepted by its end-users. The use of a DSS can allow the user's role to evolve from a tense environment to a higher level view of controlling and monitoring operations and making more well-grounded decisions. This study is done in collaboration with KLM Cityhopper (KLC).

KLC relies on its human experts to solve disruptions during its operations. Currently, it is very difficult for KLC to oversee the quality of the decisions made by the operations controllers. Hence, KLC is interested in ways to make the decision-making process more transparent and homogenous; towards a situation where it is known all legal options have been evaluated. As KLC is a regional operator, the study will be carried out in the context of short-haul operations that may differ from long-haul operations. The study will also be limited to prototyping and no finished product will be deployed for actual use. However, recommendations will be made that can be used to further advance the prototype and aid the development of a final product.

In Chapter 2, we will introduce the company and processes that are related to airline disruption management. We will analyse these processes in Chapter 3. Next, in Chapter 4, we will summarise the findings in the previous chapter and formulate the problem statement. In Chapter 5, the design requirements for the decision support system are presented. Chapter 6 contains an extensive literature review on past effort. In Chapter 7, we present the implemented algorithm. In Chapter 8, we present and discuss the results from our tests using the model in Chapter 7. Finally, we will come to a conclusion and make recommendations for further research in Chapter 9.

2 KLM Cityhopper: Operations, Planning & Control

This chapter provides a general description on the company, followed by a more elaborate explanation on the company processes that lead to the subject of this work..

2.1 Company profile

KLM Cityhopper (KLC) is a subsidiary of KLM Royal Dutch Airlines. It operates as KLM's regional carrier from Amsterdam and functions as a capacity provider to KLM. The set-up between both companies is such that KLC operates the flights on behalf of KLM that sells the tickets. Hence, KLC transports passengers for KLM. The financial basis of KLC is based on the number of executed flight legs and not on the amount of tickets sold.

Fig. 1 Fokker 70

Depending on the season, KLC operates at roughly 55 destinations, hereby operating approximately 250 flights and transporting around 20 thousand passengers a day. These flights are operated within KLM's so-called hub-and-spoke network, where KLC either travels to or departs from Amsterdam. Its fleet (May 2016) consists out of 16 Fokker 70s and 30 Embraer 190s and one Embraer 175, where, at the time of writing, the entire Fokker fleet is being phased out to be replaced by 16 new Embraer 175s. At the end of 2014, the total company size was 1266 FTEs (Full time equivalent), of which 80% are crew members and 20% are ground employees. [7].

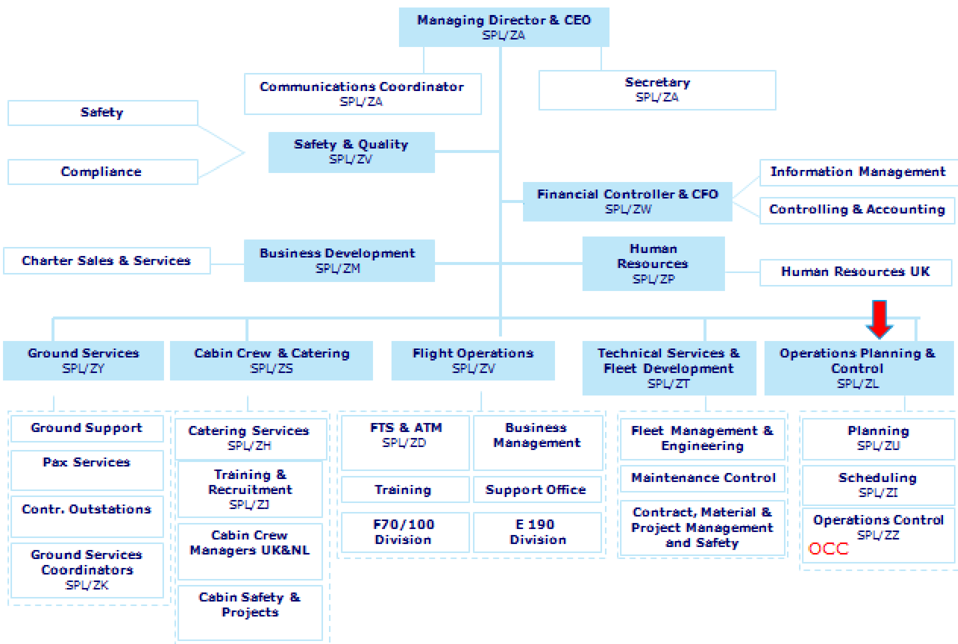


Fig. 2 The KLC Organisation

In Fig 2, the company is set up in several business units (in blue). The bottom five departments in the figure, Ground Services, Cabin Crew & Catering, Flight Operations, Technical Services & Fleet Development and Operations Planning & Control are the executing departments. Ground Services is responsible for monitoring ground handling services and carrying out passenger services. Cabin Crew & Catering is involved in the training and recruitment of cabin personnel, carrying out the cabin crew duties on each flight leg and everything catering related. Flight Operations is among other things concerned with training cockpit crew, and carrying out the pilot duties on the scheduled flight legs. The goal of Technical Services is to optimise fleet availability by ensuring all aircraft are operational as much as possible. Lastly, the Operations Planning & Control (OP&C) is responsible for the planning and scheduling of resources as well as controlling operations during the daily operation. This research project takes place within the domain of OP&C.

Fig. 3 Embraer 190

2.2 The Operations Planning & Control department

The OP&C department exists out of three divisions, namely: planning, scheduling and operations control.

The OP&C process typically starts with a timetable that is provided by KLM. The timetable is a collection of scheduled flights for a year and contains information about the scheduled departure and arrival times, departure and arrival stations and the aircraft type that should be used.

The timetable is created by factoring in the expected passenger demand and known fleet capacity in a certain period. Once the timetable is provided to KLC, the Operations Planning & Control department's objective it to realise each scheduled flight on this timetable.

The process of performing all scheduled flights on a timetable starts with the planning division, see Fig. 4. This division first develops the production figures that indicate the net amount of FTEs required for realising the timetable. Once these figures are found, the planning division will calculate the gross amount of crew required, hereby taking into account potential disruptions (e.g. crew illness, nonavailability) based on experience. When the required figures are found, the planning division will construct crew pairings. These are anonymous crew duties that are constructed to (near) optimality while respecting all legal rules and minimising costs. This entire process is iterative and starts about two years before and ends 2 months before the day of operation (DoO).

Based on the anonymous crew pairings provided by the planning division, the scheduling division will construct personalised crew rosters and publish every week the rosters for the following four weeks. This is then followed by a roster maintenance process; this is a continuous process during which changes in personnel availability are used to adjust the crew rosters. This takes place after the rosters are published and ends a day before the DoO. In the last step of the scheduling process, tail assignment, a fleet schedule is constructed by assigning all scheduled legs to tail numbers (aircraft registrations).

During the day of operation, operations are monitored from the Operational Control Centre (OCC). This is where the division Operations Control (OC) is situated. The objective of this division is to ensure that the original timetable is performed with minimal deviations. Nonetheless, on the day of operation, unforeseen events can cause for the carefully constructed fleet and crew schedules to become infeasible. For example, bad weather can cause airports to operate with a lower runway capacity and force KLC to operate on an adapted flight schedule, or a sudden aircraft malfunction right before departure that may not only affect that flight, but also cause a domino effect and affect other scheduled flights. Thus, Operations Control works to coordinate operations and prevent or reduce the impact of such disruptions.

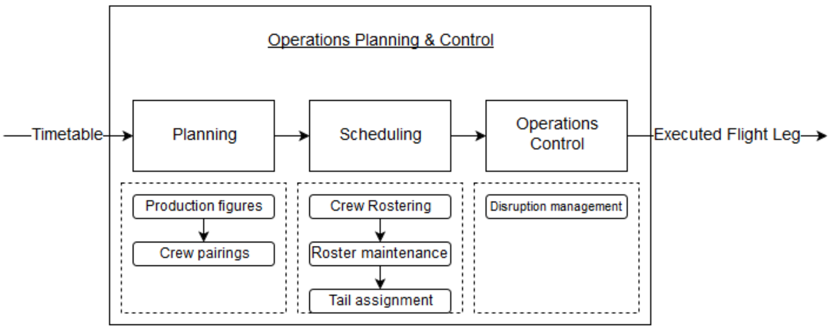


Fig. 4 The Operations Planning & Control department processes

2.3 Operations Control

Before we describe the processes that take place in the context of this work, we will first present the environment and the dynamics in which operations take place.

The KLC Operations Control division is together with their colleagues from KLM and Martinair Cargo situated in the KLM Operations Control Centre (OCC). This facility is set up to bring together everyone involved in the daily coordination of operations for KLM, KLC and Martinair Cargo. Though, during peak hours there can be about 120 active staff members, there are six functions with nine employees per shift that are manned by KLC personnel.

Fig. 5 The KLM Operations Control Centre

These six functions are: Duty Manager Operations Control, Fleet Controller, Crew Controller, Fleet Scheduler, Duty Manager Passenger Services, and Flight Watch. Together, they coordinate and manage all facets of the daily operation.

Duty Manager Operations Control (DMOC)

The Duty Manager Operations Control has a managing role in the OCC for that shift. The DMOC is responsible for the effective management of the Operations Control team. They are usually not directly solving any of the occurring problems. However, they are involved when problems escalate and the fleet controller is unable to solve the problems on their own, or for when approval from higher management is needed regarding certain decisions. For experienced fleet controllers, the position of DMOC is interchangeable with that of the FC.

Fleet Controller (FC)

The fleet controller monitors and coordinates the resource fleet. They fulfil a central role during disruptive operations in the operations control team. Therefore, they are principally the first contact person for any type of disruption. An FC will try to minimise the impact of disturbances on fleet availability and subsequently the flight schedule as much as possible.

Crew Controller (CC)

There are two crew controllers for the day of operation, one for cabin crew and one for cockpit crew. The crew controllers monitor and coordinate the resource crew. The main task for all crew controllers is to ensure all flights are staffed and that the crew operates respecting all law and statutory regulations. Like the FC, the crew controllers try to minimise the impact of disturbances on crew duties and subsequently the flight schedule as much as possible.

Fleet Scheduler (FS)

The fleet scheduler works for the days after the day of operation. They assigns the aircraft to the legs in the flight schedule. The fleet scheduler has to cover all unassigned flight legs while respecting the already scheduled maintenance

Duty Manager Passenger Services (DMPS)

The Duty Manager Passenger Services coordinates and communicates with gate agents during the boarding and disembarking processes. They also communicate with the service provider for passengers with reduced mobility to ensure there is assistance for them during both processes.

Flight Watch (FW)

Flight Watch fulfils a supporting role to the fleet controller. They coordinate and communicate actions that need to be taken during the aircraft rotations with the service providers. For example, towing to and from the hangar; keeping an eye out on outstations; informing cockpit regarding aircraft swaps; informing the fleet controller on weather changes or other relevant information. This position is fulfilled by two people that work together.

All these functions report to the Manager Operations Control.

Another function that does not belong to the KLC organisation, but is very relevant in the context of this work, is that of the Jr. Operations Controller.

jr. Operations Controller (jr. OC)

The jr. Operations Controller is a KLM position and is involved with the KLC Operations Control organisation during disruption management. This position is the result of the aforementioned set-up and relationship between KLM and KLC. The jr. OC has access to information about the financial consequences of delays and cancellations and provides this information to the fleet controller from a passenger perspective.

2.4 Disruption management

As stated earlier, KLC works with a time table (or flight schedule) that is provided by KLM. It is evident that a timetable with maximum utilisation of aircraft and very little slack generates the most revenue. However, a timetable with minimal slack is very susceptible to even minor disruptions such as delayed aircraft. On the other hand, a flight schedule with plenty of slack makes it simpler to cope with disruptions, but generates less income. Therefore, the provided time table is optimized to perform as many flights as possible whilst also factoring in for potential disruptions based on historic data and stochastic models. Now, regardless of how well constructed this flight schedule is, ultimately, disruptions remain inevitable. Unexpected bad weather could limit airport runway capacity, forcing the airline to fly with fewer flights; crew members can become ill, requiring a substitute to replace them; technical problems can prevent an aircraft from operating and many other disruptions could make the flight schedule infeasible. Dealing with such irregularities is named disruption management.

Disruptions typically affect the availability of resources in one or more domains hereby also adversely impacting passenger itineraries. The resources KLC controls are that of aircraft and crew. The process that is initiated to recover from a disruption is named the airline recovery or airline rescheduling process and can be split into three separate but overarching problems, namely that of the aircraft recovery problem, the crew recovery problem and that of the passenger recovery problem.

2.4.1 Aircraft Recovery

The fleet controller is responsible for the aircraft recovery process. This commences once an event has caused an aircraft to not be timely available for its assigned flight and the original fleet schedule needs to be modified to find a new workable one. The main objective here is to resume planned airline operations as quickly as possible with minimal deviations at the lowest possible cost. Deviating from the fleet schedule is generally undesirable as it may cause problems in other planned areas. The time span in which the recovery process is completed is called the recovery period. The length of a recovery period, which starts when a disruption occurs, varies with the problem, the type of flight and airline. The recovery period for a large airline that operates intercontinental flights may be a few days. For KLC the recovery period is typically a day.

To recover from disruptions, the fleet controller can take various actions as: delaying, cancelling, diverting (flying to an alternate airport), ferrying (flying an empty aircraft), and swapping aircraft, both within the aircraft type or between aircraft types. Generally, cancelling is the least desirable option, though this may not necessarily be true. There can be exceptions where cancelling is more cost efficient than delaying or other alternatives. For example, in some cases delaying a full flight may cause many connecting (i.e. transfer) passengers to miss their connection while cancelling another – rather empty – flight may not impose heavy financial consequences to the airline.

Financial information for the potential decisions is provided by the jr. OC. Due to the aforementioned set-up between KLC and KLM, the FC does not have the authority to access information about the impact of a delays and cancelations on the passengers. So, when a certain decision affects passengers (typically delays and cancellations) the FC will consult with the jr. OC to see if they agree to their proposal of delaying or cancelling a flight. In the case of a disagreement, the FC will discuss alternatives with the jr. OC. Though ultimately, the decision on how to handle a situation lies with the FC.

This example also illustrates that the aircraft recovery problem is not a fully isolated problem. Solving the disturbance from just a fleet perspective may create havoc in the crew schedule or passenger itineraries. Delaying or cancelling a flight without considering the impact on crew and passenger connections, can cause other problems if there is no crew available to operate the following flight, or if many passengers miss their connection; both having a dire financial impact. That is why the impact on crew and passengers is always considered during the aircraft recovery problem.

2.4.2 Crew recovery

The crew recovery problem lies with the crew controllers. Here crew controllers are responsible for finding solutions to disrupted crew schedules. These can be disrupted by crew related problems or due to preceding decisions or occurrences during operations or the aircraft recovery process. An example for crew related problems is that crew can become unavailable due to sickness and an example for the latter type is that crew can become unavailable due to a delayed preceding flight. Either way, the effect is that crew members become unavailable at the locations where and for when they are required. The crew recovery problem is in that way similar to the aircraft recovery problem: a resource becomes unavailable and the airline needs to find an alternative to continue planned operations.

Uncovered duties can be covered by swapping duty assignments between crew members, using reserve crew, asking for consent if crew members whom are not on duty are willing to work; and, of course, delaying or cancelling flights to make crew available again. The crew controllers will try to solve the crew recovery problem without affecting the planned flight schedule as much as possible. Though, sometimes it may be more logical to delay a flight until a crew member become available. This means that in some cases delaying can be a more attractive solution than using reserve crew members. This is because reserve crew members are a limited resource and require travel time to get to the airport.

Additionally, laws and regulations impose complex restrictions on crew work schedules. There are duty-hours limitations, rest-time limitations and contractual obligations to consider that are not present for aircraft. Such restrictions can be that crew members are not allowed to work for more than a certain amount of hours per fortnight or allowing crew members an hour to switch between duties. Such restrictions illustrate the difference between the crew recovery and aircraft recovery problems, namely that the crew recovery problem is imposed to more restrictions.

2.4.3 Passenger recovery

Traditionally, the passenger recovery problem is the last step in the recovery process. Here, the airline tries to fix itineraries for passengers whose itinerary is broken due to issues arising in the previous two problems domains. A broken itinerary is when at least one flight leg is cancelled, when the connection time between two legs is too short or when a booked seat becomes unavailable.

Because of the relationship between KLM and KLC, KLC does not directly employ anyone who is responsible for the passenger recovery problem. Instead, this process is initiated by the jr. OC. When passengers are affected by certain changes in the schedule, it is the responsibility of the jr. OC will inform their colleagues from the Commercial Desk who will initiate the passenger recovery process.

To recover the passengers, the airline can rebook disrupted passengers on other flights executed by the same or a different (partner) airline or refund the ticket. Usually, the preferred solution method is rebooking the passenger on another flight with available seats. When this is not possible the airline will try to rebook with a partner (or competitor) or ultimately, if there is no alternative, a refund will be given.

The interaction between the fleet controller, crew controllers and jr. operations controller in their respective domains is presented in Fig 6.

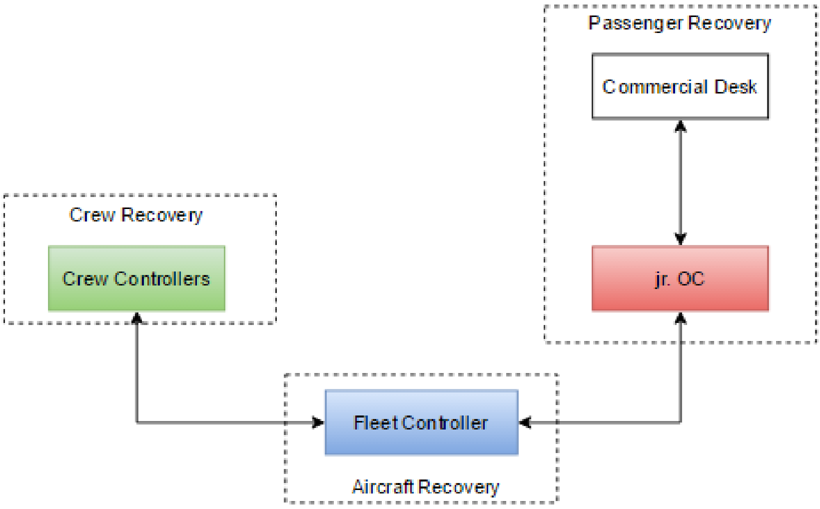


Fig. 6 A high level overview of the interaction between the fleet controller, crew controllers and jr. operations controller

The disruption management process is mathematically similar to the planning and scheduling process, described in Chapter 2.2. However there are also important differences that make the disruption management problem more complicated and challenging to deal with. These differences are: problem scope, input data quality, solution time and the objectives [6].

Problem Scope: Global vs Local

All available resources and information need to be considered to efficiently construct a flight schedule during the planning process. However, during disruption management, not all information may be available and only the affected part of the schedule needs to be considered.

Input Data: Static vs Dynamic

The data used for planning are static and reliable whereas disruption management takes place in a dynamic environment where the information is subjected to constant change.

Solution Time: Solving ahead of time vs reacting in real-time

The planning process takes place long before the execution process and is therefore not so time-sensitive whereas the disruption management process calls for quick decisions in real-time.

Objectives: Minimize total planned costs vs minimize deviations from scheduled plan

During the planning process, the objective is to define a plan that makes optimal use of resources at the lowest costs. An additional goal during the disruption management process is to minimize the adjustments to the scheduled plan as this is often already an optimized schedule.

2.5 Information systems

To gain a better understanding of the workflow, this subchapter explores the relevant information systems that are used in the process of disruption management.

The main information system used to monitor operations is Netline, a product of Lufthansa Systems. The Netline information system exists out of two modules, namely a Netline OPS (operations) and a

Netline Crew module. The first module is used by the fleet controller to track fleet positioning, fleet status and flight execution. The latter module is used by the crew controllers to track crew positioning, crew status and crew schedules. Another system that contains important information is the Traffic Flow Manager (TFM-tool).

2.5.1 Fleet tracking in Netline OPS

The Netline OPS system contains a Gantt-chart view of the fleet schedule, see Fig 7. On the left are depicted all aircraft. The horizontal lines for each aircraft represent the assigned flights, where the top row represents the scheduled times and the bottom row represents the actual or estimated times. The red vertical line represents the current moment. The user is able to tell the status of a flight by its colour. Red means the flight is not yet airborne whilst it should have been, and yellow indicates a propagated delay (because an earlier flight is delayed). Blue and green blocks are used for flights in progress, where the first means the flight will arrive late and the latter means everything is according to schedule. Other colours are used to indicate maintenance blocks, reserve aircraft or the status of a landed flight. More details can be accessed by double-clicking on the flights or in the text fields at the bottom of the window.

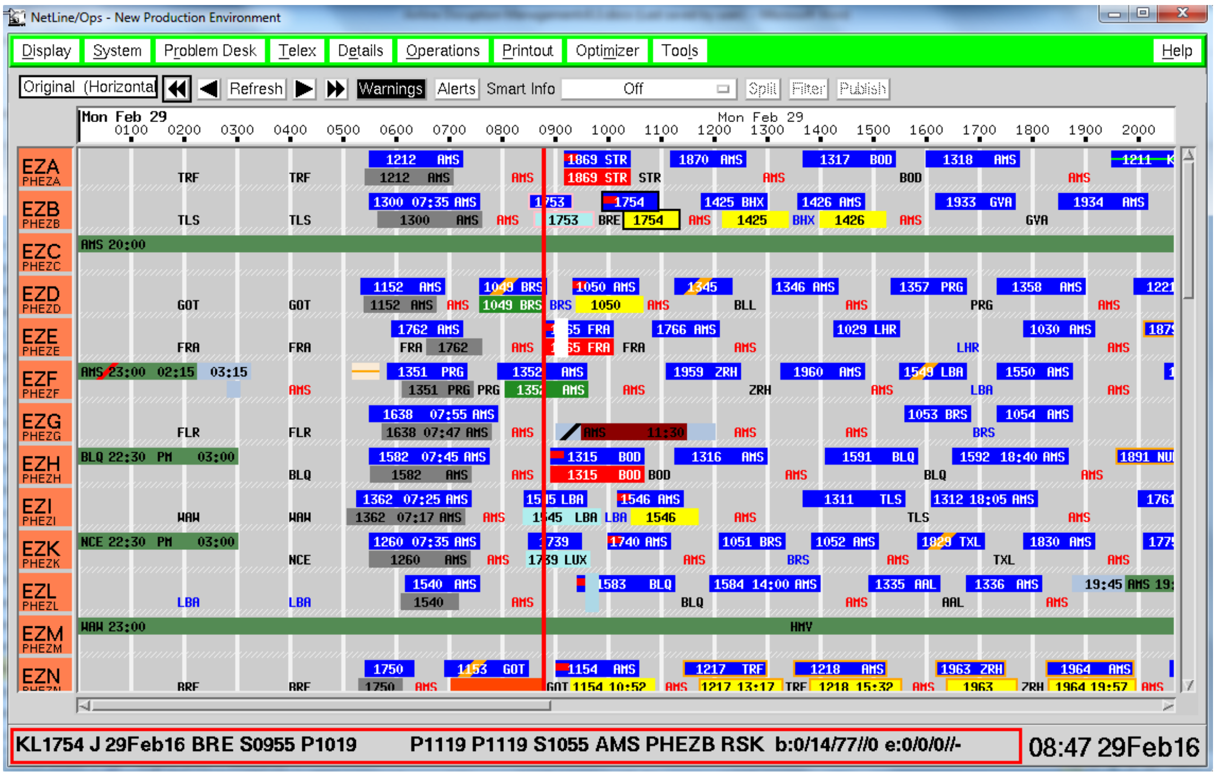


Fig. 7 Netline OPS. This image is only an excerpt of the entire schedule.

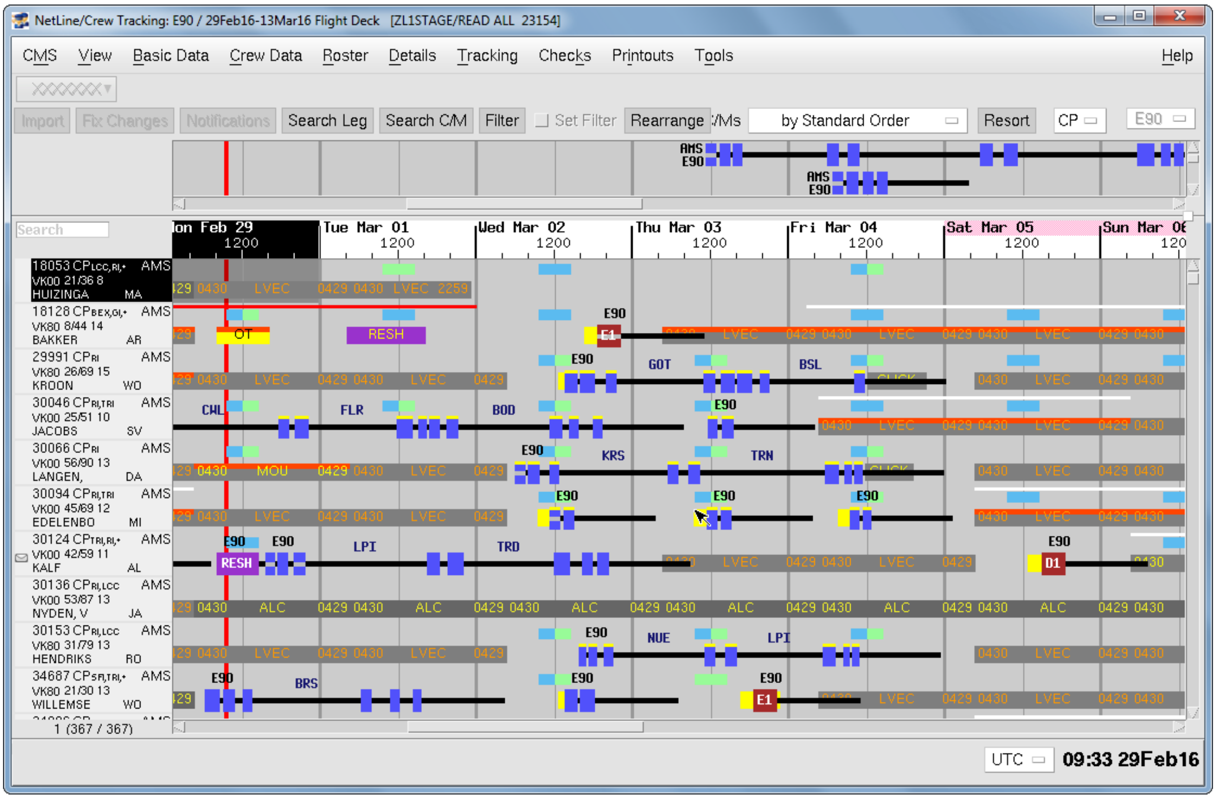


Fig. 8 Netline Crew. This image shows an excerpt for the Embraer pilots

2.5.2 Crew tracking in Netline Crew

Similar to the fleet module, the crew module also contains a Gantt-chart view. However, instead of showing the fleet schedule it depicts the crew schedule, see Fig 8. On the left are depicted crew members. The horizontal lines this time around show crew pairings, the black horizontal lines, and duties, the purple blocks. As described earlier, a crew pairing is a group of back-to-back duties, typically spanning 4 to 5 days. A duty is a flight where a crew member is assigned to. A difference in the view is that crew controllers typically track the crew members for over 5 days. This allows crew controllers to monitor crew members over the period of their crew pairing. Another difference in the view is that unassigned pairings (for the next days) are visible on the top part of the screen. The crew controllers knows that these duties are yet to be reassigned to a crew member. Other than that, the system works pretty similar. Using Netline Crew, the crew controllers can spot when crew members are not able to make it to their subsequent duty and can take measures accordingly.

2.5.3 The Traffic Flow Manager

The traffic flow manager is used to assess the impact of a delay or cancellation on passengers in terms of ensuing costs. For delays this calculation is based on the amount of passengers that will miss their connection if an incoming flight arrives too late. For cancelations it is based on the amount of passenger that are affected minus the fuel costs saved by not operating that leg. Both numbers consists out of a 'hard' and a 'soft' part. The 'hard' costs are the actual costs the airline has to make to rebook, to refund or to compensate passengers that are affected by a delay or cancellation. The 'soft' costs consist out of the negative impact on brand image and the likelihood the passenger will fly with KLM again.

The TFM arrivals-tool is presented in Fig. 9. This graph presents the cost of delay for a single flight. Each bar on the histogram represents an interval of 5 minutes, this can be seen on the top row. The cost break down is given by the colour segments, mainly cash impact and future value for this flight. The row 'Total miscon Pax' indicates how many passengers will miss their connection if the flight arrives too late. The TFM cancellations-tool, not presented here, informs in a similar fashion about the cost of a cancellation.

As stated earlier, this information is situated with the jr. operations controller and not directly accessible by the fleet controller.

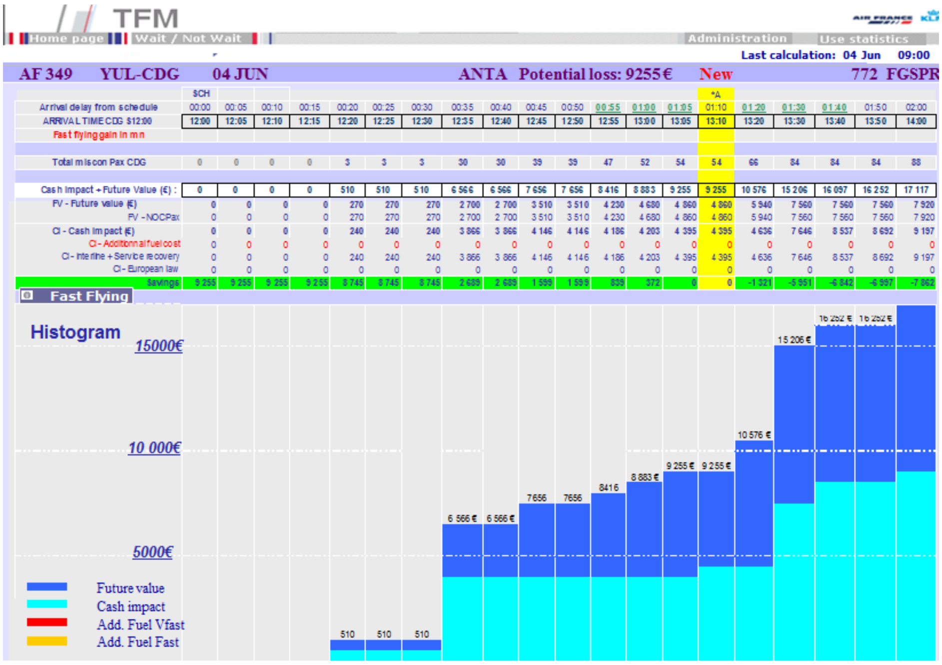


Fig. 9 The TFM arrivals-tool

2.6 Chapter Summary

This Chapter aimed to introduce KLM Cityhopper and its operations, planning & control department. As the topic of this work is improving on the disruption management process, only the relevant processes that take place before, during or after the disruption management are described. Extra attention is devoted to the used information systems as these play a very central role in providing the necessary information.

3 Process Analysis

This chapter analyses the processes presented in the previous chapter. For a systematic analysis and a thorough representation of the analysed system, we will make use of the Delft Systems Approach (DSA) [8]. By using a systematic approach we can identify existing problems and formulate the shortcomings herein. Once we have identified them we can define a definitive problem formulation.

First, the highest aggregation layer and system boundaries will be presented in the form of a black box. Then we will 'open' up the black box and zoom in on the relevant layers.

3.1 Main function and system boundaries

KLC's function as a whole is to transport passengers to their destination for KLM. We can draw this as the following black box:

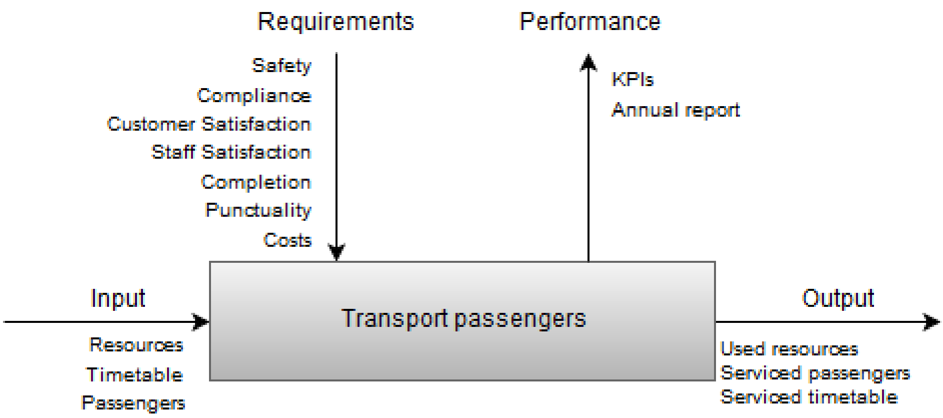


Fig. 10 The black box representation of KLC

KLM places a batch of orders with KLC in the form of a timetable that needs to be performed. Resources are needed to perform the service and passengers are what needs to be serviced. Thus, resources, timetable, and passengers as seen as input. The main function itself is to transport passengers on behalf of KLM. Each order is considered delivered when the scheduled flight leg is performed and the passengers are transported. Thus, as output, we have used resources, serviced passengers, and a serviced timetable.

Certain *requirements* that stem from KLC's mission – *"to carry passengers for KLM in Europe – safe, comfortable and on time – focused on flexible solutions at competitive costs"* – formulate to what standards the performance of the function of transporting passenger should be held against. Such requirements are: safety, completion, costs, punctuality compliance and staff and customer satisfaction. The *performance* of this function is quantified in certain Key Performance Indicators like, punctuality and completion percentage, certain scores for customer and staff satisfaction and many other metrics.

We draw the limits for the system's boundary at the order placement and the execution of a flight leg. Passengers order their itinerary with KLM and many of the passenger processes (tracking, rebooking, etc.) remain with KLM. Passengers only enter the analysed system when embarking the aircraft and they immediately leave the system when disembarking the aircraft after a performed flight leg. In the system, we can identify multiple aspects and their interrelations, to illustrate them we make use of a process performance model.

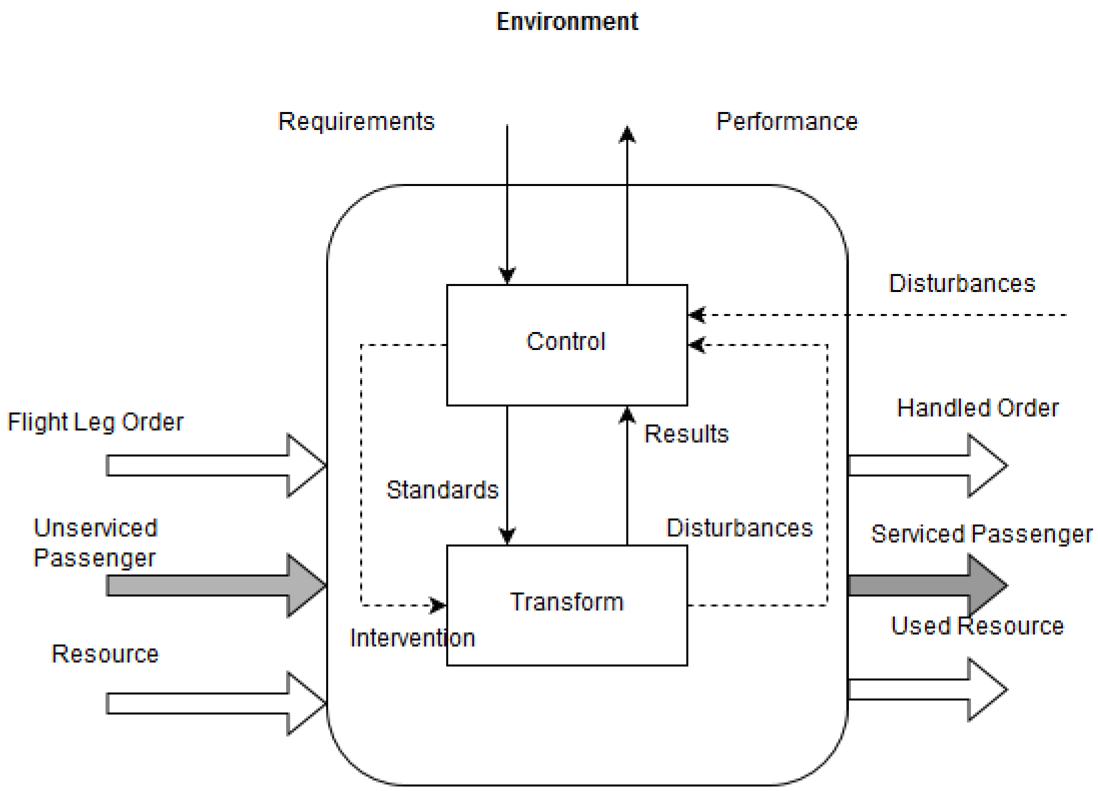


Fig. 11 The conceptual process performance model

The outer box in Fig. 11 represents the system boundaries. From the environment, the system receives three input flows: the order flow, that when nuanced, is a scheduled flight leg, the product flow in the form of unserved passengers and the resources flow in the form of resources that are required to perform the service. The thin, vertical arrows represent data or information flows. The incoming requirements arrow contains requirements from the environment for the executing process that are translated into standards for the executing process. Then, the results are measured and translated into performance metrics for the environment. Furthermore, there are internal and external disturbances that are measured in the controlling function. The impact of disturbances is evaluated and intervention is made if these have a significant effect on operations.

When we expand this conceptual model in Fig 12, we can gain insight in the relationships between the system's different aspects.

Here, the order that is placed is a scheduled flight leg. The order is considered 'handled' once the flight leg is performed. The operating process is to transport passengers. Crew and aircraft are resources that are needed to execute any flight. Finally, there is a controlling function named process control that intervenes when disturbances impact operations.

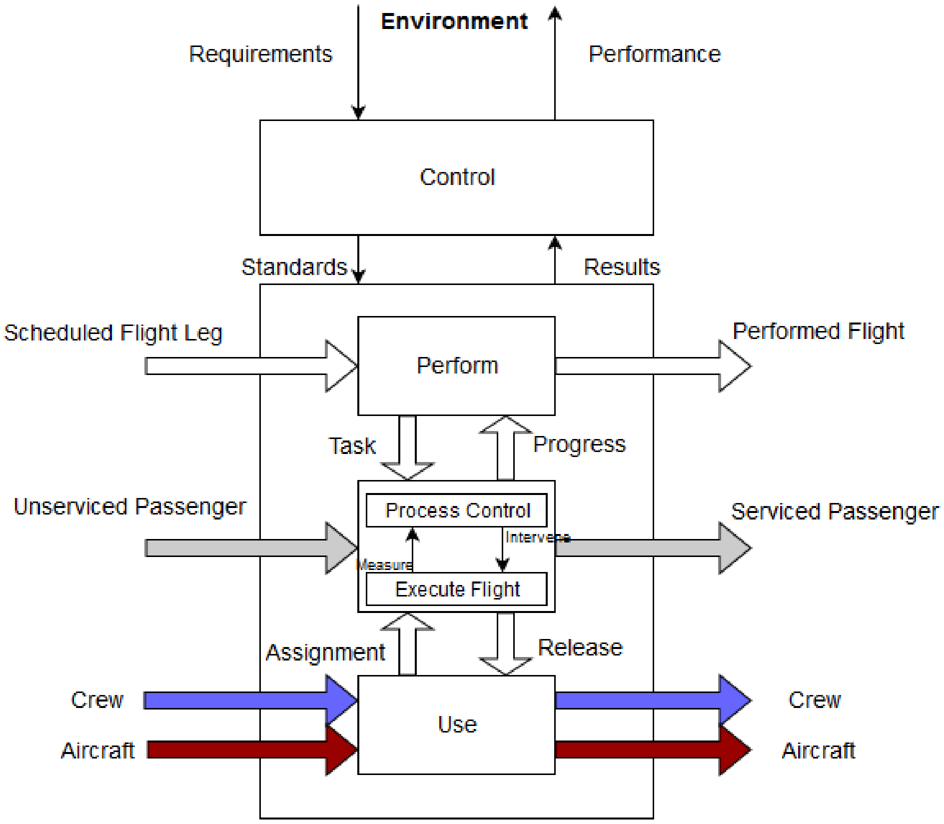


Fig. 12 The process performance model with the system's distinct aspects and their interrelations

3.2 Operation, Planning & Control

So far, we have given a formal description of the analysed system. However, before expanding on this, we should put this in the context of what is presented in the previous chapter, the operations, planning & control department.

On the whole, the operations, planning and control department's goal is to perform each flight on the contracted timetable by KLM. To do so, different processes take place in the three divisions in the operations, planning & control department.

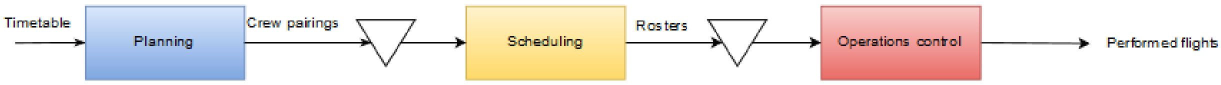


Fig. 13 The operations, planning & control dept.

Planning is responsible for creating anonymised crew pairings that are transferred to the scheduling division. The scheduling division uses these anonymised crew pairings to create personalised crew rosters and assigns aircraft to each flight. Finally, the operations control is responsible for ensuring the planned rosters and schedules are adhered to where possible.

We notice two things in this process: the planning and scheduling division are handling a certain order (i.e. timetable), and that operations control serves as a controlling function and only coordinates and controls the resources during operations. When this is linked to the model presented in Fig. 12, we see that the planning and scheduling process belong to the box 'perform', whereas Operations Control serves as the process control, located in the middle box.

Now, before we continue, let us restate the initial problem statement. Summarised, this was that the daily coordination of operations is heavily influenced by the skill and experience of the operators on shift. It is obvious that this problem relates to the process control.

3.2.1 Process control

In the Delft Systems Approach, process control is introduced as a unit that reacts to disturbances. Two types of process control are distinguished: feedback and feed forward. Feedback is when the value or state of the output, i.e. the real situation, is measured against the standard situation. If there is a deviation from the standard, the control function intervenes to compensate for the disturbance. Thus, feedback is reacting to disturbances.

The second form – feed forward – is when disturbances are measured and accounted for before they have time to affect the system. After which compensations are made for the influence of the disturbance. Thus, feed forward is a proactive form of control.

Before we analyse what type of control form is employed at operations control we should identify the different types of disturbances, hereafter called *disruptions*.

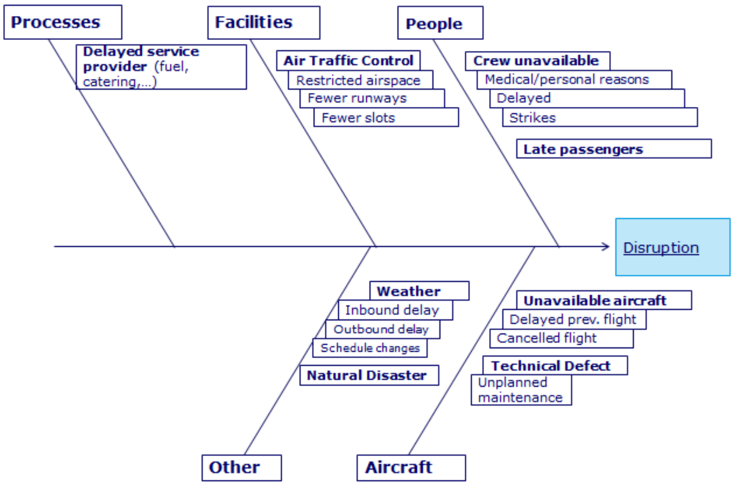


Fig. 14 The causes of disruptions

3.2.2 Types of disruptions

Many different events can cause a disruption. The Ishikawa – diagram in Fig. 14 introduces the type of disruptions and categorises these according to where they originates. The disruptions can occur in processes, facilities, people, aircraft, and weather. A disrupted process can be that certain service providers are delayed during the aircraft turnaround. A weather and facility related disruption is that air traffic control entities can limit traffic to and from the airport in the case of adverse weather. Disruptions caused by people are, for example, unavailable crew that can hinder departure if there are no replacements found, or if the flight waits for late passengers. Aircraft related disruptions are aircraft that can become unavailable due to a technical defect or delays elsewhere in the network. Table 3 specifies when information for each category is available. As can be seen in the table, most of the time information about a disruption is only available when it occurs. Only weather related information is consistently available beforehand. However, the accuracy of it may be inconsistent. Nonetheless, when the weather forecast is bad enough, the airline will act accordingly to alter the schedule as deemed necessary.

Table 1

Type	*Category*	*When is this information available?*	*When it occurs*	*Beforehand*
Other	Weather	The weather forecast is consulted multiple times a day. However, at times the expected weather may not match the actual weather. So while this information is known beforehand, the weather may still take an unexpected turn.	✓	✓
	Natural disaster	When it occurs	✓	×
Processes	Service providers	When it occurs	✓	×
Facilities	ATC	A few hours ahead, when the air traffic control entities inform the airlines. Or due to unexpected weather	✓	✓
People	Crew	Information about unavailable crew members can become available both beforehand (e.g. strikes or long illness) or when it occurs	✓	✓
	Passengers	Information on late passengers is usually available a short time (5-15 min) before departure. Except when a group of passengers is on an earlier delayed flight, than that information is known beforehand	✓	✓
Aircraft	Unavailable aircraft	Aircraft that become unavailable due to operations is mostly known when it occurs	✓	×
	Technical defects	Any technical defects become known during inspections between flights	✓	×

3.2.3 Disruptions and process control

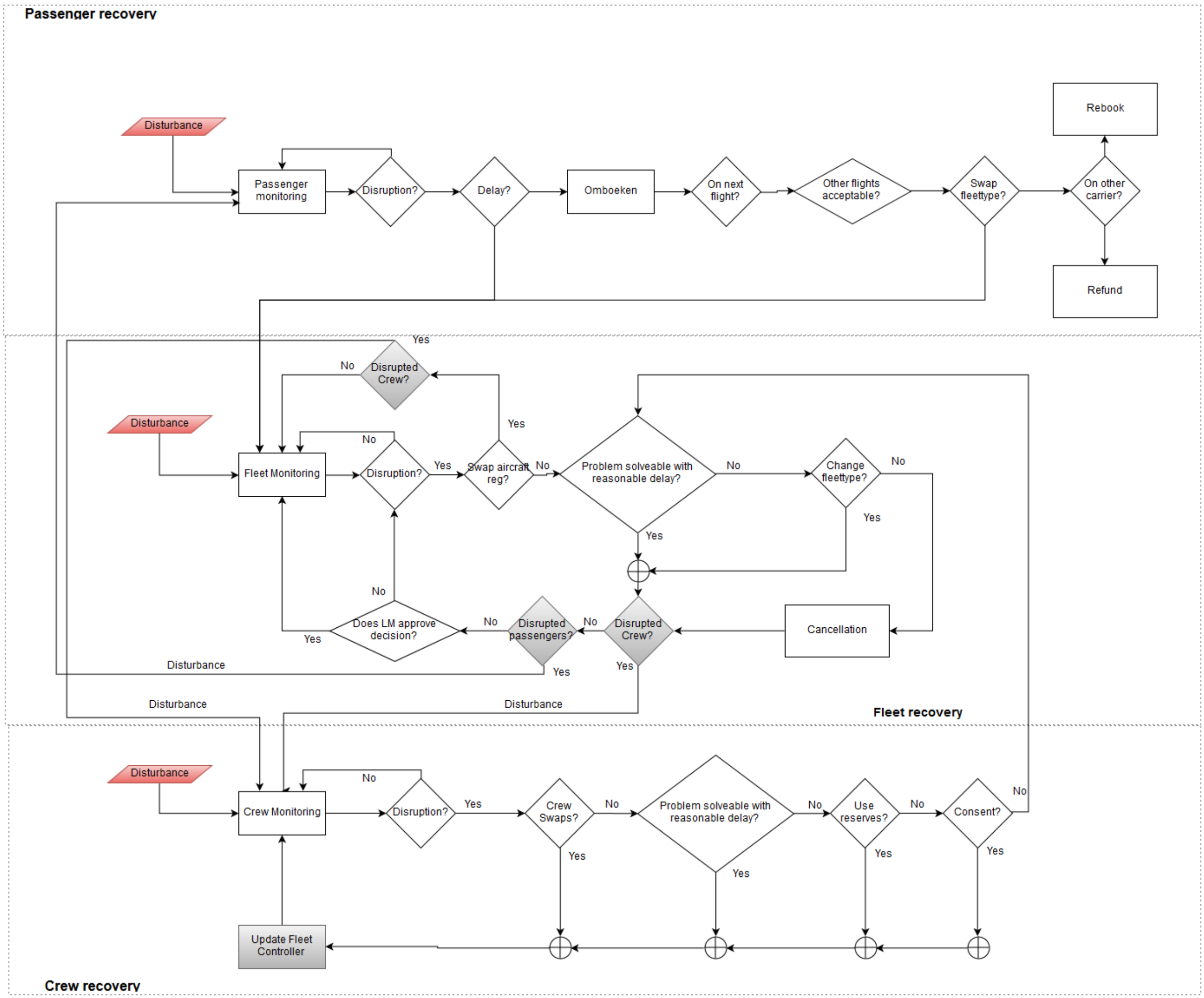
Operations control employs both types of control forms. For large disruptions that are known beforehand, i.e. weather, strikes, KLC evaluates potential actions based on the expected impact of it. A possible solution can be revising the flight schedule by cancelling a number of flights, thereby creating more slack in the schedule. During such problems, the airline has more time to analyse and explore the potential actions and employs a feed-forward type of control. For disruptions that only become known when they occur, a feedback type of control is employed. In this type of control, the airline has typically less time to analyse and explore the potential actions because a quick solution is desired.

A control function, be it feedback or feed-forward, exists out of four phases: measuring, comparing, controlling and intervening. Measuring is the stream of information that is used (e.g. Netline). Comparing is evaluating this information against the set standards, i.e. the current schedule. Controlling is determining the actions. Finally, intervening is implementing these actions. Now, we will put this in the context of the current decision-making process.

3.2.4 The decision-making process

The flowchart in Fig. 14 depicts the current decision-making process for the entire airline recovery problem at KLC. The domains passenger recovery, aircraft recovery and crew recovery can be seen in the top box, middle box and bottom box respectively. The flowchart depicts the decision-making process for each subproblem and also shows where and how each subproblem connects to another subproblem.

Fig. 15 The decision-making process



The recovery process for each subproblem starts when any of these three monitoring processes signals a disruption. The flowchart then follows the actions to mitigate a disruption in order of preference. Monitoring is the measuring of information and comparing it against the set standards, i.e. the planned flight schedule, crew schedule and passenger itineraries. If a significant amount of deviation from the standard (i.e. schedule) is found the control phase begins. Herein, the controller decides on the appropriate actions: delay the flight(s), swap aircraft, swap to a different fleet type or cancel the flight. In the illustration, we see that there is a certain hierarchy to the decision-making process. It is important to note that this hierarchy is not definite but serves as a rule-of-thumb. The consensus is to avoid cancellations where possible.

Sometimes a delay of flight A may be preferred over a swap of the aircraft on flight A with the aircraft on flight B, or vice versa. As stated in Chapter 2, it can even such that a cancellation of a flight may be preferred over delaying. Also, a controller has to be able to evaluate all feasible actions to mitigate a problem. Currently, there is no information system available that suggests/provides viable moves to mitigate problems (i.e. aircraft swaps, subtype swaps, delays and cancellations). So, a controller is limited to their skill and experience in finding these feasible actions.

We see that during the decision-making process, the impact of on the flight schedule and the ensuing costs are important factors that are consdidered before implementing a solution. From Chapter 5 we know that the impact on the flight schedule can be assessed in the current fleet tracking software. We also noted that the costs of delays and cancellation are available in the TFM tool, albeit to the Jr. OC. This means that during decision-making, as we see in Fig. 13, the Jr. OC needs to be consulted for any potential action that involves delays or cancellations. Only then is the FC informed about the ensuing cost of a decision. Because of this they can only find out this information for a handful of potential actions when there is time pressure. Exploring a plethora of solutions becomes difficult in this situation: there is a certain time pressure and it is impractical to enquire about all possible solutions the FC has in mind that involve delays or cancellations. Next, we identify what criteria are considered during the control phase of the decision-making process.

3.2.5 Criteria for decision-making

A plethora of criteria is considered during decision-making. Though not exhaustive, *Table 2* gives a list of the most frequently considered criteria.

Table 2 What are the criteria are considered during decision making?

	Criterion	*Description*	*Items*	*Importance*
FC	On-time performance	The on-time performance of flights is of high importance to the company	ETA/ETD, Block times, Runway capacity, Slots, airport opening hours	High
	Slack in schedule	The more slack there is in the schedule, the more options there are for rescheduling		High
	Weight limitations		Dry-operating weight-fuel, Max-take-off weight, Fuel	Low
	Maintenance restrictions	Planned and unplanned maintenance can impose certain restrictions on fleet availability	01/02 notification, possibility to postpone	High
	Operational restrictions	Certain defects can impose restrictions on aircraft that prevent it to fly to some destinations		Medium
	Weather	Weather can influence flight operations before take-off, during flight and before landing	Thunderstorms, heavy winds, mist, impact on flight schedule	High

	Passengers	The impact of decisions on passengers is always considered during decision-making (with the jr. OC)	NOCs, TFM-tool, capacity restriction when changing to a smaller fleet type	High
	Crew Connections	The impact on crew duties is also considered during decision-making		High
CC	Ensuring all flights are staffed	A flight always needs 2 pilots and 2 cabin attendants to be operational		High
	Duty hours	Law, statutory and contractual rules impose many restrictions on working and resting hours, that need to be adhered to at all times. These restrictions are also different for UK and NL crew members.	Law, CAO, rest time, LTA, Earliest Reporting Time, connection times between flights,	High
	Qualifications	Law, statutory and contractual rules impose restrictions on which positions crew members can man.	RHQ, Rank, Fleet type, Experience (Green on green, blue on blue, grey on grey)	High
	Trainings	CC are careful to not schedule employees that are on training		Medium
	Reserves	CC are careful to preserve reserves where possible as they may be needed later		Low
	Other agreements	There are certain agreements between KLC and its employees	Meal regulations	Medium
	Consent	CC can ask if crew members that are not on duty are willing to work		Low

For the aircraft recovery process we find that the following criteria are of high importance:

- On-time performance: This is one of the KPIs of an airline. Flights that depart and arrive on time do not create havoc in the flight schedule or cause passengers to miss their connections, keeping the operational costs down.
- Slack in the schedule: Rescheduling is easier when there is plenty of slack available in a schedule. Consequently, rescheduling becomes more difficult when there is very little slack. Hence, the size of the solution space depends in part on how much slack there is in the schedule.
- Weather: Weather can direly impact airline operations. The decisions that are made are in part affected by the (expected) state of the weather.
- Passengers: The airline's main goal is to serve its passengers. Hence, the effect of a decision is also evaluated in terms of how it impacts the passengers.
- Crew Connections: Even if an aircraft is available there may not enough crew available to operate it. So obviously, it is important to consider the availability of crew when rescheduling flights. However, there are different restrictions on crew transfer time than to aircraft turnaround times. So even though a solution may be feasible from a fleet perspective, it may be not feasible from a crew perspective if the solution violates certain crew restrictions.
- Maintenance restrictions: Safety is an important aspect of airline operations and therefore aircraft are regularly scheduled for different types of maintenance. Scheduled maintenance may

or may not be definite at that time. Definite maintenance cannot be rescheduled, whereas indefinite maintenance can be rescheduled to a different time.

3.2.6 Summary

We have noted that the Operations Control entity within the Operations, Planning & Control department serves as the controlling organ (process control in DSA terms) of the airline. Operations Control employs two types of control, namely feed forward and feedback. A feed forward type control is used with larger expected disruptions (e.g. expected bad weather, strikes). A feedback type control is used to mitigate the impact of disruptions that have already occurred (e.g. late incoming aircraft, technical defect).

Decisions made during feed forward are typically a team effort. Because there is less time pressure more information sources can be consulted during this process. Decisions made during feedback are done by the fleet controller. The most prominent information sources that are used are Netline and TFM. Both information systems contain important information, though the TFM information is not readily accessible. This information is situated with the jr. OC and it is impractical to consult them on information for all feasible solutions that involve delays or cancellations. Thus, they are usually only consulted for a handful of possible solutions. Furthermore, criteria like the impact of decisions on passengers, crew connections and on-time performance, available slack and scheduled maintenance are important parameters to evaluate decisions on.

4 Research question

This study is initiated with the aim to investigate to aid operators in their decision making process. Currently, this is a manual process based on experience and quick-thinking of these operators, with little to no transparency regarding the implemented decisions. In the previous chapter there is a clear implication that this is because not all the required information is timely available or accessible during the control phase of the process control function, namely:

- The financial impact, i.e. the resulting cost of decisions
- The passenger impact, how many passengers are affected by the intervention?
- The available slack, what possible moves were there available?
- The impact on crew connections. Are there any crew regulations violated and are there crew members available?
- The impact on on-time performance, i.e. do these action delay other flights elsewhere and is this necessary?

To address these items the following research goal is formulated:

"Design a decision support system that provides insight on the quality of possible solutions to a disruption by informing on the consequences thereof."

To reach this goal, we formulate the following items:

- Define performance criteria for a decision support system
- Study relevant literature to identify approaches that are able to deliver the solutions for the occurring problems
- Develop a prototype decision support system for operational disruptions occurring
- Verify and validate the tool using representative data
- Make recommendations to further advance the model

In the following chapter, design requirements, we will discuss the quality criteria for the decision support system. In Chapter 6, we will discuss relevant literature on the airline recovery problem to identify what type of approaches have been taken, from which perspectives the problem at hand has been investigated and to identify what has been already produced. In Chapter 7, we will present the selected and implemented model. Then to verify and validate the correctness of the model, a few cases will be presented in Chapter 8. Finally, in Chapter 9, we will summarise and discuss the findings and make recommendations for implementation and further advancing the decision support system (DSS).

5 DSS design requirements

Here, we will discuss the design requirements and define the performance criteria for the DSS that have been selected with the Operations Control management.

5.1 Scope

From Chapter 2, we know that the airline recovery problem consists out of three individual but overarching problems. We also noted that KLC has an operational responsibility (resources), while KLM has a commercial responsibility (passengers). Hence, KLC is in charge of the aircraft and crew recovery problem while KLM oversees the passenger recovery problem. This has some consequences for the governance of this project.

Ideally, our decision support system would address and support across all three domains. However, the many restrictions and regulations make the design of such a system a very complex endeavour. And seeing how this study is limited by manpower and time, we will have to set priorities. Therefore, to prove the concept and utility of this decision support system, we will first limit ourselves to the aircraft recovery problem. Both the crew recovery and passenger recovery problem will not yet be considered. The aircraft recovery problem has the benefit over the crew recovery problem that there are fewer restrictions and regulations to consider as well as that there are fewer overall resources, making the problem more manageable. The passenger recovery problem will not be considered as this lies organisationally with KLM. Whilst, the problem domain of crew recovery will not be considered, recommendations for further research will be made in how to incorporate this in the selected approach. Also, at this stage crew restrictions will also not be considered during this stage. That being said, seeing as how information about the passenger impact is readily available, the impact on passengers will be considered.

Additional restrictions/features can be added incrementally if the aircraft recovery module is proven to work. We that the decision support system will serve as a prototype and not yet be implemented in the existing IT infrastructure during the course of this research project. Implementing requires an application with graphical user interfaces, user accounts and connection to the existing IT infrastructure. This falls in the domain of software development and is beyond the scope of this work. Instead by using extracts it is possible to demonstrate the use and interaction between these systems. That being said, there is a good possibility at implementation if it is proven successful.

Finally, the DSS is aimed to be used reactively and not on expected disruptions. Therefore, during this phase the weather information will not be considered by the DSS.

5.2 Requirements

As stated in the research goal, from the DSS it is desired that it provides insight on the quality of solutions. To be able to do so, characteristics have to be selected to assess an initial situation and a potential outcome, but also provide insight in the amount of effort that is needed to reach the situation. The characteristics are (1) number of affected passengers, (2) the number of aircraft swaps and (3) subtype swaps, (4) the amount of delay minutes, (5) the number of delays and (6) cancellations, and (7) the cost value of the solution.

Additionally, important information will not always be available in the used computer systems. Therefore, it is not desired of the DSS to find a single optimal solution. Additionally what may be optimal from a quantitative perspective may be sub-optimal from a qualitative perspective. Thus, the DSS should be able to generate a diverse pool of solutions. This enables the operator to weigh the solutions differently depending on circumstances that may not be taken into account in the DSS or other information systems.

In Chapter 2.5, we have also discussed the existence of the Traffic Flow Manager. This information system contains relevant information from a commercial and passenger perspective. We can use this as input for our DSS to evaluate the passenger impact. So while our model will be limited to the aircraft recovery problem, we will still be able to consider the passenger impact. Hereby, making the DSS more valuable and realistic. Doing so, also leaves the responsibilities of KLM and KLC as they currently are. By using the same information, KLC operators will be able to better evaluate outcomes without undermining KLM's authority.

A very important requirement is that the DSS should adhere to the existing maintenance slots. At KLC, maintenance is typically scheduled in before the tail assignment of flights. While it is occasionally possible to postpone scheduled maintenance, this is not something that should be decided by the DSS.

To search for solutions to a problem, the system should be able to react on the most common type of occurrences, i.e. late incoming aircraft and technical failures (AOGs), and react using the most common actions, i.e. swapping aircraft within and between fleet types, delaying flights and cancelling flights.

To recap, the DSS should adhere to the following requirements:

- A diversity of solutions should be presented using characteristics
- The TFM information should be used as input
- Maintenance slots should be considered
- Finding solutions using the following actions:
    - Swapping aircraft
    - Swapping subtypes
    - Delaying flights
    - Cancelling flights

Performance criteria

The DSS quality will be evaluated on the following criteria:

- Its ability to find a diversity of solutions
- Its ability to find realistic solutions , i.e. feasible and no-nonsense solutions
- Its ability to provide insight

- CPU time

6 Literature review

In this chapter, we will investigate what is already produced in literature. We will be looking at the various perspectives researchers took. A holistic approach will be taken towards the disruption management problem, so not to miss any innovative work – though in the end, the focus will remain on the aircraft recovery problem. The literature will be evaluated based on the design requirements mentioned in the previous chapter and on benchmarks that are provided by the investigators and researchers.

6.1 Introduction

Airline disruption management has seen an increasing amount of attention since it was pioneered three decades ago by Teodorović and Guberinić [9]. The initial efforts focused on solving conflicts for a single resource at a time. With the advancement of computer hardware and the progress in solving methods, researchers have started involving the multiple aspects of the recovery process in more sophisticated models. More recent work focuses on integrating aircraft with either crew, passengers or both with the aim of reducing the total impact and thus improving the overall quality of the solution. This chapter will discuss the different approaches to airline disruption management found in the literature.

In the late ‘80s and early ‘90s, all research was focused on aircraft recovery. Crew recovery started to appear in the late ‘90s, during which integrated approaches were also first discussed. Passengers were first considered in a fully integrated approach in a Ph.D. thesis in ’97 and received more attention in the 2000s. It becomes conspicuous that most, if not all, work in literature can be categorised according to which aspects of the airline recovery process they address, namely aircraft recovery, crew recovery, passenger recovery or integrated methods.

The bulk of the produced work, 73% according to Castro et al. [2], has been done using operations research methods, such as integer optimization, network flow models, column generation and metaheuristics. Most approaches (75%) can be classified as models and algorithms, meaning that they were not included in tools or systems. Approximately 22% can be classified as decision support systems, meaning they were included in tools and 3% is considered an automatic or semi-automatic system. It is also interesting to note that research into the isolated problems has peaked between 1996-2001 and 2002-2007 for aircraft recovery and crew recovery respectively. Contributions that propose integrated methods have been on the rise since 2007.

Clausen et al. [10], Kohl et al. [11] Clarke et al. [12] and Castro et al. [2] provide comprehensive reviews of the earlier work on airline disruption management. To save time and energy, and to focus on new developments, the focus in this review will be on papers released from 2006 onwards.

First approaches that focus only aircraft recovery are discussed, followed by approaches that focus on the isolated crew recovery. Finally, the integrated approaches – aircraft and crew, aircraft and passenger, and fully integrated methods are discussed.

6.2 Aircraft recovery

Out of the three problems, aircraft recovery has received the most attention. Probably because airlines operate much fewer aircraft than the number of crew members and the rules are typically less complex [10]. Most results indicate that the proposed exact methods can only solve relatively small problems. Hence, many heuristics are developed that find good feasible solutions in a reasonable time frame. Much of the current focus is on developing more sophisticated solution methods that provide better solutions for larger instances.

Andersson [13] compares approaches based on meta-heuristics. Flight data is used from a Swedish airline with the largest extrapolated instance of 58 aircraft, 5 aircraft types 436 flights and 41 airports. The model allows for delays, cancellations, aircraft changes, and fleet swaps; the objective is to minimize passenger delays. He finds that an approach based on tabu search is always able to find a solution within 15 seconds and that is less than 0.3% from the best-known solution for that instance.

Liu et al. [14] adopt a multi-objective evolutionary algorithm method for when an airport is temporarily closed. The method allows for delaying and swapping, cancelling and ferrying are not considered. The method is tested on a Taiwanese airline with 7 aircraft of one aircraft type (MD90), 70 flights and 7 airports. The authors report that the method finds feasible solutions in an acceptable short time.

In [15], Liu et al. propose another multi-objective optimization approach based on an evolutionary algorithm. They incorporate aircraft swaps, total delay time, delay flights and delay over 30 minutes in their model. They use the flight schedule of a Taiwanese airline with 7 aircraft, 72 flights and 6 airports. The authors state that the experimental results show that the proposed method can recover the disrupted schedule within a very short duration.

Liu, Chen and Chou [16] introduce one more multi-objective optimization approach. This time, based on a hybrid genetic algorithm. This approach, similar to the ones in [14] and [15], is developed to use during temporary airport closures. Apparently, something that happens often in Taiwan. No description is given for the instance used, but the authors argue that the simulated experiment shows that a hybrid method outperforms "many" in most study cases and that the solution meets "most requirements"; statements that are rather vague.

Zhu and Zhu [17] propose a mixed set programming approach that is scalable for larger instances. They find that the proposed MSP approach finds solutions in < 10 minutes that are better and faster than a greedy simulated annealing algorithm found in literature, where the largest instance is 46 aircraft, 3 aircraft types and 173 flights with 5 defective airports. The model incorporates delaying, cancelling and changing aircraft; it is unclear if they also allow changing fleet types.

An improved greedy random adaptive search procedure (GRASP) is proposed by Zhao and Guo [18]. They use data consisting out of 250 flights serviced by 50 aircraft. The schedule is then disrupted in different scenarios. They find that the improved GRASP provides faster and better solutions than the GRASP found in the literature. The same authors also propose a GRASP approach based on ant colony optimization [19]. They state they find similar results using this approach though a more extensive comparison is omitted.

Zhang and Hansen [20] explore whether surface transport modes or the combination of alternative airports and surface transport modes can be used as a substitute for further delays when the cost associated with either option is more cost effective than the latter. They find that is "potentially a useful strategy to alleviate the terminal congestions in the US". Though, further research is needed to test on real life instances.

Eggenberg et al. [21] develop an approach that can be adapted to the aircraft recovery, crew recovery, and passenger recovery problems. They validate the model for the aircraft recovery problem with heterogeneous fleet and maintenance restrictions by successfully solving a disrupted situation with real data from a medium-sized airline. Generally, their approach works very fast (<1 min). They conclude that their approach shows that the algorithm is efficient and is able to solve complex instances in low computational time.

An aircraft recovery model with stochastic elements is adopted by Arias et al. [22]. The authors combine a simulation and optimization approach that allows for evaluating the robustness of the provided solutions. The objective in this approach is to minimize total delay and cancellations. An instance of real data, based on 51 flights and 13 airports, from a commercial airline is used. They find that the improved flight schedule shows more robust behaviour than the original one.

Quansheng et al. [23] discuss the aircraft recovery problem. Only delaying and cancelling is considered in and swapping of resources is ignored. They find good solutions to a small analysed instance of three aircraft, twelve flights and four airports. This work is too simplified and thus not so useful in practice.

Aguiar et al. [24] discuss a hill climbing, simulated annealing and genetic algorithm approach to the aircraft recovery problem. Their model allows for delaying, cancelling and aircraft swaps. Real data from TAP Portugal is used that consists out of 51 aircraft, 2 fleet type and 3521 flights in one month. The objective is to minimize delay minutes. The authors find that the genetic algorithm performs best though all three approaches converge to a solution in approximately 4 seconds.

Sousa et al. [25] presents an algorithm based on ant colony optimization that can solve both the aircraft assignment problem from the airline planning process and the aircraft recovery problem from disruption management. The instance used is based on real data from [26]. The algorithm aims to minimize the operational cost and also incorporates disrupted passengers in the cost function. The model allows for flights to be delayed, cancelled and rerouted. Experiments show that the proposed approach finds very good final solutions in reasonable time. However, the authors also note that the model is not yet adaptable for use in operations because the constant change in aircraft assignments could affect security protocols and long-term flight planning.

6.2.1 Summary

Thirteen methods that address the aircraft recovery problem have been discussed. All, with the exception of [17], use a heuristic as the selected approach. There is also great variety in the instance sizes used. Thus, comparing the performance of these methods becomes somewhat complicated. The methods in [13], [17], [18], [21], [24], [25] incorporate the same actions as used in practice. These authors also report good results with good or reasonable solution time. The most promising of these are [13], [18] and [24] as the first two produce (cost-wise) a good solution within an impressive 15 and 5 seconds, respectively; the latter converges to a solution in about 5 seconds for the instances presented.

6.3 Crew recovery

Approaches addressing crew recovery can be split into three categories. The first category is crew recovery with a fixed flight schedule. Recovery models in this category follow the traditional sequential approach used by airlines and attempt to repair broken pairings that stem from changes in the flight schedule. The second category fit approaches that incorporate decision variables, which allow for cancelling flight legs. Approaches that allow for flight delays to solve the crew recovery problem form the third category.

The nature of the recovery problem makes it so that it requires quick solutions. To achieve this, many approaches will limit the solution space by either applying a time-window technique or by only including the affected crew members and a number of selected candidate crew members. A time-window is a period that starts when the disruption occurs and ends after a certain number of hours into the future. The length will typically vary from a few hours until a few days.

The crew recovery problem has received considerably less attention than the aircraft recovery problem. Only fourteen papers have been published so far, of which only four in the last decade.

Nissen and Haase [27] present a duty-based formulation that assumes a fixed schedule and is tailored to the needs of European airlines, where usually fixed crew salaries are the norm. The duty-based approach ensures that a disruption is solved within each duty period. This allows for shorter recovery horizons and thus a smaller solution space. The approach is tested with short- and medium-haul flights and on different scenarios ranging from the delay of a single flight to a several hour long airport closure. The authors find that, after some tweaking of the model parameters, this approach is capable of providing solutions within a short period of time.

According to Medard and Sawhney [28], the crew pairing and rostering problem have to be solved in the same time in the recovery phase. Thus, the crew recovery problem is a combination of both. They propose a model that assumes a fixed schedule and merges the pairing characteristics into a rostering problem and solve it using column generation. The used instances are for single base and multi-base problems with a recovery period of 48 hours. The largest single base instance considers 422 planned crew members with 20 illegalities and the largest multi-base instance considers 855 planned crew members with 77 illegalities. Single base instances are solved in approximately 90 seconds and multi-base instances can take up to several minutes. Almost all instances are solved except for the largest instance.

Castro and Oliveira [26], [29] propose, in two similar papers, the use of a distributed multi-agent systems (MAS) that represent the existing roles in an OCC. Though the MAS include an aircraft recovery agent, a crew recovery agent and a passenger recovery agent, both papers' experiments address only the crew recovery problem with fixed flight schedule. In their work the authors do not discuss what kind of algorithms and heuristics are used and only one test scenario is disclosed. (They do discuss the algorithms and heuristics in a recently published book that proposes an integrated approach discussed in 6.4.3 [2]). The test scenario is a simulated situation where 15 crew members, with different ranks, are reported absent. They compare their solutions with human operators and find that, in average, their method took 25 seconds to find a solution with a cost of 3839 whereas the human operators took 101 seconds with a cost of 7040.

Aguiar et al. [24] also propose a hill climbing and simulated annealing approach to the crew recovery problem in the same work. The approach allows for swapping crew members under the assumption the flight schedule is fixed. Three disrupted scenarios are simulated with 473 cabin crew members and 109 pilots active in a full month flight schedule. The objective calls for a minimal cost function that consists out of a salary and overtime component, and a penalization for using a spare crew member. The authors report that the used method is able to solve all crew connection problems in a very fast timeframe (largest in 13 seconds). However, the improvement in the cost function between the disrupted situation and the solved situation is marginal at best.

Zhao et al. [30] transform an earlier approach to crew recovery with departure delays to a grey programming model. The authors' purpose is to introduce the concept of grey programming to irregular flight operations as they do not present much detail on their test case.

6.3.1 Summary

In the last decade, four papers have been published on this topic. Four that fit the first category, crew recovery with fixed flight schedule, and one that fits in the third category, crew recovery with departure delays. No papers have been published in the last decade that fit in the second category, crew recovery with flight cancellations.

The inherent problem with the isolated crew recovery problem is that the successfulness of the approach depends on the robustness of the flight schedule. Either the flight schedule needs to be already fixed or more problems that need to be resolved will be introduced (in the form of delays or cancellations). Aside from this, the mentioned approaches do show promising results in terms of solution time and it can be interesting to conduct a thorough evaluation of the state-of-the-art in real scenarios.

6.4 Integrated recovery

The entire airline recovery process with its three subproblems is a combinatorial task that is very challenging to solve. Hence, most of the initial efforts focused on a single subproblem. The increase in computational power has allowed researchers to develop integrated approaches that achieve more robust recovery results than approaches that address a single subproblem. The first truly integrated framework was first proposed in a Ph.D. thesis in 1997 though only parts of it were implemented at the time.

Integrated approaches can be categorised in approaches that address the aircraft and crew recovery, the aircraft and passenger recovery and fully integrated approaches.

6.4.1 Aircraft and crew recovery

An integrated recovery approach is reported by Abdelghany et al. [5]. The authors propose a decision support tool to deal with larger disruptions that are foreseeable as in the case with Ground Delay Programmes. They formulate a modelling framework that integrates a schedule simulation model and an optimization solver in a rolling horizon framework. A possibility to both cancel and delay flights is incorporated in the model. The model is then tested on a scenario 522 aircraft, 1360 pilots, and 2040 flight attendants. The authors report promising results of approximately 30 seconds for solution time and delay savings of about 5% compared to a "do-nothing" state.

Zhang et al. [31] argue that the approach proposed by Abdelghany et al. [5] is "highly efficient but solution quality is not guaranteed" and state their aim is to propose a new algorithm that makes "a better trade-off between solution quality and efficiency". Thereto, they propose a two-stage heuristic algorithm for the integrated aircraft and crew recovery problem. In the first stage, they solve the integrated aircraft problem with partial crew considerations. In the following step, the integrated crew recovery with partial aircraft considerations is solved. This allows the aircraft connections generated in the first step to remain feasible. The approach incorporates delays, crew deadheads, use of reserve crew, crew and aircraft swaps, and flight cancellations. Computational results show that high-quality solutions can be generated within 2 min.

6.4.2 Aircraft and passenger recovery

Bratu and Barnhart [32] present two passenger recovery models, which allow for delays, cancellations and assigning reserve crew and aircraft to flight legs. The first model is the so-called Passenger Delay Metric model and the second model is the Disrupted Passenger Metric model. The difference is that the first uses exact delay costs and the latter uses approximate delay costs. The authors report that the first approach cannot be solved in real time as it takes too long but that the second approach is fast enough to be used by operations controllers.

Jafari and Zegordi [33] consider flight rotations and effect of cancellations and delays on passenger itineraries. They test their approach - in a scenario with 13 aircraft, 2 fleet types, 100 flights, 19 airports and 2236 passengers - against an earlier approach that focuses only on aircraft recovery. The authors report that their approach achieves a cost reduction of 12.2% or 6.8% depending on the parameters used.

In 2009, the French Operational Research and Decision Support Society (ROADEF[1]) issued an airline disruption management challenge. The goal was to create a tool that recovers both passengers and aircraft in an integrated approach. The challenge required that the computation time should not exceed 10 minutes. The challenge's winner, Bisaillon et al. [34], and seventh in place, Jozefowiez et al. [35], published their work.

Bisaillon et al. [34] introduce a large neighbourhood search heuristic for the integrated airline and passenger recovery problem. Given an initial schedule, a list of disruptions and a recovery period, this approach alternates between construction, repair and improvement phases, which iteratively destroy and repair parts of the solution. The first two phases find an initial feasible solution and the third phase improves upon this solution. Randomness is introduced in the first phase to diversify the search. The results show that the proposed method finds a solution in all tested instances that range from small to very large (256 aircraft, 44 airports, 1423 flights and 11,565 itineraries). The found solutions are almost always in the top 3 of all the tested methods. Computation times are set to 10 minutes.

Sinclair et al. [4] improve upon the model presented by Bisaillon [34] et al. by introducing a number of refinements in each phase so as to perform a more thorough search of the solution space. The model is tested on the same instances and yields the best-known solution cost for 17 out of 22 instances within 5 minutes of CPU time and for 21 out of 22 within 10 minutes. The authors find the cost between delay and cancellations disproportional such that delays appear less significant. They suggest a better understanding of the relation between cost of delay and cost of cancellations to come to a more accurate objective function. The same authors present in [36] a column generation post-optimization heuristic which, when applied after the LNS heuristic proposed in [4], improves greatly upon the solution cost with a slight increase in CPU time. This improvement finds the best-known solution to all instances from the 2009 ROADEF challenge. The authors note that the algorithm can also be modified to solve larger instances by only considering passenger variables.

A heuristic based on shortest path problems is proposed by Jozefowiez et al. [35]. Their approach consists out of three phases. During the first phase, the disruptions are integrated into the initial plan. The goal in the second phase is to reassign passengers to itineraries with the same origin and destination. If not all passengers can be reassigned in the second phase, new flight legs are created in the third phase. Though this method does not yield better solution costs, they do outperform Bisaillon et al. [34] in terms of CPU time. The model proposed by Jozefowiez et al. [35] never takes longer than 4 minutes to reach a solution, in most cases even less than a minute.

Marla et al. [37] are the first to propose an integrated aircraft and passenger recovery approach that also considers flight planning to trade off delays and fuel burn. On data from a European airline, they

[1] http://challenge.roadef.org/2009/en/

find a total cost saving of about 6% percent, a decrease in passenger disruption of up to 83% and increase in fuel burn of 0.15% percent.

Similar to Marla et al. [37], Arikan et al. [38] discuss an integrated aircraft and passenger recovery approach with cruise time controllability. They contribute to the literature by integrating cruise speed control with other recovery actions such as retiming departure and arrival times and swapping aircraft. They argue that delays can be greatly mitigated by factoring in cruise speed control. They use real life data from a U.S. airline and find that 97% of the instances can be solved to optimality in real time.

Le et al. [39] model the integrated aircraft and passenger recovery as a vehicle routing problem with time windows (VRPTW). The aircraft are modelled as vehicles, airports as nodes and passengers as commodities. The time window consists of a lower bound that is the STA and an upper bound that is the STA plus maximum affordable delay. They integrate the VRPTW with a genetic algorithm to improve CPU time. Data from a Chinese airline is used to validate the model. The authors report near-optimal solutions in a fast period for their instances. However, they only factor in delaying and do not consider other possible actions, such as aircraft swaps or flight cancellations.

A column-and-row generation approach is presented to achieve an exact solution by Maher. [40]. The author attempts to directly provide passengers with alternative travel arrangements following flight cancellations, something which has not been attempted so far. The model is tested on a point-to-point network that consists of 262 flights, transporting 28,492 passengers travelling on 48 aircraft to 20 destinations. The vast majority of the instances are solved within 20 minutes - which is, according to the author, "an acceptable runtime for practical use of the algorithm"; I would like to argue otherwise. A 10 minute solution time is discussable, but a 20 minute solution time in a dynamic environment is definitely not practical.

Hu et al. [3] discuss a multi-fleet routing approach considering passengers transiting under airline disruptions. They consider aircraft swapping, flight delays and cancellations and ignore ferrying, diverting and using of reserve aircraft because they are, according to the authors, "rarely if ever available". They use data from a Chinese airline (largest instance of 628 flights, 13 fleet type, and 178 aircraft) and introduce randomly generated disruptions based on the airline's experience. The CPU time is less than 30 seconds in the largest instance with an average optimality gap of 3%. The authors report that the airline's manual heuristics found solutions with significantly greater costs.

6.4.3 Fully integrated recovery

Lettovsky was the first to discuss a fully integrated system in his doctoral thesis. The model's immense dimensions make it computationally intractable and thus unusable for a real application.

Petersen et al. [12] find that the crew recovery forms a bottleneck in the fully integrated recovery process. They propose to make the problem tractable by finding a solution that is globally optimal with respect to passengers, locally optimal with respect to crew, and feasible to aircraft. The model is integrated by applying Bender's decomposition to create a master problem and three subproblems. The model is tested with data from a U.S. airline that consists out of approximately 800 daily flights and two fleet types. The integrated approach is then tested against the sequential approach on five scenarios. The authors report that their approach finds better solutions than the sequential approach for two scenarios and equally good solutions for the other three in terms of passenger and flight delays. The integrated approach takes approximately half an hour to solve.

A distributed multi-agent system approach that represents the existing roles in an airline operations centre is adopted by Castro and Olivieira [2]. The agents are so-called specialist agents and use different heuristics (hill climbing, simulated annealing and Dijkstra's algorithm) to tackle each subproblem. A negotiation protocol is implemented to find the best solution for the three subproblems. They find that the multi-agent approach obtains feasible solutions faster and with less cost compared with the manual and sequential approach used at a Portuguese airline. Unfortunately, nothing is reported on solution times.

6.4.4 Summary

This subchapter categorised the integrated approaches according to which part of the recovery process is addressed. The bulk of the published work is on the integrated aircraft and passenger recovery, presumably because this is the easier than integrated aircraft and crew recovery.

All discussed approaches use some sort of heuristics to search the solution space and the consensus is that integrated approaches often offer better solutions in terms of ensuing cost. There is, however, a lot of difference in reported solution times. Some authors report solution times in less than a minute [3], [5], whereas others report solution times up to 20-30 minutes [12], [40].

For aircraft and crew recovery, both the discussed work by Abdelghany et al. [5] and Zhang et al. [31] look promising; a high-quality solution for very large instances is reported in under 2 minutes.

For aircraft and passenger recovery, the approaches discussed by Sinclair et al. [36] and Hu et al. [3] look the most promising; the first reports the best-known solutions for the used instances and the latter finds solutions with an optimality gap of 3% in less than 30 seconds.

Integrated recovery is the most desired and most challenging approach for disruption management. Petersen et al. [12] report good solutions that take long to solve. Castro et al. [2] report an innovative[2] approach where they model the operations control centre as a distributed multi-agent system where the agents represent the roles that already exist in the OCC. The approach is tested with a Portuguese airline and the proposed approach finds better solutions than the manual approach and a sequential approach. The authors also report that this approach is being implemented[3] with the airline. We have also discussed this approach with the authors in the context of disruption management at KLC. The authors stated that they have not been able to fully test their model in practice with TAP. Partly, due to

[2] This approach has won the "Best Innovation Award" at the AGIFORS 2015 conference: http://masdima.com/home/2015/05/17/agifors-airline-operations-2015-2/

[3] http://masdima.com/home/2015/07/21/poc-at-tap-is-coming-to-an-end/

the political circumstances at the airline in the wake of a recent privatization and partly because not all required information is available. Nonetheless, a POC with the authors and KLC is being investigated.

6.5 Chapter summary

The topic of airline disruption management has received an increasing amount of attention ever since it was first pioneered 30 years ago. Almost all initial work was focused on a single subproblem. Researchers agree that to achieve more robust and cost effective solutions, the subproblems need to be integrated. Hence, the more recent work has been focusing on developing integrated methods – with different success.

It is difficult to conclude what the best performing method is as they almost all are used with different instances and on different problems. It would be comparing apples to oranges. However, some approaches – as discussed in the subchapter summaries – are more promising than others.

That being said, there are some problems that are seldom addressed:

1) Researchers argue that longer solution times (> 10 minutes) are acceptable:

From experience with a different application at KLM, we believe this to not be true. Operators are quick to give up on software systems that take too long. This is especially true in a time-sensitive environment such as the airline operations control. Operators, need to solve problems fast and are generally not willing to wait 10 or more minutes every time a disruption occurs. However, what is an acceptable run-time is also vague, because it is heavily influenced by the circumstances in which it is run. That being said, we believe run-times – for especially complicated problems – of up to a few minutes is acceptable.

2) Disruptions are not modelled consistently:

In the discussed literature, there is a difference in how disruptions are modelled. Some approaches assume all the disruptions are known. Others solve for a single set of disruptions as they become known. The former can be useful when bad weather is expected or a strike is planned, and it is less useful when these are not known – which is often. The latter's objective, solving for disruptions as they become known, is to return to the original flight schedule. This is great for the first set of disruptions that occur on a particular day. However, this becomes problematic if the tool suggests returning to the original flight schedule when this is already modified due to earlier disruptions. In this case, this will not be so desirable. The challenge is in being able to do both, solving for disruptions that are already known and for solving new disruptions as they occur.

3) Unrealistic simplifications:

Some of the works in literature make simplifications that are unrealistic. A bare minimum for evaluating aircraft recovery should, in our opinion, incorporate a combination of delaying and cancelling flights, swapping aircraft, and using reserves; not just a few of these. Another thing that is often overlooked yet is important, is the inclusion of maintenance requirements. Aircraft maintenance is typically planned ahead and should be adhered to. This imposes restrictions on the availability and location of the aircraft.

4) User acceptance:

In the introduction, it was stated that it has proven to be difficult to implement recovery tools. An example of this is the Global Rescheduling System, developed by Air France. This recovery tool is able to delay flights, cancel flights, and swap aircraft. The users are able to modify the parameters to manipulate the suggested solution. However, the GRS is not used for its intended purpose because users find it too complicated. Almost none of the works in literature discusses this rather important aspect of user acceptance. Typically, flight schedules are represented in Gantt-charts, but schedule modifications can be presented in many different ways. For the user, it is important that the solution is presented in a simple and understandable manner.

7 A model for the aircraft recovery problem

The aircraft recovery model we will use is based on the work of Andersson [13]. As discussed in Chapter 5.1, this approach – together with the approaches in [19] and [24] – is one of the more promising ones amongst the discussed aircraft recovery literature. It is difficult to make an objective comparison between the approaches as they are all used on different datasets and allow for different actions. However, the tabu search proposed in [13] is better equipped to obtain structurally different solutions than the other two approaches that both converge to a solution with a lower cost over time. One of the design requirements was for a diversification of solutions and the tabu search allows for easy and configurable diversification. Certain solutions can be made 'tabu' for some period so the search can be diversified. Something that is more difficult to achieve by using mathematical programming or the other presented heuristics which are designed to converge to a global optimum.

The presented results in Andresson's work for the tabu search are within 0.3% of the best-known solution for the tested instances. Additionally, the instances used are comparable in size to KLC's fleet size; the approach is able to find a solution within 15 seconds for the tested instances; and, as required, the approach allows for delays, cancellations, aircraft changes and fleet type changes.

7.1.1 Modelling assumptions

The problem being modelled is that of the aircraft recovery problem. The objective is to find a revised aircraft schedule to an initial disrupted schedule. To do so, flights can be delayed or cancelled. Aircraft can be swapped between assignments, both within the same aircraft type or between aircraft types. This includes upsizing and downsizing. The passenger capacity lost when downsizing is factored by penalising it in a cost function.

The aircraft recovery problem starts when a problem occurs, e.g. aircraft malfunction or delayed flights. The moment when the problem is observed and has to be solved is called the *start time*. The period in which changes can be made to the schedule is named the *decision period.* The decision period ends at the *end time.* This is when the aircraft schedule should have returned to normal and flights can be resumed as originally scheduled. The decision period should be long enough for there to be solutions and not too long as the number of solutions grows exponentially as this time increases.

At the start time, each aircraft is either operating a flight or scheduled for one, and assigned for a sequence of flights. Deviating from this sequence increases costs. Thus, unnecessary swapping is discouraged and penalized. Most of KLC's incoming passengers are connecting passengers, meaning AMS is not their end destination and they have to transfer to another leg to continue their journey elsewhere. Delayed flights can cause for these passengers to miss their connecting flights. Hence, incoming late arrivals are associated with cost. This information is extracted from the TFM data. Finally, cancelled flight legs are also associated with a cost value that is also acquired from the TFM data.

The value of a solution is thus based on the associated cost. This is a function that is the sum of the cost of delayed flights, cancelled flights and swapped aircraft. Using TFM data allows for a more refined evaluation of solutions. It can be the case that a 5 minute delay has a zero cost value while a 10 minute delay on the same flight has a significantly larger (non-linear) cost value. This can occur when a 5 minute delays allows for all passengers to be able to make their connection while with a 10 minute delay many passengers will miss their connection. Though solutions will be evaluated on the cost, it is interesting and beneficial for the controller to have a pool of structurally different solutions to choose from. After all, depending on the circumstances, the operator may not find the solution with the lowest cost the most desirable.

A solution S will have as characteristics cancelled flights c, delayed flights d, aircraft swaps with a different aircraft type f, aircraft swaps with the same aircraft type s, and an objective function z. These solutions can be ranked on these parameters and then presented to the user who will have distinctly different solutions to choose from. This allows for the user to utilise their experience, as they are able to judge the solutions with other aspect that are not covered by the system.

7.1.2 Local Search

Heuristics have been long used in the field of operations research to find solutions to combinatorial problems. A heuristic can be described as a problem specific, approximate solution technique. Unlike exact methods, a heuristic does not guarantee that the found solution is mathematically an optimal solution. Nonetheless, heuristics have been very popular to problems that are typically classified as NP-hard. A benefit of a heuristic over an exact method is that a solution can be reached that is satisfactory to a problem in a shorter time than that is possible with exact methods.

A class of heuristics, named metaheuristics, have been shown to be applicable for a broader set of problems. Metaheuristics employ some sort of stochasticity to diversify the found solutions. Though again, no guarantee for optimality can be given, quality solutions can be obtained in a reasonable amount of time. A very popular example of a metaheuristic is the local or neighbourhood search. A local search is an algorithm that starts with an initial solution. A new solution is obtained by making a change in this initial solution. Solutions that are found in the vicinity of this initial solution are called neighbours. The entire collection of solutions that can be reached from a solution is named the neighbourhood. In a local search, the neighbourhood is explored and the best neighbour is selected as the improving solution. When no better neighbours can be found, the algorithm terminates. An example of the local search in pseudocode is given below.

Algorithm 1: Local Search generic form

1. x_0 ← Some initial random candidate solution
2. **While** the current solution x has a superior solution y:
3. set $x = y$
4. **Return** the final solution x

7.1.3 Tabu Search

The tabu search used in this work, is, in essence, an enhanced local search algorithm. It differs from the local search by the addition of a *tabu list*. The tabu list allows for temporarily accepting worse solutions than the current solution. This enables the search to escape from local optima and to find better solutions. The tabu list contains a set of rules that describes if a move to a certain solution is allowed. This prevents being stuck in a local optimum and also prevents cycling, repeated visits to the same solutions. The size of a tabu list and the exact rules are to be determined for the problem at hand. An example of the tabu list in its simplest form would be making the L latest solutions tabu.

Algorithm 2: Tabu Search generic form

1. S_{best} ← Some initial random solution
2. $TabuList \leftarrow \emptyset$
3. **While** some stopping condition is not met {
4. $CandidateList \leftarrow \emptyset$
5. $S_{bestCandidate} =$ **null**
6.   **For** each candidate solution S_{cand} in the neighbourhood of s {
7.   **If** TabuList does not contain S_{cand} **and** $fitness\ (S_{cand}) > fitness(S_{bestCandidate})$ {
8. $S_{bestCandidate} \leftarrow S_{cand}$
9.   }
10.   }
11. $S \leftarrow S_{bestCandidate}$
12.   **If** $fitness(S_{bestCandidate}) > fitness(S_{best})$ {
13. $S_{best} \leftarrow S_{bestCandidate}$
14.   }
15.   **If** size of $TabuList$ is equal to its maximum size {
16.   remove first entry in $TabuList$
17.   }
18.   Add $S_{bestCandidate}$ to $TabuList$
19. }
20. **Return** S_{best}

Lines 1-2 represent the initial set up. Here, a random solution is selected to begin the search with and an empty tabu list is created. The tabu list serves as a short term memory structure that contains the last visited elements.

The algorithm starts at line 3. This loop will search for an optimal solution until a certain stopping criteria is met. This can be a certain number of iterations, or a specified CPU time. In lines 6-7 each candidate solution is checked for tabu status. If the solution is not tabu and if it is 'fitter' than the current solution it is accepted as the new best neighbouring solution in line 8. The fitness of a function is usually a mathematical function that returns a value for that solution. If this neighbouring fitness has a better fitness value than the current best solution, it is selected as the new best solution. If the tabu list is full the oldest entry rejected at line 16. Then the local best candidate is always added to the tabu list, this happens at line 18. Finally the best solution is returned at line 20.

Two of the main factors contributing to whether a tabu search is successful, or not, are: the size of the tabu list and the way the tabu aspiration criterion is defined [41]. The tabu list size is problem specific and thus to be selected for the problem at hand, which can be done in three ways: fixed to a predetermined value, randomly chosen from a specific range, or by dynamically adjusting the value. Salhi [41] argues that the last method of determining the list size is the most convincing, though the other two may produce good results. It is found that the size of the tabu list depends on the tabu conditions, where more restrictive condition require a smaller tabu list size and less restrictive tabu conditions require a larger tabu list size [42].

It is important to carefully select the tabu aspiration criterion. Making this too strict can make the search too restrictive and prevent obtaining good solutions. On the other hand selecting them too weak can increase the search time or make it even impossible to reach such a solution.

For the problem at hand, making an aircraft tabu is probably the most obvious choice of a tabu attribute. If an aircraft is made tabu, it cannot be selected again for the next L iterations. However, this also means that a large part of the neighbourhood cannot be visited for the next L iterations. This is an example of making a change in the solution tabu. Andersson [13] argues that making an aircraft tabu is not suited for the problem studied here. Andersson instead suggests making a solution tabu. Whether a solution has been visited recently can be evaluated by checking the assignment for an aircraft. A solution is a timetable that contains information on what routes the aircrafts fly. However, this requires enumerating over all routes, which is not so efficient. A more efficient way of doing this is by making the value of a solution tabu. Meaning, the number of delays, delay minutes, cancellations, swaps within aircraft type, swaps between aircraft types and the cost associated with it are used as the tabu characteristics. This makes it less time consuming to compare solutions and check for tabu status. A caveat is that in rare cases different solutions may have the same values, but this also means that the solutions have very similar characteristics. In this case making this solution tabu can still benefit the algorithm as the solutions share very similar characteristics, and accepting such a solution does not diversify the search.

7.1.3.1 The tabu search algorithm

The tabu search algorithm is adapted from Andersson [13] with some minor but useful improvements. The first modification is adding a Fisher-Yates shuffle to the algorithm. The second modification is that Andersson uses a recursive search and a set-packing algorithm to construct routes, whereas we use a recursive nested tree build algorithm.

In our algorithm we modify the schedule by traversing aircraft pairs and seeing if we can modify the existing schedule by using that aircraft pair. Without shuffling the list of aircraft pairs, we will continually try the same combinations using the aircraft pairs in the same order. Shuffling can help us obtain solutions we would not have found before. The Fisher-Yates shuffle is an unbiased algorithm that shuffles items in an equally likely probability. While this is not necessary to obtain a diverse pool of solutions, we found that by using the Fisher-Yates shuffle, the algorithm was able to generate even more solutions. In a brief test, one with and one without the Fisher-Yates shuffle, the algorithm generated 12 solutions in 50 iterations and no new solutions were obtained after the 12th iteration. With the Fisher-Yates shuffle, the algorithm was able to find between 42 – 49 solutions to the same case in the same solution time. Nearly a unique solution for each iteration. In Fig 14 and Fig. 15 it is demonstrated that by using the Fisher-Yates algorithm, it is possible to diversify the obtained solutions even more.

Andersson first constructs all possible routes for each aircraft in an aircraft pair, as we will explain later. Then selects two routes for both aircraft in the aircraft pair by considering all routs for each aircraft a set-packing problem, which is NP-hard [13]. As is evident in an NP-hard problem, we found that the time required for finding both routes would increase exponentially with each added flight. Instead we opt for a recursive nested tree build algorithm. We first create a route for one aircraft, after we each immediately create the route for the other aircraft. Hereby eliminating the set-packing problem. We can then obtained the desired route by simply iterating over the generated routes.

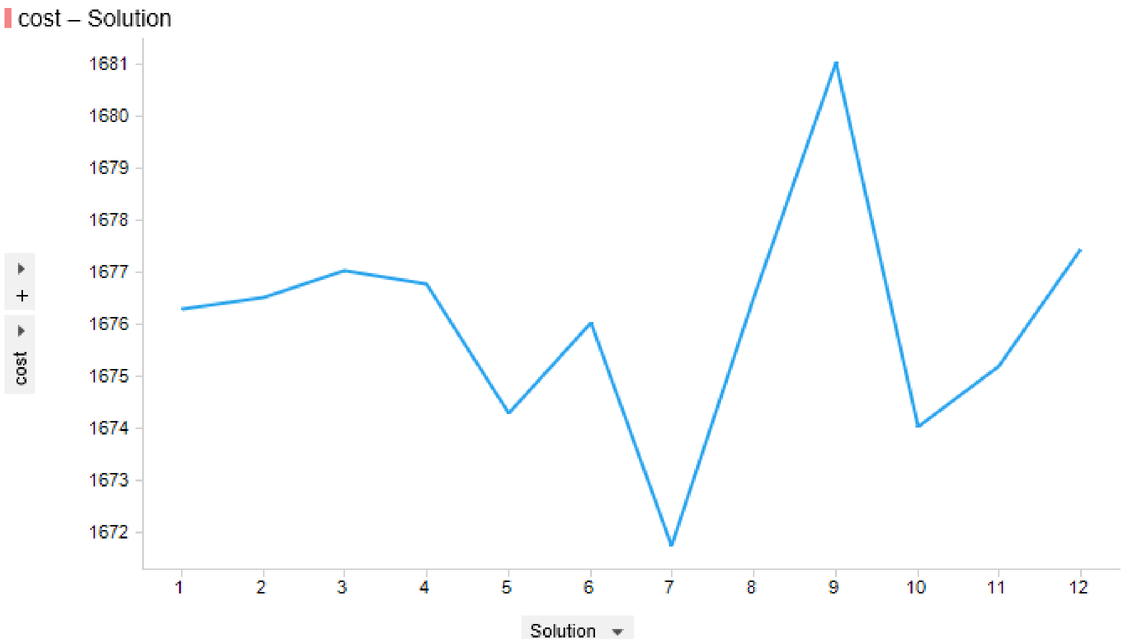


Fig. 16 Running the algorithm for 50 iterations without the Fisher-Yates shuffle

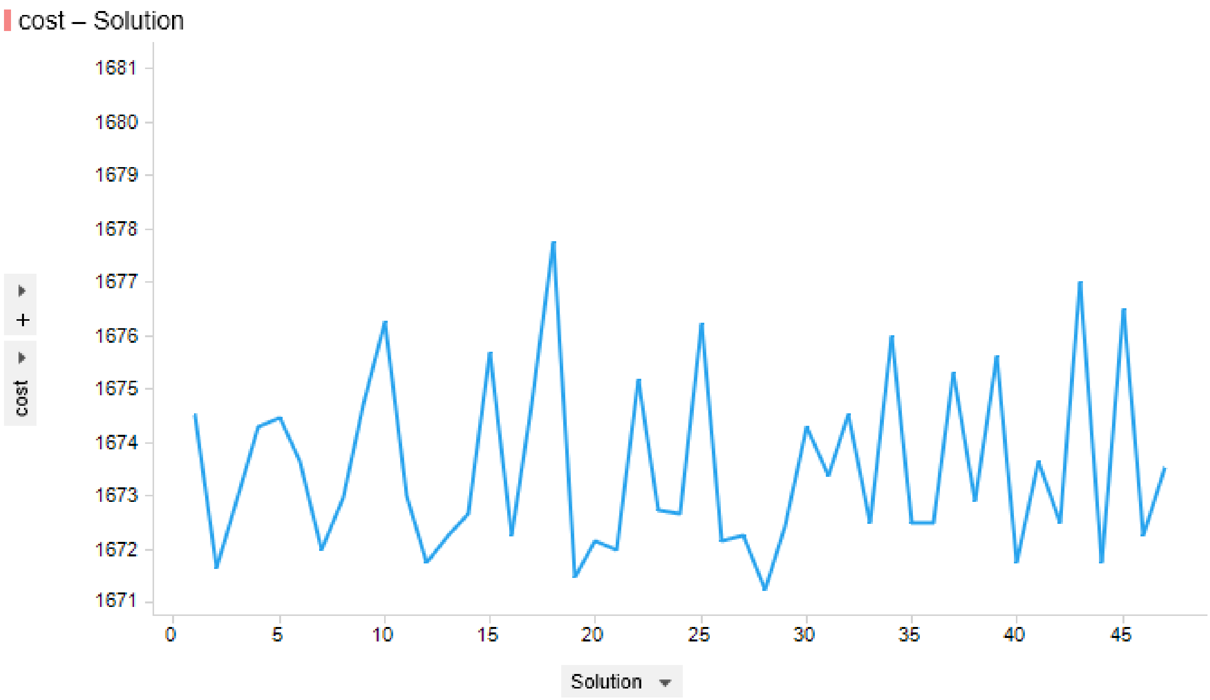


Fig. 17 Running the algorithm for 50 iterations with the Fisher-Yates algorithm

The tabu search algorithm we used is presented below. In Step 1 of this algorithm, the current status of the fleet schedule is accepted as the initial solution. This is the situation in where a disruption occurs that results in (propagated) delays or cancellations in the flight schedule. The aircraft routings in this initial solution are stored in the set F_{sk}. The set N in Step 3 will not contain any flights in the beginning (it makes little sense to reassign already cancelled flights), but will contain possible cancelled or unassigned flights over the next few iterations of the algorithm.

<u>**Algorithm 3:** Tabu Search</u>

1. *Start with a feasible solution, s, with an objective value of z*
2. *Let F_{sk} be the set of flights assigned to aircraft k in solution s*
3. *Let N be the set of flights not assigned to any aircraft*
4. ***For** a certain number of iterations {*
5.   *Let z^{temp} = large positive value, $N^{emp} = N$, $s^{temp} = s$*
6.   *Let P = the set of unique aircraft pairs in s*
7.   *Shuffle P*
8.   ***While** $P \neq \emptyset$ {*
9.   *Select an aircraft pair in P: aircraft a and b*
10.   *P = P – (a,b)*
11. $N^{temp} = N^{temp} \cup F_{sa} \cup F_{sb}$
12.   *Find all feasible routes for flights in N^{temp} using the combination of aircraft a and b*
13.   *Select the combination of routes for aircraft a and b that has the cheapest cost*
14.   *value associated with it*
15.   *Let s_{new} be the solution with the two new routes, z_{new} the value of this solution*
16.   *and N_{new} the set of cancelled flights in this solution*
17.   ***If** solution not tabu **and** $z_{new} < z^{temp}$ {*
18.   *Add solution to tabu list*
19. $s^{temp} = s_{new}, z^{temp} = z_{new}, N^{temp} = N_{new}$

```
20.         }
21.     }
22. }
```

The main search starts in Step 4 where an iterative search is started that runs for a fixed number of iterations. In each iteration, the neighbourhood of the current solution, *s*, is explored and the best solution stored; this takes place between Step 8 and Step 18. Before starting the while loop the aircraft pairs in P are shuffled using a Fisher-Yates shuffle. Then, in the while loop, a pair of aircraft is selected from all unique aircraft pairs in *P*. All the flights that are yet to be flown added to a pool with the cancelled flights (Steps 8-10). Then, in Step 11, a nested recursive tree build algorithm is used to find all possible routes for the aircraft in the selected pair, assuming the aircraft can only fly the flights in the set N^{temp}. In Step 12, a combination of routes is selected that has the least associated cost with it.

In the tree build algorithm, described in Algorithm 4, connecting flights have to be feasible from connection perspective. This means that a flight, Flight 1, can connect to another flight, Flight 2, if the destination of Flight 1 is the departure airport of Flight 2. In addition, there should be enough time between the arrival of Flight 1 and departure of Flight 2 to be able to carry out the arrival and departure services (minimum turnaround time). If Flight 2 needs to be delayed to be able to meet this requirement, a minimum delay should be used. Also, the last flight at the end of the recovery period should connect to the original schedule. That way the original schedule can be followed again after the recovery period.

Although not restricted, unnecessary swaps and swaps between aircraft types are discouraged. This is simply because each swap has to be communicated to the different parties involved during operations (e.g. crew, ground handlers) and this creates an extra workload, something that has to be avoided if possible. A more detailed description of how this is prevented is outlined in 7.1.3.2.

Algorithm 4: Recursive Nested tree build algorithm

```
1.  Start with a flight leg flight1, L as an empty list of routes, N as the set with unassigned flight legs,
2.  U as a list of flight legs that are either being flown, or the last flown flight leg and T as the minimum
3.  turnaround time
4.  Let Q be all flights in N that can be scheduled after flight1
5.  If Q ≠ ∅ {
6.          For each flight2 in Q {
7.          Determine earliest departure time for flight2 while respecting T
8.          Remove flight2 from N
9.          Let R be an empty route
10.         Add flight2 to R
11.         Add R to L
12.         Store this potential solution
13.         Start the nested tree build algorithm again with flight2
14.  } Else {
15.         Remove the first flight in U
16.         If U ≠ ∅{
17.         Start the nested tree build algorithm again with the second flight in U
18.         }
19. }
```

In Step 14 of Algorithm 13, the selected solution is checked for tabu status. If the solution is not tabu, the new found solution is checked against the current best neighbour candidate and saved if it is better (i.e. has a lower cost value). Without the tabu list there is no stochasticity and the algorithm will most likely always converge to a single solution. If a solution is tabu, the algorithm will skip the Steps 14-18 and move on to the next pair in P without modifying the current solution.

The neighbourhood in this context is defined as all solutions that can be reached by picking a unique aircraft pair and finding new routes for them. This means that with n number of aircraft, there are $\frac{n(n-1)}{2}$ number of solutions in the neighbourhood. To avoid long computation runtimes, only the lowest cost combination of routes is selected and tested against the current best solution.

7.1.3.2 Cost function

The cost function used to evaluate solutions consists of two components: estimated costs and a weighing factor. The estimated costs are based on information provided by the TFM tool and it has a 'hard' and 'soft' component. The 'hard' component are the actual costs that stem from delays and cancellations, i.e. reimbursements, re-bookings, hotel and meal costs, passenger compensation for excessive delays as mandated by European law. The 'soft' component consists of the costs that are associated with loss of future value, e.g. likelihood of returning as a customer, damaging word of mouth marketing. Both in the 'hard' and 'soft' components the costs are evaluated by passenger value, which is based on passenger class. This is not the actual cabin class or determined by the actual ticket price, but more so on the importance of the customer derived from their Flying Blue status, KLM's loyalty programme.

The weighing factors $C, D, Cost_{swap}\ and\ S$ are there to restrict unnecessary changes to the flight schedule and are used as weights to evaluate the solution. Setting high weights to a certain characteristic, for example on subtype changes, forces the application to obtain solutions with fewer (or none) subtype changes.

The cost of a solution is then evaluated by summing over all flights n and get the cost of each flight F:

$$\sum_{i=1}^{n} F_i^{cost}, where:$$

$$F_i^{cost} = C \cdot Cost_{cancellation}\ if\ flight\ is\ cancelled$$
$$F_i^{cost} = D \cdot Cost_{delay} + Cost_{swap} + S \cdot Cost_{subswap}\ if\ flight\ is\ not\ cancelled$$

where: $Cost_{cancellation}$ and $Cost_{delay}$ are obtained from TFM, $Cost_{swap}$ is a weight selected to prohibit unnecessary swaps, and $Cost_{subswap}$ is either a penalty (e.g. € 300 pp) for leaving passengers behind when downsizing an aircraft, or a standard penalty if no passengers are affected, S is then also a weight selected to prohibit subtype swaps.

8 Computational results

The purpose of providing computational results is not only to present the efficiency of the model but also to show how the decision support system can be used. All presented cases are selected by the manager operations control. However, before we present them we will first discuss the time complexity of the algorithm.

8.1 Time complexity

When we examine Algorithm 3 and 4 in the previous chapter, we find that the runtime mainly depends on the number of iterations it has to run for (line 4), the number of aircraft pairs (line 8) and the size of N^{temp} (line 11 and Algorithm 4). The number of aircraft considered has a quadratic influence on the solution time; after all, the amount of aircraft pairs possible is $\frac{n(n-1)}{2}$. The while loop has to run for each iteration, thus a first indication of the time complexity is the amount of iterations multiplied by the number of aircraft squared. Finally, we note that for each additional flight in N^{temp}, Algorithm 4 has to be called more than once recursively. The size of N^{temp} and consequently the duration of the recovery period have an exponential impact. Thus, the time complexity of the implemented tabu search algorithm can be described as $T(n) = \theta((mn^2)^t)$, where m is the amount of iterations the tabu search runs for, n is the number of aircraft considered in the problem and t is the duration of the recovery period in hours.

We will demonstrate this by changing for one variable while keeping the other two constant.

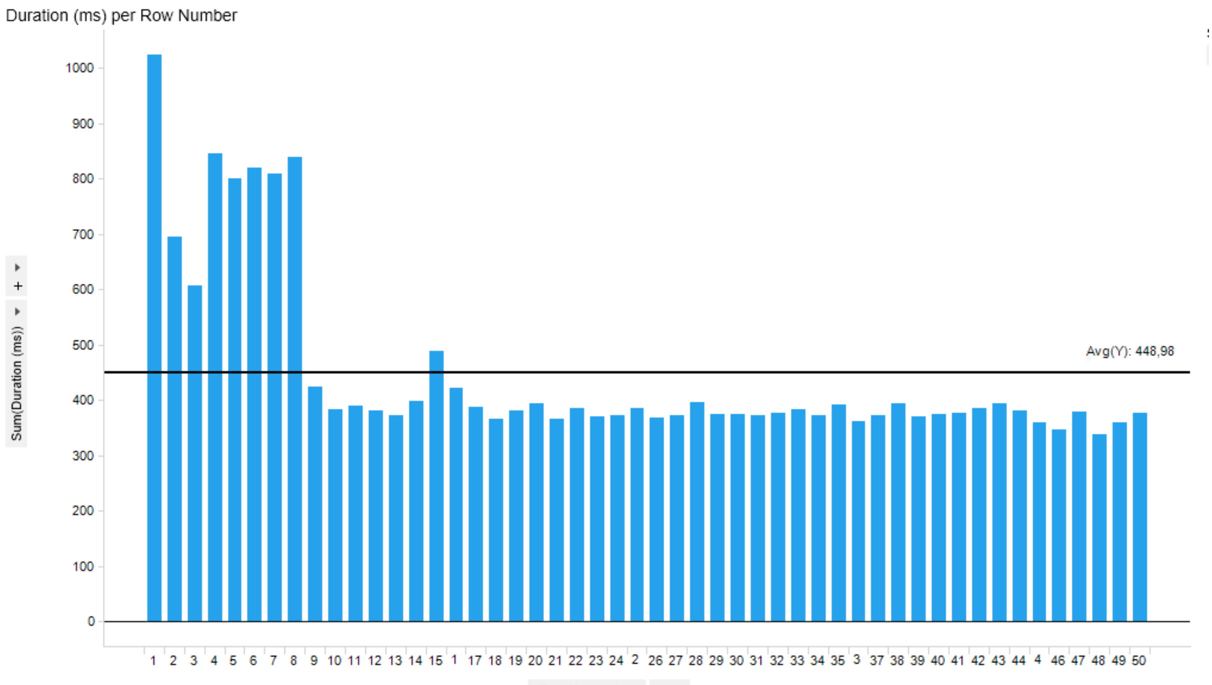


Fig. 18 Test run for 50 iterations and 44 aircraft, total runtime 22.5 s

8.1.1 Iterations

In Fig. 16, we have a test run for 50 iterations with a total runtime of 22.5 seconds. The average CPU time per iteration is 0.45 seconds. We see that after an initial start-up phase, the CPU time per iteration converges to about 0.38 s.

In Fig. 17, we run the same case for 25 iterations. Now, we get a total runtime of 14.82 s and an average CPU time per iteration of 0.593 seconds. It again has an initial start-up phase and converges to about 0.4 s per iteration. In Fig. 18, we do the same for 10 iterations. We notice that as the amount of iterations increases the average CPU time per iteration converges to approximately 0.4 s. Here, we notice that the amount of iterations and CPU time have a linear relationship.

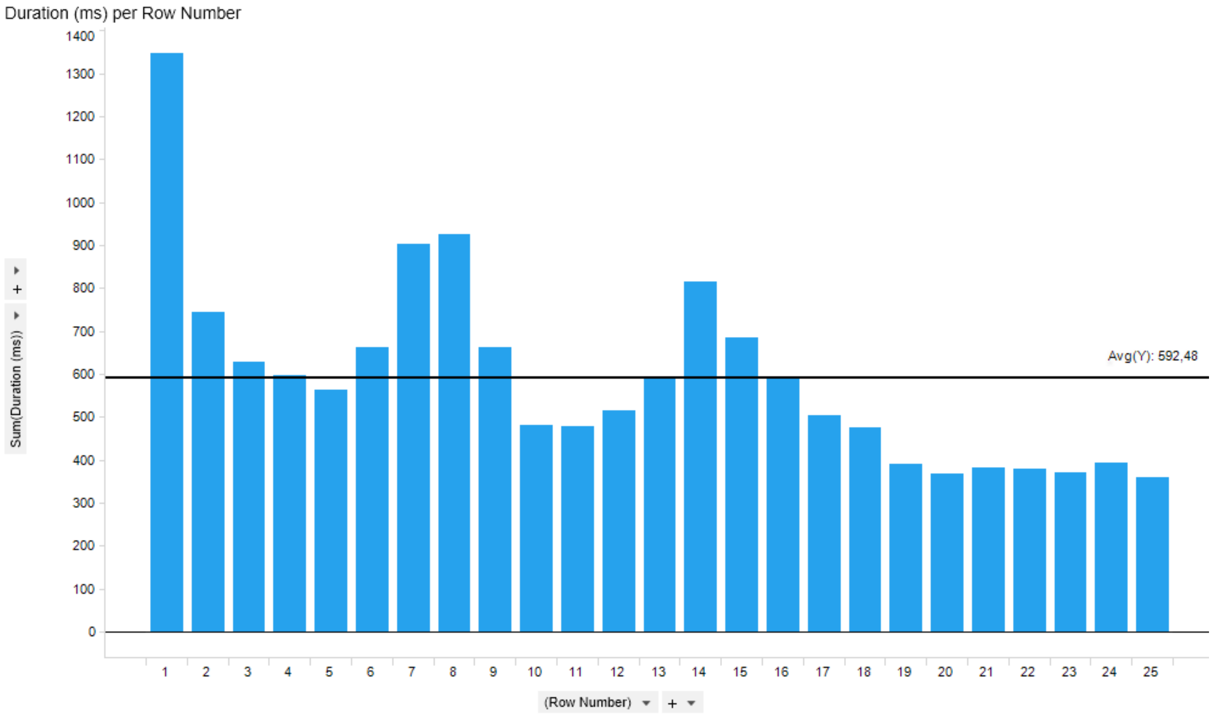


Fig. 19 Same case run for 25 iterations 14.8 s

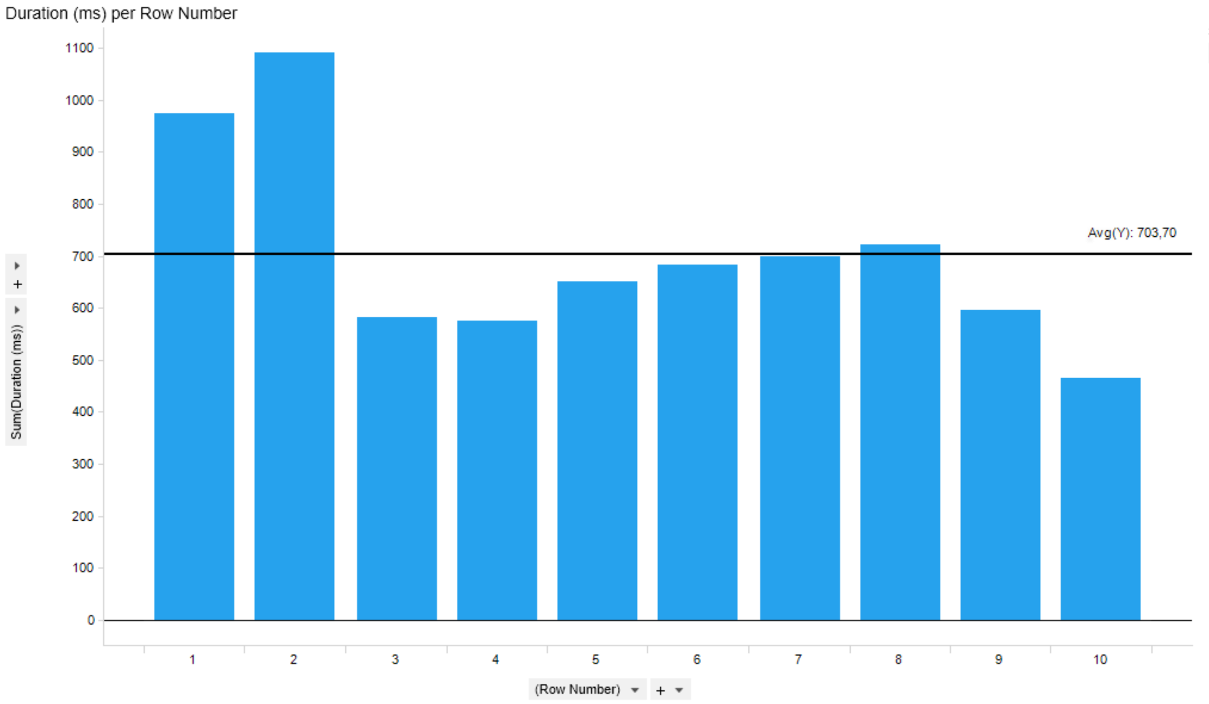


Fig. 20: Same case for 10 iterations, total runtime 7 s

8.1.2 The recovery period

The length of the recovery period has an exponential impact on the total CPU time. The cases in Fig. 16-18 were run from 12:45 UTC till the end of the day, a recovery period of approx. 7 hours. In Fig. 19 the average runtime for also 50 iterations, but with a recovery period of 11 hours, is 2.16 s. The average CPU time per iteration increases by approximately 5 times. Similarly, when the same case is run for a recovery period of 3 hours, the average CPU time per iteration decreases to 0.22 seconds. That is a reduction of half compared to Fig 16. The recovery period – and consequently the amount of flights included in the iteration – has thus an exponential impact on the CPU time.

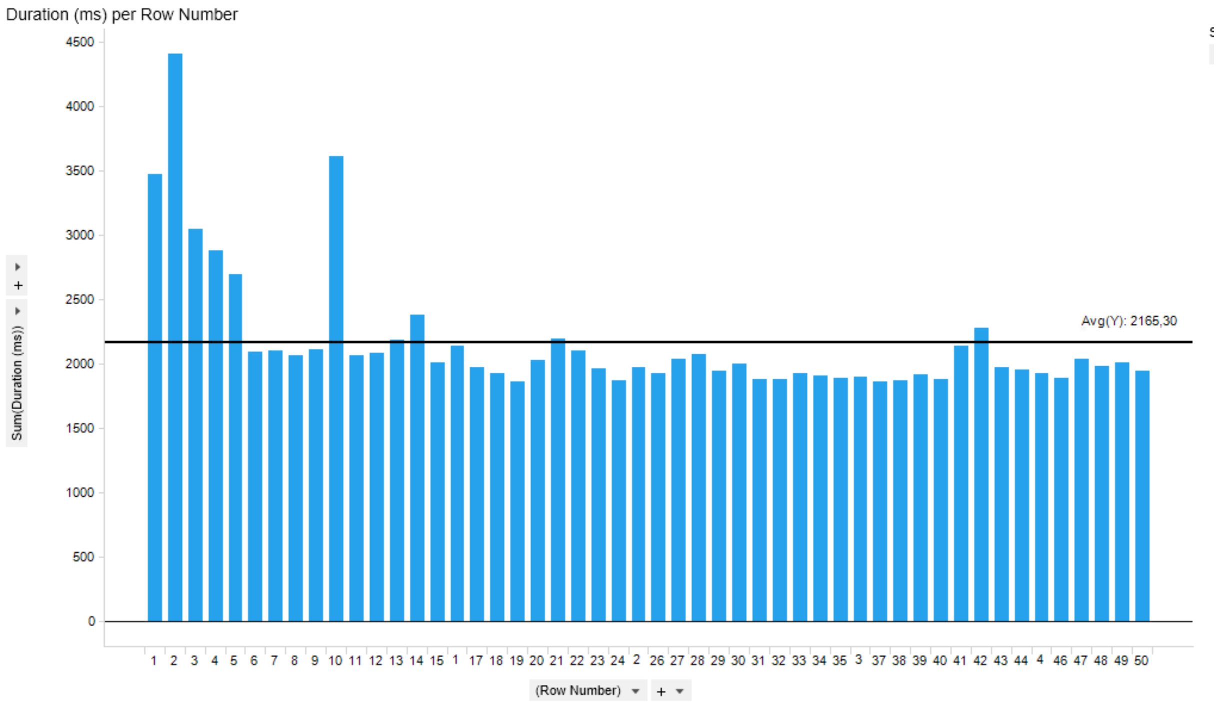


Fig. 21 CPU time for 50 iterations at 08:45, total runtime 108.3 s

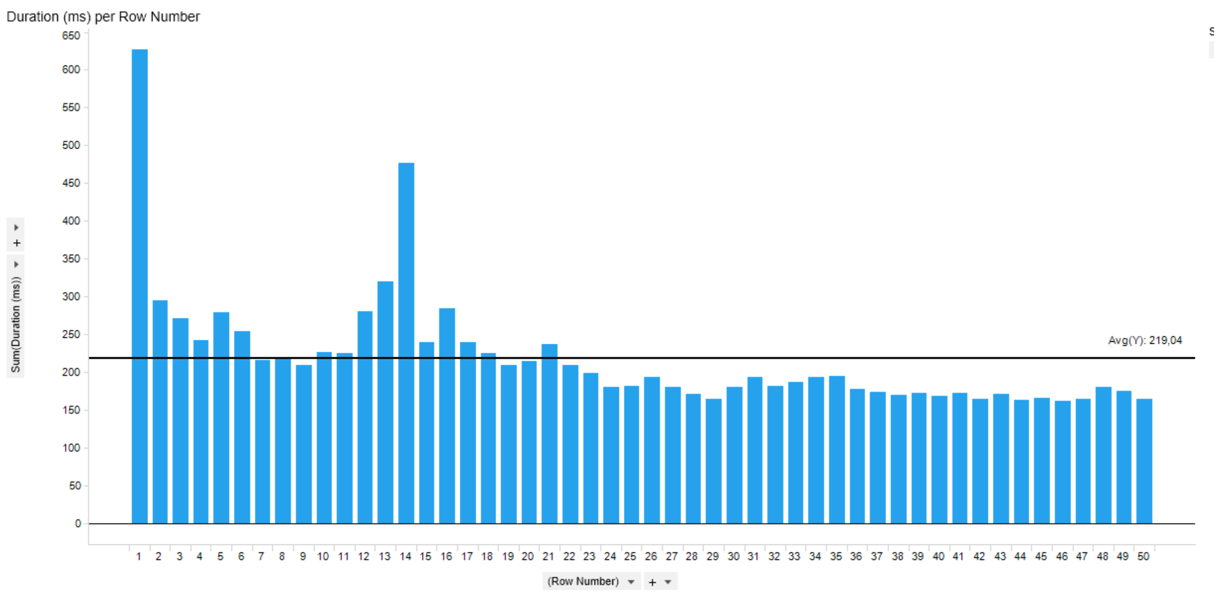


Fig. 22 CPU time for 50 iterations at 16:45, total runtime 10.95 s

8.1.3 Number of aircraft

Because of the pairs constructed, the number of aircraft considered has a quadratic influence on the solution time. The amount of aircraft pairs possible is after all $\frac{n(n-1)}{2}$. Fig 20, shows the same case as Fig. 16, but with 30 aircraft instead of 44. The total runtime is 10 seconds for this test, whilst it was 22.5 seconds for Fig. 16. We note that the amount of aircraft is reduced by 32% which should lead to an approximate runtime of $(1-0.32)^2 = 54\%$ less than 22.5 s. If we check this, we find that the total runtime for Fig 20 is 56% less than the case in Fig. 16.

Fig. 21 indicates similar results. The amount of aircraft in this test is down by 31 (or 70%) to 13. By approximation the runtime should be $(1-0.7)^2 = 91\%$ less than the total runtime for Fig 16, thus 2 s. In Fig. 21 we found that the total runtime is 1.48 s. The amount of aircraft has thus a quadratic impact on the total CPU time.

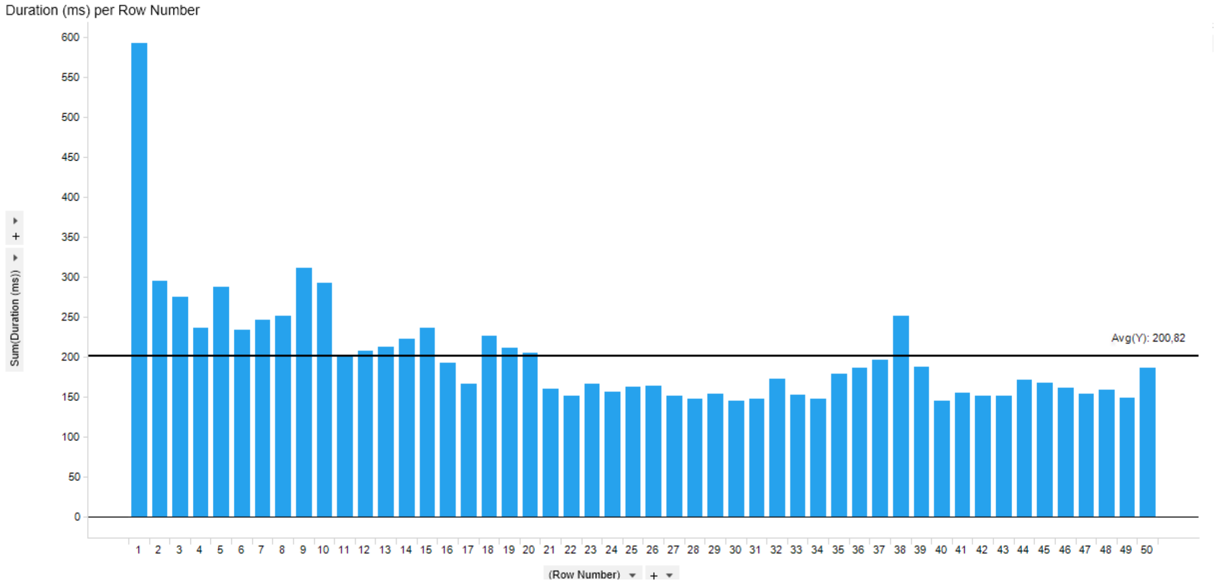


Fig. 23 Same case as in Fig. 15 but with 30 aircraft instead of 44, total runtime 10s

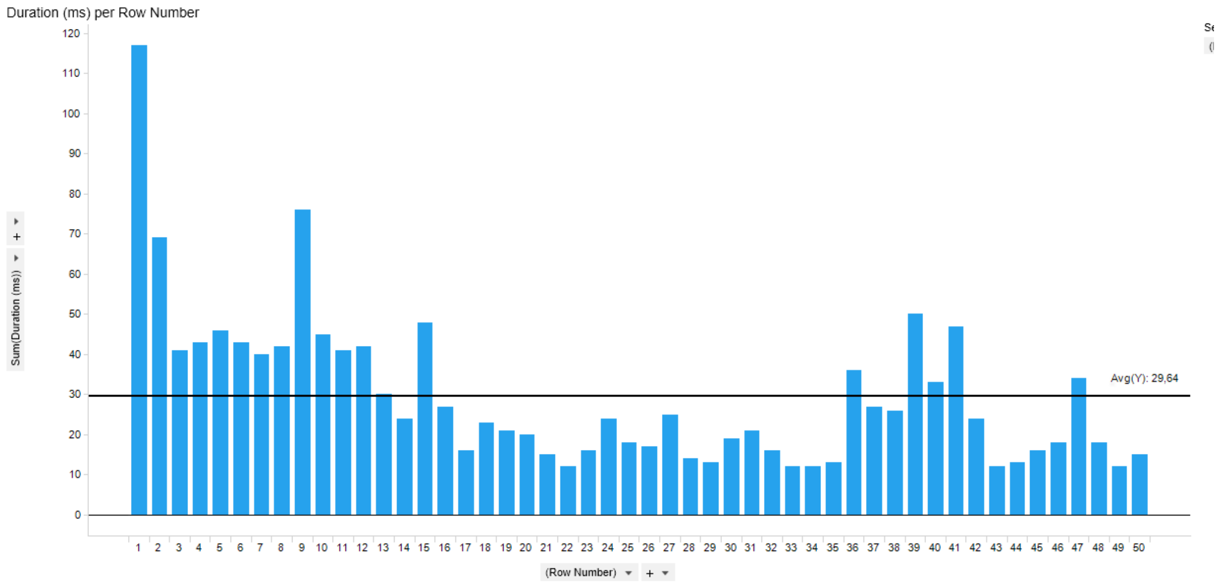


Fig. 24 Same case as in Fig. 15, but with 13 aircraft instead of 44, total runtime 1.48 s

8.2 Computational tests

The cases presented here will involve realistic disruptions that could happen in the daily operation. We will demonstrate how the DSS could be used to find solutions to a problem. A solution is a revised aircraft schedule, where flights may be delayed, cancelled or swapped between aircraft.

In our computational tests, the maximum delay allowed is 90 minutes, for longer delays it will be more cost efficient to cancel a flight. The recovery period is from a certain moment on the day until the end of that day, which is around 21:00 UTC. We will make use of actual flight schedules of KLC.

Cases will be presented as an initial situation in which the disruption occurs, and as solutions that are found for that situation. These will be depicted in a table using the following characteristics, affected passengers, aircraft swaps, subtype swaps, delay minutes, delays, cancellations and cost. Affected passengers are passengers affected by either cancellations or downsizing of aircraft. Aircraft swaps are swaps within the same fleet type. Subtype changes are swaps across subtypes. Delay minutes is the total amount of (propagated) delay in a schedule. Delays is the number of delays. Cancellations represents the number of cancellations and cost is the cost associated with this situation/solution in thousands of euros.

One of the solutions will then be plotted to show the aircraft schedule. We will also demonstrate how the obtained solutions can be manipulated by using weights. The algorithm is implemented in Java and the tests are done on a laptop with an AMD A8-4500M CPU that is clocked at 1.9 GHz and equipped with 8 GB RAM.

8.2.1 Case 1

Table 3 The dataset in case 1

Flights	296
Aircraft	44
Subtypes	3
Airports	52
Maintenance slots	16
Delayed aircraft	1
Aircraft on ground	0

Table 3 shows the characteristics for this case. This case is run on a full day's schedule on May 19, 2016. The initial schedule is presented in Appendix B, Case 1. In this Gantt-chart, the blue colours are flights that are incoming, the grey colours are flight that are outgoing and the red colours indicate maintenance slots. Changes to the schedule will be depicted in green.

Here flight KL1548 on aircraft PHEZH from Leeds to AMS is delayed 60 minutes, which propagates to flights KL1159 (to Goteborg), KL1160 (to AMS) and KL1173 (to Nice). The characteristics of this initial situation are as follows:

Solution	*Affected passengers*	*Aircraft swaps*	*Subtype swaps*	*Delay minutes*	*Delays*	*Cancellations*	*Cost (x 1000)*
-	0	0	0	230	4	0	32.7

The 60 minute delay propagates to three more flights, leading for a total delay of 240 minutes. The associated cost is € 32,700.

This case is then run for 50 iterations (37 seconds), where 47 solutions are found. Here we selected the top 10 solutions with the least cost. The full list is presented unsorted in Appendix B Case 1. The commercial data used contains delay costs for only incoming flights as this is where passenger itineraries may get interrupted. Thus, delaying an outgoing flight is (for now) not associated with any cost. Aircraft swaps have a penalty cost of € 250 and subtype swaps have a penalty cost of either $Affected\ passengers \cdot € \ 300$ or a fixed cost of € 250 if $Affected\ passengers$ is zero. Additionally, to prevent cancellations, the cost of cancellation is multiplied with a weight of 100. These are the default parameters.

During the case, flight KL1548 is in progress and cannot be swapped or cancelled. Thus the best case scenario would be a flight schedule with still 60 minutes of delay in it.

Table 4 Top 10 solutions with least cost

Solution	*Affected Passengers*	*Aircraft swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
26	0	4	0	60	1	0	10,5
25	0	5	0	60	1	0	10,75
24	0	6	0	60	1	0	11
29	0	2	4	60	1	0	11
46	0	2	4	65	2	0	11
10	0	6	0	75	2	0	11
31	0	6	0	80	3	0	11
14	0	6	0	90	2	0	11
17	0	6	0	90	4	0	11
35	0	6	1	90	2	0	11,25

When we analyse the solutions in Table 4, we note that the first four solutions all solve for the flights with propagated delay. The delayed flight in Solutions, 26, 25, 24 and 29 is flight KL1548, which was already in progress. The solutions have a slightly different cost value due to the penalty for aircraft swaps. Upon further inspection, we notice that some solutions have more delays and more delay minutes, yet the cost is very close to that of solutions with only one delay. Table 5, shows which flight are delayed in the corresponding solution. All the delayed flights are uneven flights, except for flight KL1548. Flights with uneven flight numbers are always outgoing flights in KLM's flight numbering system, which means that all these flights do not have any associated cost with them in our model. It is also interesting to note that there are three solutions, 29, 46 and 35, that incorporate subtype swaps, yet the amount of affected passengers remains 0 for all three solutions. This means that no passenger were affected by the downsizing flights from Embraer to Fokker.

Table 5 The delayed flights in first four solutions of Table 4

Solution	*Delayed flights*
26,25,24,29	KL1548
46	KL0919, KL1548
10	KL1319, KL1548
31	KL1289, KL1319, KL1548
14, 35	KL1273, KL1548
17	KL1289, KL1319, KL1548, KL1879

Now, an operator may prefer solutions where all delayed flights – be it incoming or outgoing – are also penalised and subtype swaps are not desired. Therefore, we run this case again and penalise, subtype swaps by multiplying the parameter with 100 and add a delay cost of € 100 per minute delay. Table 6 shows these solutions with least cost with these parameters.

When we analyse the solutions this time around, we note that there aren't any solutions with more than 73 minutes delay in the top 10 as opposed to solutions with 80 or 90 minutes in Table 4. The DSS also still finds solutions that contain subtype swaps, but these are now at the bottom of the list due to the high penalties. These solutions also have a higher cost value than the initial situation. We could prevent this by only accepting solutions with a lower value than the initial situation, thus change the parameters z^{temp} from a large positive value to the value of the initial situation in algorithm 3 in Chapter 7.1.3. However, this imposes heavy restrictions on the amount of different solutions that can be obtained as we will demonstrate next.

Table 6 All solutions with changed parameters sorted on cost

Solution	*Affected Passengers*	*Aircraft swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
14	0	4	0	60	1	0	10,5
13	0	6	0	60	1	0	11
18	0	7	0	60	1	0	11,25
16	0	8	0	60	1	0	11,5
12	0	6	0	66	3	0	11,6
11	0	8	0	62	2	0	11,7
19	0	9	0	60	1	0	11,75
20	0	10	0	60	1	0	12
35	0	11	0	60	1	0	12,25
10	0	10	0	73	2	0	12,6
40	0	4	0	70	3	0	12,65
34	0	13	0	60	1	0	12,75
9	0	11	0	73	2	0	12,85
29	0	15	0	60	1	0	13,25
1	0	12	0	68	4	0	13,3
8	0	13	0	73	2	0	13,35
30	0	4	0	90	2	0	13,5
7	0	12	0	79	4	0	13,7
32	0	6	0	90	2	0	14
31	0	4	0	103	3	0	14,1
26	0	8	0	90	2	0	14,5
24	0	8	0	70	4	0	14,74
36	0	22	0	60	1	0	15
23	0	10	0	90	2	0	15
37	0	8	0	96	4	0	15,1
21	0	11	0	90	2	0	15,25
38	0	9	0	96	4	0	15,35
27	0	12	0	90	2	0	15,5
41	0	6	0	113	6	0	25,89
33	0	13	1	60	1	0	37,75
6	0	6	1	90	2	0	39
3	0	13	2	90	2	0	65,75
17	0	8	4	60	1	0	111,5
28	0	12	4	62	2	0	112,7
39	0	9	5	60	1	0	136,75
25	7	10	4	68	3	0	299,28
2	7	16	4	78	5	0	303,08
5	7	11	5	90	2	0	325,25
4	11	11	5	60	1	0	417,25

15	18	4	4	60	1	0	625,5
22	30	7	4	60	1	0	961,25

Table 7, contains all solutions found when z^{temp} is initialised to the value of the initial situation. With this setting changed, the DSS is only capable of finding 13 solutions and what perhaps is more interesting, is that the DSS is not able to find the global optimum with respect to cost. Solution 14 from Table 6 and Solution 26 from Table 4, which share the same characteristics, are nowhere to be found. Therefore, we initialise z^{temp} to a large positive value as it helps to obtain a large pool of diverse solutions and with a higher chance of also obtained the global optimum.

Table 7

Solution	*affectedPax*	*regSwaps*	*subSwaps*	*delayMinutes*	*delays*	*cancellations*	*cost*
1	0	6	0	60	1	0	11
2	0	8	0	60	1	0	11,5
3	0	7	0	60	1	0	11,25
4	0	8	0	90	2	0	14,5
5	0	4	0	75	2	0	12
6	0	12	0	90	2	0	15,5
7	0	10	0	97	3	0	15,7
8	0	4	0	90	2	0	13,5
9	0	6	0	90	2	0	14
10	0	7	0	170	4	0	29,85
11	0	6	0	110	3	0	26,65
12	0	0	0	230	4	0	43,7
13	0	8	0	62	2	0	11,7

Solution 26 from Table 4 is depicted in Appendix B, Case 1. This solution contains four aircraft swaps (depicted in green), a 60 minutes delay and a total cost value of € 11,000. The four aircraft swaps means that four flights are assigned to a different aircraft. In this solution, flights KL1159 and KL1160 that were originally delayed on aircraft PHEZH are assigned to aircraft PHEXC. The flights KL1549 and KL1550 that were originally scheduled on aircraft PHEXC are assigned to aircraft PHEZE where there was some slack. The only delayed flight is flight KL1548 that was in progress.

The manager operations control verified this as an acceptable and logical solution. Swapping the delayed flights to any other Embraer aircraft would have resulted in more delay than 60 minutes as there is very little slack in the schedule. And swapping to a Fokker aircraft would either affect passengers or more swaps need to be done to prevent delays, as can be seen in Table 4, solution 29.

8.2.2 Case 2

In case 2, we use the same flight schedule as in case 1, but this time without all the Fokker aircraft. Also, instead of a delay, we will deal with two aircraft on ground (AOG) situations, see Table 8 for the data set and Appendix A, Case 2 for the initial situation.

Table 8 Dataset for case 2

Flights	198
Aircraft	30

Subtypes	2
Airports	45
Maintenance slots	13
Delayed aircraft	0
Aircraft on ground	2

In this case, flights KL1271, KL1272 on aircraft PHEZV and flights KL1289, KL1290 and KL1879 on aircraft PHEZP are 'cancelled', i.e. unassigned because they cannot be operated anymore. This is depicted in Appendix A, Case 2 with the colour green. The characteristics of this case is as follows:

Solution	*Affected passengers*	*Aircraft swaps*	*Subtype swaps*	*Delay minutes*	*Delays*	*Cancellations*	*Cost (x 1000)*
-	340	0	0	0	0	5	3515

The four 'cancelled' flights impact 340 passengers in total. The cost value of € 3,515,000 is skewed as solutions with cancellations are multiplied with a weight of 100. So the actual cost would be € 35,150.

This case is again run for 50 iterations (32 seconds), where 36 solutions are found. Here we selected the top 10 solutions with the least cost. The full list is presented unsorted in Appendix A, Case 2. All parameters are the same as in the previous case, but an additional penalty of € 100 per delay minute is invoked on outgoing flights.

Table 9 Top 10 solutions with least cost for case 2

Solution	*Affected Passengers*	*Aircraft Swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
19	356	9	0	0	0	4	1672,25
4	356	9	0	1	1	4	1672,49
5	356	7	0	10	1	4	1672,75
32	356	12	0	3	1	4	1673,42
31	356	14	0	0	0	4	1673,5
25	356	7	0	13	2	4	1673,89
14	356	11	0	11	2	4	1673,99
13	356	12	0	3	1	4	1674,14
11	356	15	0	13	2	4	1675,89
12	348	7	0	0	0	4	1676,75

When we analyse the solutions in Table 4, we find that all solutions contain 4 cancellations but are a factor 2 less than the cost for the initial situation. For all solutions, except solution 12, the cancelled flights are KL1023, KL1024, KL1289 and KL1290. For solution 12, the cancelled flights are KL1023, KL1032, KL1289 and KL1290. Flights KL1024 and KL1032 are both incoming flights from London, but are scheduled on different times and on different aircraft. However, both solutions are valid solutions, meaning no aircraft will have to remain on an outstation due to a cancellation on an incoming flight. The reason why these flights are consequently cancelled is because they are the cheapest to do so. Upon inspecting the initial situation in Appendix A, Case 2, we note that it is impossible to assign all 'cancelled' flights as there is barely any slack in the schedule. Another interesting thing to note is that solution 12 affects fewer passengers than any other solution in the table yet is associated with the highest cost. This is because cancelling flight KL1032 is slightly more expensive than cancelling flight KL1024, even though flight KL1032 has fewer passengers than KL1024, 95 and 87 passengers respectively. The difference in value can be explained by a difference in perceived passenger value/class.

Table 4 also gives an idea as to why a diversity of solutions is more preferred than a single solution with the lowest cost. When we compare solution 19 to solution 12 for example, we note that though solution 12 is costlier, it can be accomplished with fewer swaps and also impacts fewer passengers. The fleet schedule corresponding to solution 12 is plotted in Appendix A, Case 2.

This solution was also discussed with the manager operations control. Whilst the discussed solution is acceptable within the confinements of the available information, the solution is somewhat impractical. So far, in our DSS, we have worked with daily schedules where no information is available on the end position of the aircraft, i.e. where the aircraft should be located after the final leg of the day so to resume the normal fleet schedule on the following day. Therefore, the DSS does not consider yet the end position of an aircraft as a requirement. That being said, the solution is definitely an acceptable solution when taken into account information about the duration of the recovery period is omitted.

8.2.3 Case 3

In Case 3 we use a different flight schedule as compared to cases 1 and 2. We also compare one of the solutions found by the DSS with that of an experienced fleet controllers. The problem size for this case is given in Table 10.

Table 10 Case 3

Flights	302
Aircraft	45
Subtypes	3
Airports	52
Maintenance slots	30
Delayed aircraft	0
Aircraft on ground	3

In this case we have three aircraft with technical defect that have to go unexpected maintenance. Aircraft PHEZG, PHEZW and PHKZD will all be unavailable until 2300h after they touched down on AMS. The start moment for this case is 09:30 UTC, during which aircraft PHEZG and PHEZW are operating a flight leg and aircraft PHKZD is in a maintenance slot. The weights used in this case are 10 for D and S in $D \cdot Cost_{delay}$ and $S \cdot Cost_{subswap}$ respectively and 100 for C in $C \cdot Cost_{cancellation}$.

The initial situation is depicted in Appendix B, Case 3. The characteristics are given in Table 11.

Solution	*Affected passengers*	*Aircraft swaps*	*Subtype swaps*	*Delay minutes*	*Delays*	*Cancellations*	*Cost (x 1000)*
-	392	0	0	0	0	5	6500

The four 'cancelled' flights impact 392 passengers in total. The cost value of € 6,500,500 is again skewed as solutions with cancellations are multiplied with a weight of 100. So the actual cost would be € 65,000. The affected flights are KL1031, KL1032, KL1183, KL1771 and KL1772.

Table 11 Top 10 solutions with least cost for case 3

Solution	*Affected Passengers*	*Aircraft Swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
5	0	10	0	21	1	0	39.6
49	0	13	0	21	1	0	40.35
29	0	13	0	26	2	0	45.35

30	126	11	6	20	2	2	47.75
11	126	12	1	26	2	2	57.6
22	5	14	7	21	1	0	68.1
50	127	12	5	26	2	2	68.1
35	126	14	10	31	3	2	69.2
33	127	17	5	26	2	2	69.35
20	0	18	8	33	3	0	73.6

A side note for this case and the solutions presented in Table 11 is that there were some inconsistencies with the data. Because of that there is a delay of 21 minutes for a past flight which is not correct. When we analyse the solutions we found that in the solution with the least cost (and many others) all flights are assigned. We note it is possible to assign all flights without delaying any other and without conducting subtype swaps. In Appendix B, Case 3 we depicted Solution 5.

We have also given this case to an expert fleet controller whom has found a solution with the following characteristics:

Solution	*Affected passengers*	*Aircraft swaps*	*Subtype swaps*	*Delay minutes*	*Delays*	*Cancellations*	*Cost (x 1000)*
User	0	11	0	0	0	0	N/A

This solution, depicted in Appendix B, Case 3, is very similar to solution 5 (when we ignore the incorrect value for cost and delays). A solution that can be reached in 10 aircraft swaps.

In the expert user solution, the 11 aircraft swaps are as follows:

Newly assigned Aircraft	*Flights*
PHEZV	KL1095, KL1096
PHEZY	KL1731, KL1732, KL1183
PHEXB	KL1771, KL1772
PHKZA	KL1027, KL1028
PHKZL	KL1031, KL1032

In solution 5, the10 aircraft swaps are as follows:

Newly assigned Aircraft	*Flights*
PHEZV	KL1731, KL1732, KL1183
PHEZY	KL1217, KL1218
PHEZF	KL1781, KL1782
PHKZA	KL1027, KL1032
PHKZL	KL1031

The solutions are more or less similar. The explanation as to why solution 5 contains one fewer swap is because the DSS suggest breaking apart the flight rotation KL1027-KL1028 to London. This is something that is typically not done in practice because it brings complications to the crew duties. In solution 5, the crew members operating the flight KL1027 to London will have to fly back on flight KL1031, which departs 3 hours later. These crew members had to fly back to AMS right after arriving in London on flight KL1027. Such crew restrictions were not considered by our DSS as we considered this out of scope for this study.

9 Conclusion

The quality of solutions in the existing daily disruption management process is heavily influenced by the skill and experience of the operators. Additionally, there is little to no transparency in the solution process. This project was thus initiated to find a way to improve upon this. It was found that currently not all required information is readily available or accessible during the control phase of the process control function. To address this problem we formulated the following research goal that this work aimed to answer:

"Design a decision support system that provides insight on the quality of possible solutions to a disruption by informing on the consequences thereof."

The designed DSS is a prototype that is able to find a multitude of possible solutions to a given situation. Insight on the quality of a solution is provided in specific characteristics. These are the number of resulting delays and the corresponding total delay minutes, the number of cancellations, the total cost associated with passenger inconvenience and the number of affected passengers. The characteristics number of aircraft swaps and subtype swaps are used to indicate how much effort it is to reach this outcome.

These solutions are then presented in a few sorted tables, which will assist the user in finding the best solution for a given situation. The algorithm used was adapted from literature with some minor but valuable improvements. Computational tests have shown the efficacy of the algorithm. In our experience a pool of solutions could be obtained in about 20 to 120 seconds depending on the input variables as number of iterations, length of the recovery period and the amount of aircraft. The inspected results in the test cases show that the obtained solutions are realistic and logical. However, within the pool of solutions only a few are interesting to analyse and the majority can be discarded, simply because they are suboptimal from a certain perspective (e.g. require too many (subtype) swaps). The obtained solutions are verified by both the manager Operations Control and a couple of senior fleet controllers.

The prototype shows promising results with respect to the performance criteria. The DSS is able to (1) find a diversity of solutions, (2) provide insight in the solution using certain characteristics, (3) find solutions within an acceptable CPU timeframe and (4) that are realistic.

Furthermore, the DSS meets all the selected requirements, as it is able to solve a problem from a fleet perspective, uses the TFM information as the basis of evaluation for passenger impact, it respects maintenance slots, is able to find solutions using aircraft swaps, subtype swaps, delays and cancellations, and within an acceptable timeframe.

We believe the prototype proved its utility as a DSS, as it provides insight on the quality of possible solutions by informing on (1) what actions need to be taken to achieve that outcome and (2) what the consequences then will be.

9.1 Recommendations and further research

Whilst we think the prototype is successful within the confinements of this research, there are some caveats. So far, we have only addressed the aircraft recovery problem without any crew restrictions. For a DSS to be truly successful, crew restrictions also need to be considered. Now, the DSS will suggest solutions that may be completely valid from a fleet perspective, but not practical when also accounted for any crew consequences (as is shown in one case). Hence, crew restrictions like latest time of arrival (LTA), obligatory crew ground time during aircraft changes, pilot restrictions in subtype swaps and aircraft-crew connections definitely need to be considered and added to the model.

Second, the crew recovery problem is still persistent even when accounted for crew restrictions. As stated in Chapter 6.3, the crew recovery problem can be approached in three ways: (1) by initiating the crew recovery problem to a fixed flight schedule, (2) incorporating decision variables that allow for cancelling flight legs or (3) allowing for flight delays. The first can be achieved in our approach by kicking off a separate crew recovery module for after each iteration to the aircraft recovery problem. Zhang et al. [31] propose an intriguing algorithm to allow for crew recovery with flight cancellations during the integrated fleet and crew recovery problem. They do this by first solving for the aircraft recovery problem with partial crew considerations followed by solving for the crew recovery problem with partial aircraft considerations. Further research is recommended to investigate how the crew recovery problem can be integrated in our model.

Furthermore, we have not yet added restrictions for the duration of the recovery period. This is an important feature. Primarily to keep the solution time acceptable but also to impose further restrictions in the form of an end position for the aircraft.

We have developed a prototype that is, in its current state, not yet implementable in the organisation. The prototype does not yet have an interface to the existing systems and neither does it have a graphical user interface (GUI). Obviously, to make the tool accessible, a GUI should be built for the DSS and an interface to the existing systems should be implemented, i.e. the Netline and TFM systems. This will help users to get familiar with the DSS and also to pinpoint any not yet foreseen shortcomings. Also, having an interface will allow for more advanced features like including or excluding flights from the solution process.

In our algorithm we assume all disruptions all known and will not change during our computation. This is of course not true. A new disruption can happen at any time and it is non-trivial that this can be included in the computation. Hence, it is interesting to investigate how a dynamic approach to disruptions can be incorporated in our static approach.

One of the objectives of the DSS is to help provide insight in the decision-making. This can be easily achieved by logging all decisions and possible alternatives. This way, a database can be filled to analyse previous decisions and find potential shortcomings in either the decision-making process or the DSS.

Our use of the tabu list in the tabu search algorithm is a somewhat primitive. Our tabu list contains a certain number of solutions that serves as a short-term memory, which is used to diversify the search. The use of more advanced tabu list memory structures may improve the quality of the obtained solutions. Further research is recommended to investigate what form of memory structure would serve our problem best.

Finally, our implementation has not been developed to make use of the extra computational power parallel computing provides. However, it should be possible to run multiple iterations in parallel for our model. Hence, an interesting subject for further research involves the use of parallel computing to speed-up the CPU time in such time-critical applications where each spent minute may have dire consequences on.

Bibliography

[1] M. E. Irrang, "Airline Irregular Operatoins," *Handb. Airl. Econ. .*, pp. 349–365, 1996.

[2] A. J. M. Castro and E. Oliveira, *A New Approach for Disruption Management in Airline Operations Control*. 2014.

[3] Y. Hu, B. Xu, J. F. Bard, H. Chi, and M. Gao, "Optimization of multi-fleet aircraft routing considering passenger transiting under airline disruption," *Comput. Ind. Eng.*, vol. 80, pp. 132–144, 2015.

[4] K. Sinclair, J.-F. Cordeau, and G. Laporte, "Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem," *Eur. J. Oper. Res.*, vol. 233, no. 1, pp. 234–245, 2014.

[5] K. F. Abdelghany, A. F. Abdelghany, and G. Ekollu, "An integrated decision support tool for airlines schedule recovery during irregular operations," *Eur. J. Oper. Res.*, vol. 185, no. 2, pp. 825–848, 2008.

[6] S. E. Karich, S. S. Altus, G. Stojkovic, and M. Stojkovic, *Quantitative Problem Solving Methods in the Airline Industry, A Modeling Methodology Handbook Series (International Series in Operations Research & Management Science)*. New York: Springer Science & Business Media, 2011.

[7] "KLM Cityhopper - Annual Report 2014," 2014.

[8] H. P. M. Veeke, J. A. Ottjes, and G. Lodewijks, *The Delft Systems Approach*, 1st ed., vol. 1. Springer London, 2008.

[9] D. Teodorović and S. Guberinić, "Optimal dispatching strategy on an airline network after a schedule perturbation," *Eur. J. Oper. Res.*, vol. 15, no. 2, pp. 178–182, Feb. 1984.

[10] J. Clausen, A. Larsen, J. Larsen, and N. J. Rezanova, "Disruption management in the airline industry-Concepts, models and methods," *Comput. Oper. Res.*, vol. 37, no. 5, pp. 809–821, 2010.

[11] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine, "Airline disruption management—Perspectives, experiences and outlook," *J. Air Transp. Manag.*, vol. 13, no. 3, pp. 149–162, 2007.

[12] J. D. Petersen, G. Solveling, J.-P. Clarke, E. L. Johnson, and S. Shebalov, "An Optimization Approach to Airline Integrated Recovery," *Transp. Sci.*, no. September 2015, pp. 1–34, 2012.

[13] T. Andersson, "Solving the flight perturbation problem with meta heuristics," *J. Heuristics*, vol.

12, no. 1–2, pp. 37–53, 2006.

[14] T. K. Liu, C. R. Jeng, Y. T. Liu, and J. Y. Tzeng, "Applications of Multi-objective Evolutionary Algorithm to Airline Disruption Management," *2006 IEEE Int. Conf. Syst. Man Cybern.*, pp. 4130–4135, 2006.

[15] T. K. Liu, Y. T. Liu, C. H. Chen, J. H. Chou, J. T. Tsai, and W. H. Ho, "Multi-objective optimization on robust airline schedule recover problem by using evolutionary computation," *2007 Ieee Int. Conf. Syst. Man Cybern. Vols 1-8*, pp. 2201–2206, 2007.

[16] T. K. Liu, C. H. Chen, and J. H. Chou, "Optimization of short-haul aircraft schedule recovery problems using a hybrid multiobjective genetic algorithm," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2307–2315, 2010.

[17] B. Zhu and J. Zhu, "Research on Mixed Set Programming for Aircraft Schedule Recovery."

[18] X. Zhao and Y. Guo, "An improved GRASP for Irregular Flight Recovery," 2012, no. 71073071, pp. 465–469.

[19] Z. Xiuli and G. Yanchi, "Study on GRAPS-ACO Algorithm for Irregular Flight Rescheduling," *2012 Int. Conf. Comput. Sci. Serv. Syst.*, no. 71073071, pp. 266–269, 2012.

[20] Y. Zhang and M. Hansen, "Real-Time Intermodal Substitution: Strategy for Airline Recovery from Schedule Perturbation and for Mitigation of Airport Congestion," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2052, no. 2052, pp. 90–99, Dec. 2008.

[21] N. Eggenberg, M. Salani, and M. Bierlaire, "Constraint-specific recovery network for solving airline recovery problems," *Comput. Oper. Res.*, vol. 37, no. 6, pp. 1014–1026, 2010.

[22] P. Arias, M. M. Mota, D. Guimarans, and G. Boosten, "A Methodology Combining Optimization and Simulation for Real Applications of the Stochastic Aircraft Recovery Problem," *2013 8th EUROSIM Congr. Model. Simul.*, pp. 265–270, 2013.

[23] Q. Lei, D. Jiang, P. Zhao, and T. Ma, "Research on the disrupted airline scheduling," in *2013 10th International Conference on Service Systems and Service Management*, 2013, pp. 332–336.

[24] B. Aguiar, J. Torres, and A. J. M. Castro, "Operational Problems Recovery in Airlines - A specialized Methodologies Approach," in *Progress in Artificial Intelligence*, 2011, no. October 10–13.

[25] H. Sousa, R. Teixeira, H. L. Cardoso, and E. Oliveira, "Airline Disruption Management," pp. 398–405.

[26] A. J. M. Castro and E. Oliveira, "Using Specialized Agents in a Distributed MAS to Solve Airline Operations Problems: A Case Study," *2007 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*,

pp. 473–476, 2007.

[27] R. Nissen and K. Haase, "Duty-period-based network model for crew rescheduling in European airlines," *J. Sched.*, vol. 9, pp. 255–278, 2006.

[28] C. P. Medard and N. Sawhney, "Airline crew scheduling from planning to operations," *Eur. J. Oper. Res.*, vol. 183, no. 3, pp. 1013–1027, 2007.

[29] A. J. M. Castro and E. Oliveira, "Solving Airline Operations Problems Using Specialized Agents in a Distributed Multi-Agent System," pp. 173–184, 2008.

[30] X. Zhao and J. Zhu, "Application of grey programming in irregular flights scheduling," *2007 IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 164–168, 2007.

[31] D. Zhang, H. Y. K. Henry Lau, and C. Yu, "A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems," *Comput. Ind. Eng.*, vol. 87, pp. 436–453, 2015.

[32] S. Bratu and C. Barnhart, "Flight operations recovery: New approaches considering passenger recovery," *J. Sched.*, vol. 9, no. 3, pp. 279–298, 2006.

[33] N. Jafari and S. Hessameddin Zegordi, "Simultaneous recovery model for aircraft and passengers," *J. Franklin Inst.*, vol. 348, no. 7, pp. 1638–1655, 2011.

[34] S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin, "A large neighbourhood search heuristic for the aircraft and passenger recovery problem," *Oper. Res.*, vol. 9, pp. 139–157, 2011.

[35] N. Jozefowiez, C. Mancel, and F. Mora-Camino, "A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions," *J. Oper. Res. Soc.*, vol. 64, no. 2001, pp. 1–20, 2011.

[36] K. Sinclair, J.-F. Cordeau, and G. Laporte, "A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem," *Comput. Oper. Res.*, vol. 65, pp. 42–52, Jan. 2016.

[37] L. Marla, B. Vaaben, and C. Barnhart, "Integrated Disruption Management and Flight Planning to Trade off Delays and Fuel Burn," pp. 1–34, 2011.

[38] U. Arıkan, S. Gürel, and M. S. Aktürk, "Integrated aircraft and passenger recovery with cruise time controllability," *Ann. Oper. Res.*, 2013.

[39] M. Le, J. Gao, and C. Zhan, "Solving the Airline Recovery Problem Based on Vehicle Routing Problem with Time Window Modeling and Genetic Algorithm," pp. 822–828, 2013.

[40] S. J. Maher, "A novel passenger recovery approach for the integrated airline recovery problem," *Comput. Oper. Res.*, vol. 57, no. 0, pp. 123–137, 2015.

[41] S. Salhi, "Defining tabu list size and aspiration criterion within tabu search methods," *Comput.*

Oper. Res., vol. 29, no. 1, pp. 67–86, Jan. 2002.

[42] S. Tsubakitani and J. R. Evans, "Optimizing tabu list size for the traveling salesman problem," *Comput. Oper. Res.*, vol. 25, no. 2, pp. 91–97, Feb. 1998.

Appendix A

Scientific research paper

A decision support system for airline disruption management at KLM Cityhopper

K Aslan BSc[a], J.A.C. van der Veer[b], Dr.ir. H.P.M. Veeke[a], Prof. dr. ir. G. Lodewijks[a]

Delft, The Netherlands

[a] *Department of Marine and Transport Technology, Faculty of Mechanical, Maritime and Materials Engineering (3ME), Delft University of Technology, Delft, The Netherlands*
[b] *KLM Cityhopper, Schiphol-Oost, The Netherlands*

Abstract - The operations of KLM Cityhopper (KLC) is a complex undertaking. Many safety, statutory and contractual rules and agreements need to be considered for managing its operations. Hence, KLC spends considerable time, effort and money on planning its resources carefully. Unfortunately, it is all too common that unforeseen events during the day of operation can make the carefully constructed timetable, fleet and crew schedules infeasible – hereby having an adverse effect on passenger itineraries. Thereto, KLC has an entity named Operations Control where human experts control the safety of operations and deal with complications that have a negative effect on the flight schedule.

Currently, there is little to no insight in the decision-making process. Operators use different information systems to assess the necessary steps to mitigate a problem. However, it is difficult to evaluate if all options have been analyzed, and if all the required information is consulted. Furthermore, the required information may not even be readily available or accessible.

To identify any potential shortcomings, the KLC airline disruption management process is analyzed using the Delft Systems Approach. It is found that no information is available on the possible solutions to a problem and the consequences thereof.

Thereto, a decision support system that can help to solve the complex problem of reallocating aircraft is presented. A metaheuristic based on tabu search is implemented to explore a plethora of solutions that is then presented to the operator. The operator can choose from this pool of solutions and select the most desirable one given the situation at hand. Interviews with operators and computational tests show that the system is capable of presenting quality solutions in relatively short computational time.

Introduction

KLC is a wholly owned subsidiary of KLM Royal Dutch Airlines. It operates as KLM's regional carrier from Amsterdam and functions as a capacity provider to KLM. It transports passenger to or from Amsterdam, often for the first or final leg of their itinerary. The relationship between both companies is such that KLM sells the tickets and KLC operates the flight on behalf of KLM.

To do so, KLC manages its own fleet and crew, which consists out of 3 aircraft types, totalling at 47 aircraft, and approximately 1000 FTE crew members. Depending on the season KLC operates at about 55 destinations, hereby operating and transporting about 250 flights and 20 thousand passengers a day.

In order to successfully manage its expensive resources, KLC starts its planning process long before the day of operation (DoO). The planning process commences with a timetable provided by KLM. Given the timetable, the net and gross crew capacity required to operate all flights on the timetable are calculated. Then, based on this information, crew pairings are formed. These are anonymous crew duties that are constructed to (near) optimality while respecting all legal rules and minimising costs. This is an iterative process that begins about two years before and ends about two months before the day of operation. Based on the provided crew pairings, personalised crew rosters are constructed and published every week for four weeks in the future. The rostering process is followed by the roster maintenance period during which modifications are made to the rosters to compensate for crew availability. Then, a short period before the day of operation all flights get assigned to the available aircraft hereby constructing the fleet schedule.

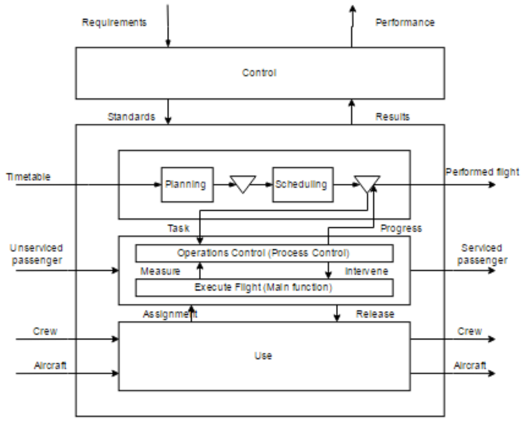
During the day of operation operations are monitored by the Operations Control (OC) division of the airline. OC ensures the original timetable is performed as much as possible with minimal deviations. Nonetheless, on the day of operation, unforeseen events can cause for the carefully constructed fleet and crew schedules to become infeasible. For example, bad weather can cause airports to operate with fewer runways and force the airline to operate on an adapted flight schedule, or a sudden aircraft malfunction may make an aircraft unavailable to perform its scheduled flights for that day. Operations Control works to coordinate operations and prevent or reduce the impact of such disruptions.

Methods

Delft Systems Approach

For the analysis of the existing airline disruption management (ADM) process the Delft Systems Approach was used. The Delft Systems Approach (DSA) provides a framework that can be used to systematically analyze processes that contain information and/or material flows. Using this approach we are able to identify the shortcomings of the existing process and form a research goal.

KLM Cityhopper's processes are modelled using the process performance model. The customer order that is to be handled is a timetable that is to be performed. Resources are needed to perform the service and passengers are what needs to be serviced. Thus, resources, timetable, and passengers are seen as input. The main function itself is to transport passengers on behalf of KLM. Each order is considered delivered when the scheduled flight leg is performed and the passengers are transported. Thus, as output, we have used resources, serviced passengers, and a serviced timetable. When we apply the process of planning, scheduling and coordinating resources as described earlier, we obtain the model in Fig. 1.



F 1 The process performance model

Here it is found that the process of airline disruption management (ADM) functions as the process control of the system. A process control function can be described as a feedback or feed forward control loop that consists out of the following four steps: measure, compare, control and intervene. Measuring is the stream of information that is used. Comparing is evaluating this information against the set standards, i.e. the current schedule. Controlling is determining the actions following this comparison. Finally, intervening is implementing these actions.

In our analysis of the KLC ADM process we find that there is not adequate information available during the control step. Controllers have little or no information on the following points:

- The financial impact, i.e. the resulting cost of decisions

- The passenger impact, how many passengers are affected by the intervention?
- The available slack, what possible moves were there available?
- The impact on crew connections. Are there any crew regulations violated and are there crew members available?
- The impact on on-time performance, i.e. do these action delay other flights elsewhere and is this necessary?

To address these items the following research goal is formulated:

"Design a decision support system that provides insight on the quality of possible solutions to a disruption by informing on the consequences thereof."

Modelling

Following this research goal, we first define a set of design requirements for a decision support system (DSS). The DSS should be able to provide a diversity of solutions using the most common actions that are used to solve problems, i.e. swapping aircraft, swapping between aircraft types, delaying flights and cancelling flights, while adhering to the maintenance schedule (meaning scheduled maintenance cannot be altered).

From literature, a metaheuristic based on tabu search is found to be capable of delivering the desired output for the DSS [2]. This algorithm is adapted to fit KLC's purposes and further improved upon by adding some minor but useful improvements. This method is then implemented in a software package using Java programming and tested with domain experts on real data using realistic scenarios.

Tabu Search

The tabu search, originally proposed by Glover [3] is an enhanced local search algorithm. The tabu list allows for temporarily accepting worse solutions than the current solution. This enables the search to escape from local optima and to find better solutions. The tabu list contains a set of rules that describes if a move to a certain solution is allowed. This prevents being stuck in a local optimum and also prevents cycling, repeated visits to the same solutions. The size of a tabu list and the tabu aspiration criterion are to be determined for the problem at hand.

For the aircraft recovery problem, making one or more aircraft tabu is likely the most obvious choice of a tabu attribute. If an aircraft is made tabu, it cannot be selected again for the next L iterations. However, this also means that a large part of the neighbourhood cannot be visited for the next L iterations. This is an example of making a change in the solution tabu. Whether a solution has been visited recently can be evaluated by checking the assignment for an aircraft. However, doing this multiple times per iteration is computationally expensive. Instead, it is more efficient to make the value of a solution tabu. Meaning, the number of delays, delay minutes, cancellations, swaps within aircraft type, swaps between aircraft types and the cost associated with it that are used as the tabu characteristics. This makes it less time consuming to compare solutions and check for tabu status. A caveat is that in rare cases different solutions may have the same values, but this also means that the solutions have very similar characteristics. In this case making this solution tabu can still benefit the algorithm as the solutions share very similar characteristics, and accepting such a solution does not diversify the search.

The tabu search algorithm we used is presented below. In Step 1 of this algorithm, the current status of the fleet schedule is accepted as the initial solution. This is the situation in where a disruption occurs that results in (propagated) delays or cancellations in the flight schedule. The aircraft routings in this initial solution are stored in the set F_{sk}. The set N in Step 3 will not contain any flights in the beginning (it makes little sense to reassign already cancelled flights), but will contain possible cancelled or unassigned flights over the next few iterations of the algorithm.

Algorithm: Tabu Search

1. *Start with a feasible solution, s, with an objective value of z*
2. *Let F_{sk} be the set of flights assigned to aircraft k in solution s*
3. *Let N be the set of flights not assigned to any aircraft*
4. ***For** a certain number of iterations {*
5. *Let z^{temp} = large positive value, $N^{emp} = N$, $s^{temp} = s$*
6. *Let P = the set of unique aircraft pairs in s*
7. *Shuffle P*
8. ***While** $P \neq \emptyset$ {*
9. *Select an aircraft pair in P: aircraft a and b*
10. *P = P – (a,b)*
11. $N^{temp} = N^{temp} \cup F_{sa} \cup F_{sb}$
12. *Find all feasible routes for flights in N^{temp} using the combination of aircraft a and b*
13. *Select the combination of routes for aircraft a and b that has the cheapest cost*
14. *value associated with it*
15. *Let s_{new} be the solution with the two new routes, z_{new} the value of this solution*
16. *and N_{new} the set of cancelled flights in this solution*
17. $\boldsymbol{If}$ $solution\ not\ tabu$ $\boldsymbol{and}$ $z_{new} < z^{temp}${
18. *Add solution to tabu list*
19. $s^{temp} = s_{new}, z^{temp} = z_{new}, N^{temp} = N_{new}$
20. *}*
21. *}*
22. *}*

The main search starts in Step 4 where an iterative search is started that runs for a fixed number of iterations. In each iteration, the neighbourhood of the current solution, s, is explored and the best solution stored; this takes place between Step 8 and Step 18. Before starting the while loop the aircraft pairs in P are shuffled using a Fisher-Yates shuffle. Then, in the while loop, a pair of aircraft is selected from all unique aircraft pairs in P. All the flights that are yet to be flown added to a pool with the cancelled flights (Steps 8-10). Then, in Step 11, a nested recursive tree build algorithm is used to find all possible routes for the aircraft in the selected pair, assuming the aircraft can only fly the flights in the set N^{temp}. In Step 12, a combination of routes is selected that has the least associated cost with it.

In the tree build algorithm, in steps 12 and 13, connecting flights have to be feasible from connection perspective. This means that a flight, Flight 1, can connect to another flight, Flight 2, if the destination of Flight 1 is the departure airport of Flight 2. In addition, there should be enough time between the arrival of Flight 1 and departure of Flight 2 to be able to carry out the arrival and departure services (minimum turnaround time). If Flight 2 needs to be delayed to be able to meet this requirement, a minimum delay equal to the turnaround time should be used. Also, the last flight at the end of the recovery period should connect to the original schedule. That way the original schedule can be followed again after the recovery period.

Although not restricted, unnecessary swaps and swaps between aircraft types are discouraged. This is simply because each swap has to be communicated to the different parties involved during operations (e.g. crew, ground handlers) and this creates an extra workload, something that has to be avoided if possible.

Cost function

The cost function used to evaluate solutions consists of two components: estimated costs and a weight factor. The estimated costs are based on information provided by the available information systems and it has a 'hard' and 'soft' component. The 'hard' component are the actual costs that stem from delays and cancellations, i.e. reimbursements, re-bookings, hotel and meal costs, passenger compensation for excessive delays as mandated by European law. The 'soft' component consists of the costs that are associated with loss of future value, e.g. likelihood of returning as a customer, damaging word of mouth marketing. Both in the 'hard' and 'soft' components the costs are evaluated by passenger value, which is based on passenger class (frequent flyer status).

The cost of a solution is then evaluated by summing over all flights n and get the cost of each flight F:

$$\sum_{i=1}^{n} F_i^{cost}\text{, where:}$$

$$F_i^{cost} = C \cdot Cost_{cancellation} \quad (1)$$

$$F_i^{cost} = D \cdot Cost_{delay} + Cost_{swap} + S \cdot Cost_{subswap} \quad (2)$$

Equation (1) is valid if the flight is cancelled and equation (2) is valid if the flight is not cancelled. the $Cost_{cancellation}$ and $Cost_{delay}$ are the cancellation and delay cost respectively, $Cost_{swap}$ is a weight selected to prohibit unnecessary swaps, and $Cost_{subswap}$ is either a penalty for leaving passengers behind when downsizing an aircraft, or a standard penalty if no passengers are affected, S is then also a weight selected to prohibit subtype swaps.

Results

The purpose of providing computational results is not only to present the efficiency of the model but also to show how the decision support system can be used. The tested cases are selected by the manager Operations Control.

The discussed cases involve realistic disruptions that could happen in the daily operation. We demonstrate how the DSS could be used to find solutions to a problem. A solution is a revised aircraft schedule, where flights may be delayed, cancelled or swapped between aircraft.

In our tests, the maximum delay allowed is 90 minutes, for longer delays it will be more cost efficient to cancel a flight. The recovery period is from a certain moment on the day until the end of that day, which is around 21:00 UTC. We make use of actual flight schedules of KLC.

Cases are presented as an initial situation in which the disruption occurs, and as solutions that are found for the initial situation. These are depicted in a table using the characteristics: affected passengers, aircraft swaps, subtype swaps, delay minutes, delays, cancellations and cost. Affected passengers denote passengers affected by either cancellations or downsizing of aircraft. Aircraft swaps are swaps within the same fleet type. Subtype changes are swaps across aircraft types. Delay minutes is the total amount of (propagated) delay in a schedule. Delays is the number of delays. Cancellations represents the number of cancellations and cost is the cost associated with this situation/solution in thousands of euros.

The algorithm is implemented in Java and the tests are done on a laptop with an AMD A8-4500M CPU that is clocked at 1.9 GHz and equipped with 8 GB RAM.

Case	Flights	Aircraft	Subtypes	Airports	Maintenance Slots	Delayed aircraft	Aircraft on ground
1	296	44	3	52	16	1	0
2	198	30	2	45	13	0	2
3	302	45	3	52	30	0	3

T 1 The datasets

Case 1

Case 1 is run on a full day's schedule. In this case one flight is delayed for a duration of 60 minutes that leads that propagates to next flights, totalling 230 minutes of delay. When the DSS is started the delayed flight is in progress, meaning it cannot be solved and the minimum amount of delay minutes in any solution should be 60. Table 2 presents the characteristics for the initial situation and the top 5 solutions.

Iterations	50
Duration	37 seconds
# Solutions	47

Solution	Affected passengers	Aircraft swaps	Subtype swaps	Delay minutes	Delays	Cancellations	Cost (x1000)
0	0	0	0	230	4	0	32.7
26	0	4	0	60	1	0	10,5
25	0	5	0	60	1	0	10,75
24	0	6	0	60	1	0	11
29	0	2	4	60	1	0	11
46	0	2	4	65	2	0	11

T 2 Case 1

Case 2

Case 2 uses the same schedule as case 1, but without all the Fokker aircraft. Also, instead of a delay, we will deal with two aircraft on ground (AOG) situations this time around. In this case, 5 flights on 2 aircraft are 'cancelled', i.e. unassigned because they cannot be operated anymore. The four 'cancelled' flights impact 340 passengers in total. The cost value of € 3,515,000 is skewed as solutions with cancellations are multiplied with a weight of 100. So the actual cost would be € 35,150. Table

3 presents the characteristics for the initial situation and the top 5 solutions.

Iterations	50
Duration	32 seconds
# Solutions	36

Solution	Affected Passengers	Aircraft swaps	Subtype swaps	Delay minutes	Delays	Cancellations	Cost (x1000)
0	340	0	0	0	0	5	3515
19	356	9	0	0	0	4	1672,25
4	356	9	0	1	1	4	1672,49
5	356	7	0	10	1	4	1672,75
32	356	12	0	3	1	4	1673,42
31	356	14	0	0	0	4	1673,5

T 3 Case 2

Case 3

In Case 3 we use a different flight schedule as compared to cases 1 and 2. In this case we have three aircraft with technical defect that have to go unexpected maintenance. The four 'cancelled' flights impact 392 passengers in total. The cost value of € 6,500,500 is again skewed as solutions with cancellations are multiplied with a weight of 100. So the actual cost would be € 65,000.

Iterations	50
Duration	41 seconds
# Solutions	50

Solution	Affected Passengers	Aircraft swaps	Subtype swaps	Delay minutes	Delays	Cancellations	Cost (x1000)
0	392	0	0	0	0	5	6500
5	0	10	0	21	1	0	39.6
49	0	13	0	21	1	0	40.35
29	0	13	0	26	2	0	45.35
30	126	11	6	20	2	2	47.75
11	126	12	1	26	2	2	57.6

T 4 Case 3

Discussion

When we analyse the solutions for Case 1, we note that the first four solutions all solve for the flights with propagated delay. The delayed flight in Solutions, 26, 25, 24 and 29 was already in progress. The solutions have a slightly different cost value due to the penalty for aircraft swaps and differences. Solution 26, reduces the cost of the initial situation by 68%.

In Case 2, we find that all solutions contain 4 cancellations but are a factor 2 less than the cost for the initial situation. In this solution there is not enough slack to be able to reschedule all flights, thus what the DSS proposes it to cancel the flights with the least associated cost. Hereby reducing the cost of the initial situation by a maximum of 53%.

In Case 3, we note that in the solutions we find three of the solutions with the least cost all flights are assigned. The 21 minutes delay we note is an inconsistency in the provided data and is thus not correct (it should be subtracted from the total delay minutes). This case was also given to a domain expert whom found a solution with the following characteristics:

Solution	Affected Passengers	Aircraft swaps	Subtype swaps	Delay minutes	Delays	Cancellations	Cost (x1000)
User	0	11	0	0	0	0	N/A

T 5 The characteristics to a soliution by an expert used to case 3

The solutions are very similar. The explanation as to why solution 5 contains one fewer swap is because the DSS suggest breaking apart a flight rotation to London. This is something that is typically not done in practice because it brings complications to the crew duties. Such crew restrictions were not considered by our DSS as we considered this out of scope for this study.

The designed DSS is a prototype that is able to find a multitude of possible solutions to a given situation. Insight on the quality of a solution is provided in specific characteristics, namely: the number of resulting delays and the corresponding total delay minutes, the number of cancellations, the total cost associated with passenger inconvenience and the number of affected passengers. The characteristics number of aircraft swaps and subtype swaps are used to indicate how much effort it is to reach an outcome. These solutions are then presented in a table that will assist the user in finding the best solution for a given situation.

The algorithm used was adapted from literature with some minor but valuable improvements. Computational tests have shown the efficacy of the algorithm. In our experience a pool of solutions could be obtained in about 20 to 120 seconds depending on the input variables as number of iterations, length of the recovery period and the amount of aircraft. The inspected results in the test cases show that the obtained solutions are realistic and logical. Improvements can be made in filtering out the unnecessary solutions in the big pool of solutions.

We think the prototype is successful within the confinements of this research, however there are some caveats. So far, we have only addressed the aircraft recovery problem without any crew restrictions. For a successful DSS, crew restrictions also have to be considered. Now, the DSS will suggest solutions that may be valid from a fleet perspective, but not practical when also accounted for any crew consequences. Hence, crew restrictions like latest time of arrival (LTA), obligatory crew ground time during aircraft changes, pilot restrictions in subtype swaps and aircraft-crew connections need to be considered and added to the model. Second, the crew recovery problem is still persistent even when accounted for crew restrictions. Further research is recommended to investigate how the crew recovery problem can be integrated in our model.

References

[1] H. P. M. Veeke, J. A. Ottjes, and G. Lodewijks, The Delft Systems Approach, 1st ed., vol. 1. Springer London, 2008.

[2] T. Andersson, “Solving the flight perturbation problem with meta heuristics,” J. Heuristics, vol. 12, no. 1–2, pp. 37–53, 2006.

[3] F. Glover, “Future Paths for Integer Programming and Links to Artificial Intelligence.” Computers and Operations Research, vol. 13, no 5, pp. 533-549, 1986

Appendix B

Case 1

Fig. 25 The initial situation

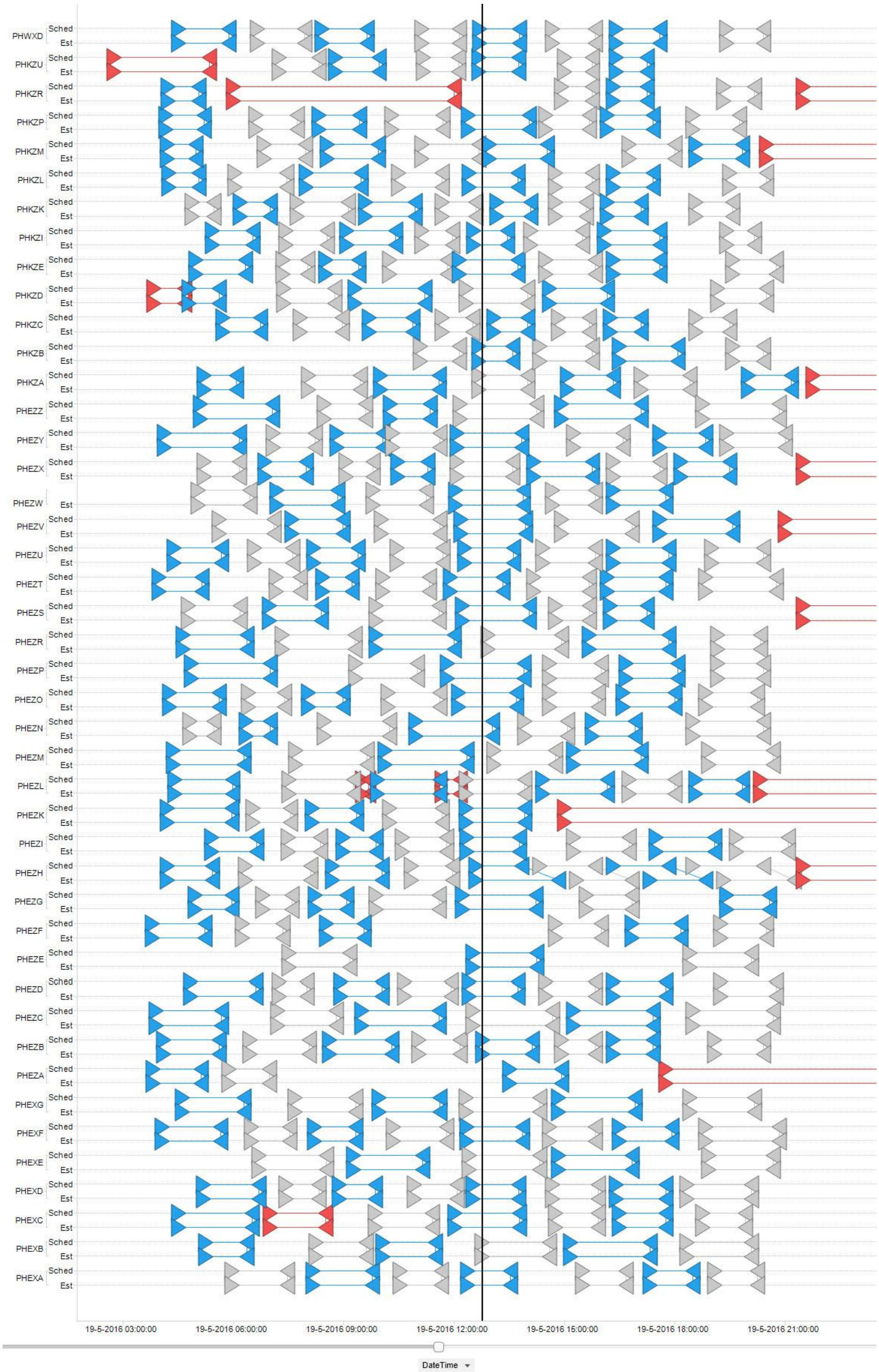
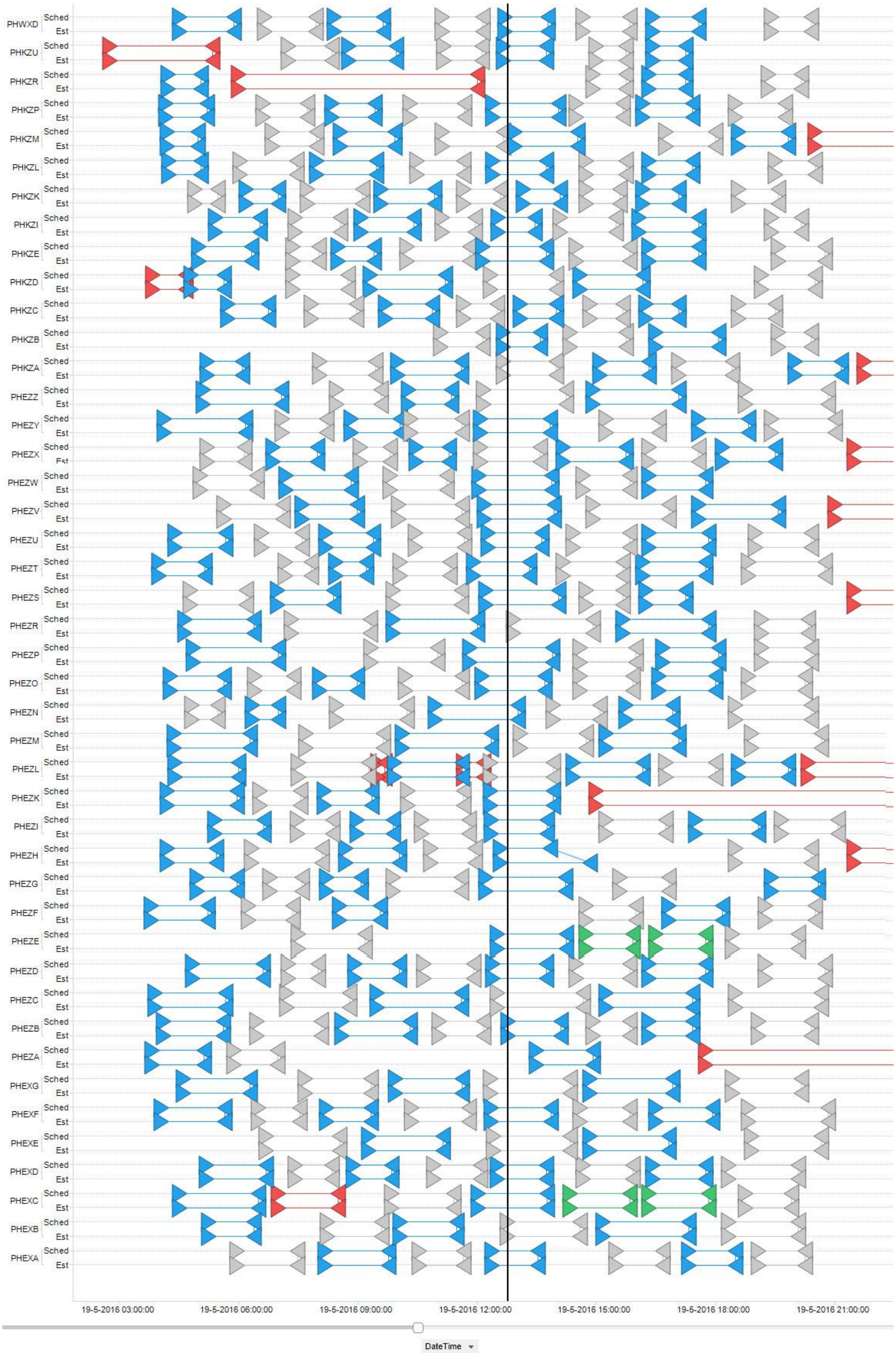


Table 12 The solutions for this case

Solution	Affected passengers	Aircraft swaps	Subtype swaps	Delay minutes	Delays	Cancellations	Cost
1	7	4	8	73	2	0	14,35
2	0	4	4	70	2	0	11,5
3	16	2	4	88	3	0	16,15
4	0	6	4	65	2	0	12
5	4	10	4	101	6	0	13,95
6	16	6	3	148	5	0	27,45
7	7	10	4	68	2	0	14,85
8	0	8	0	75	2	0	11,5
9	11	4	5	85	2	0	14,55
10	0	6	0	75	2	0	11
11	4	4	5	70	2	0	12,7
12	0	8	0	62	3	0	11,93
13	4	10	4	75	3	0	13,95
14	0	6	0	90	2	0	11
15	0	8	2	90	2	0	12
16	0	10	2	90	2	0	12,5
17	0	6	0	90	4	0	11
18	0	8	0	90	2	0	11,5
19	30	0	4	66	3	0	19
20	0	10	4	100	3	0	13
21	0	10	0	100	3	0	12
22	12	6	4	90	2	0	15,35
23	0	8	0	96	4	0	11,5
24	0	6	0	60	1	0	11
25	0	5	0	60	1	0	10,75
26	0	4	0	60	1	0	10,5
27	0	10	0	81	4	0	12
28	0	8	0	60	1	0	11,5
29	0	2	4	60	1	0	11
30	4	2	4	60	1	0	11,95
31	0	6	0	80	3	0	11
32	0	6	4	96	4	0	12
33	0	13	4	60	1	0	13,75
34	0	5	6	60	1	0	12,25
35	0	6	1	90	2	0	11,25
36	0	9	5	85	4	0	13,45
37	0	10	0	97	3	0	12
38	0	15	0	103	3	0	13,85
39	0	12	0	97	3	0	12,5
40	30	2	4	90	2	0	19,5
41	0	10	0	75	2	0	12
42	30	4	4	75	2	0	20
43	0	10	4	65	2	0	13
44	0	7	5	80	2	0	12,5
45	30	2	4	129	7	0	26,7
46	0	2	4	65	2	0	11
47	4	5	4	73	2	0	13,3

Fig. 26 Solution 26 from Table 8



Case 2

Fig. 27 Initial situation in case 2, aircraft PHEZV and PHEZP are not able to fly due to a technical failure.

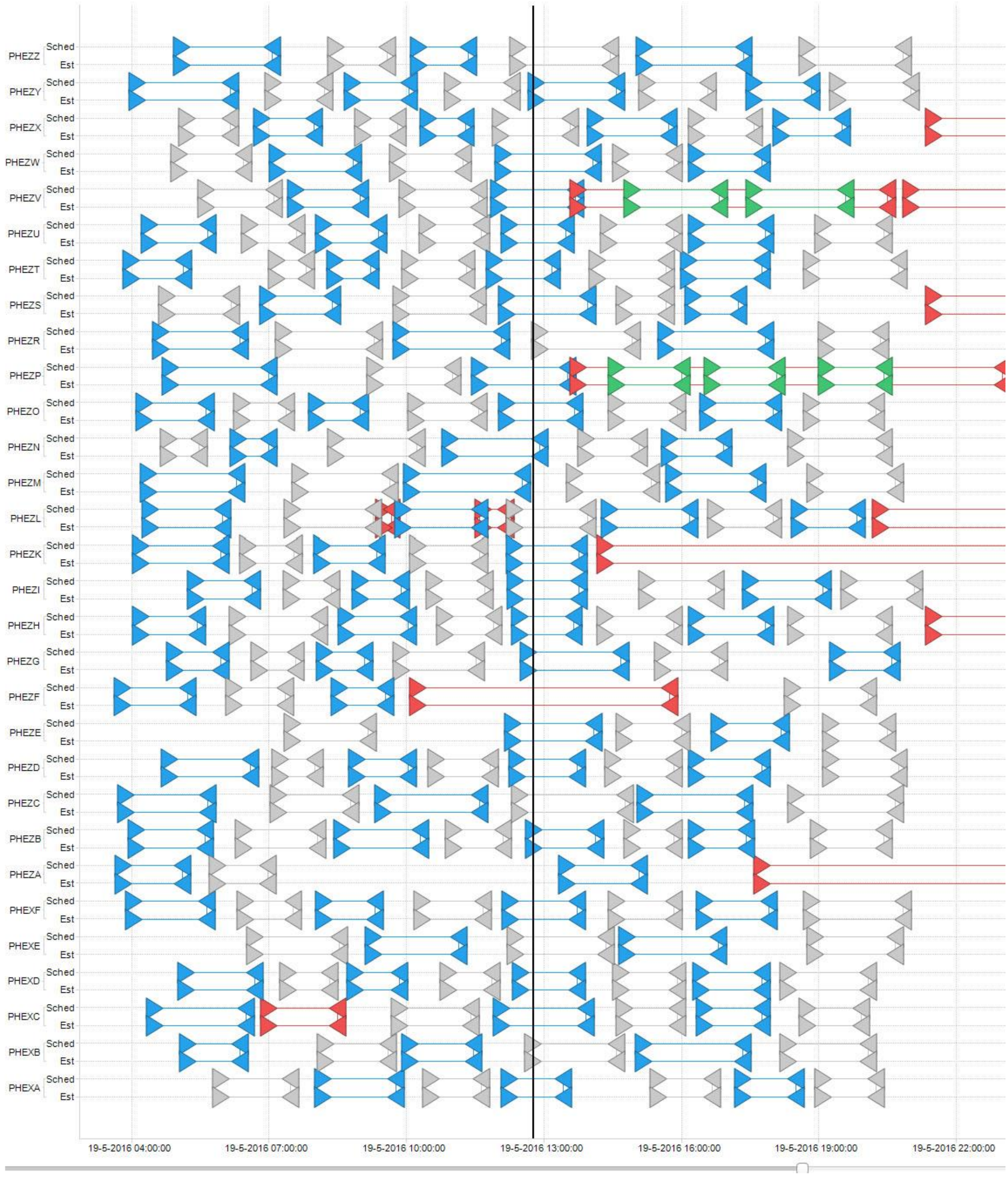
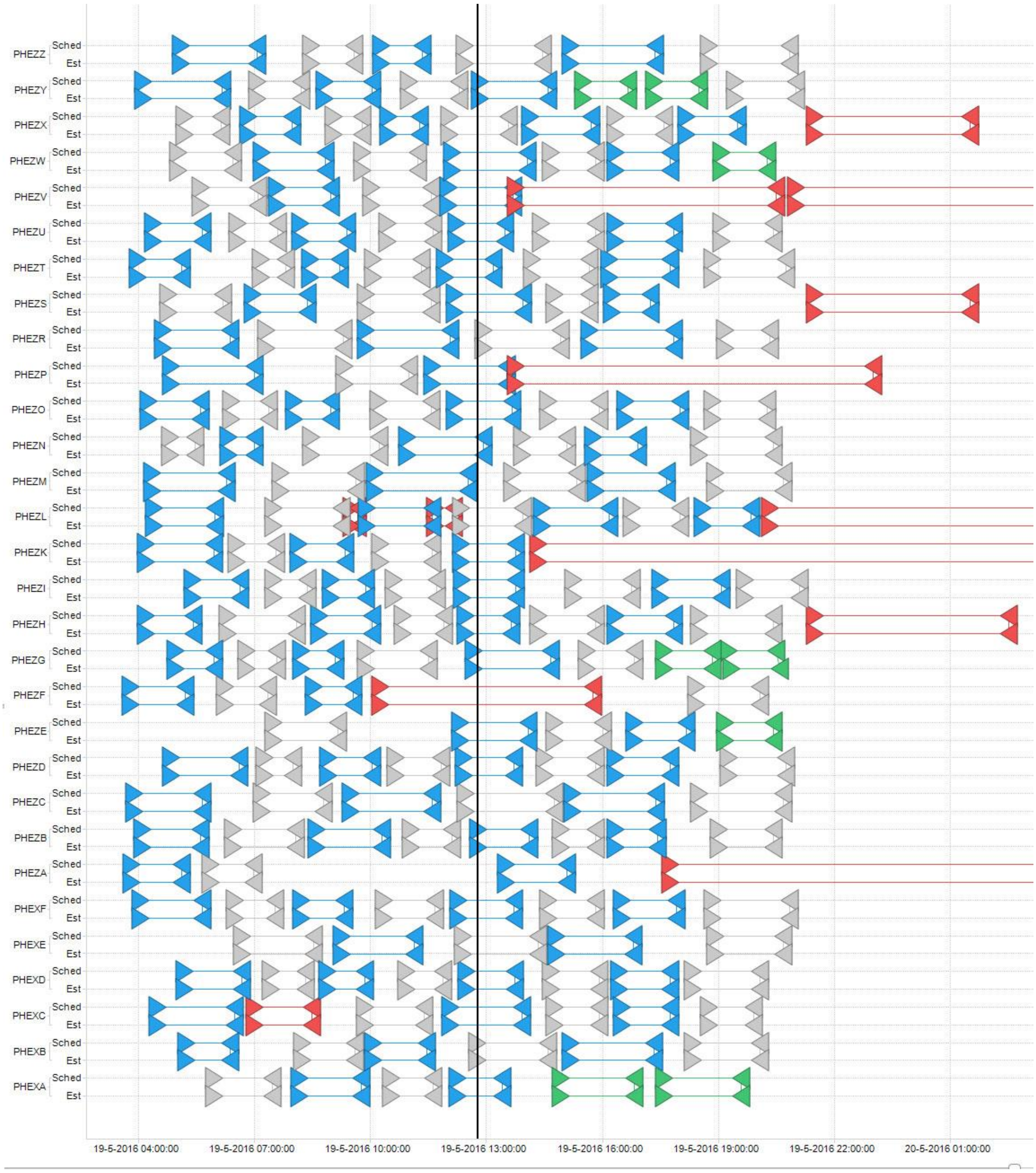


Table 13 Solutions to case 2, unsorted

Solution	*Affected Passengers*	*Aircraft Swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
1	348	9	0	0	0	4	1677,25
2	348	7	0	3	1	4	1677,17
3	348	6	0	10	1	4	1677,5
4	356	9	0	1	1	4	1672,49
5	356	7	0	10	1	4	1672,75
6	348	6	0	13	2	4	1677,92
7	348	8	0	4	2	4	1678,38
8	356	16	0	35	2	4	1678,3
9	348	8	0	13	2	4	1679,14
10	348	8	0	3	1	4	1678,14
11	356	15	0	13	2	4	1675,89
12	348	7	0	0	0	4	1676,75
13	356	12	0	3	1	4	1674,14
14	356	11	0	11	2	4	1673,99
15	348	9	0	10	1	4	1678,25
16	348	8	0	3	1	4	1677,42
17	348	5	0	10	1	4	1677,25
18	348	8	0	4	2	4	1677,66
19	356	9	0	0	0	4	1672,25
20	348	8	0	0	0	4	1677
21	348	6	0	13	2	4	1678,64
22	348	5	0	15	1	4	1677,75
23	348	11	0	1	1	4	1677,99
24	348	9	0	5	1	4	1677,75
25	356	7	0	13	2	4	1673,89
26	348	8	0	1	1	4	1677,24
27	348	7	0	6	2	4	1678,31
28	348	12	0	0	0	4	1678
29	348	7	0	4	2	4	1677,41
30	348	9	0	3	1	4	1678,39
31	356	14	0	0	0	4	1673,5
32	356	12	0	3	1	4	1673,42
33	348	7	0	1	1	4	1676,99
34	348	10	0	10	1	4	1678,5
35	348	5	0	13	2	4	1678,39
36	348	10	0	0	0	4	1677,5

Fig. 28 Soluton 12 to case 2



Case 3

Fig. 29 Initial situation in case 3 Aircraft PHKZD, PHEZK and PHEZH are AOG

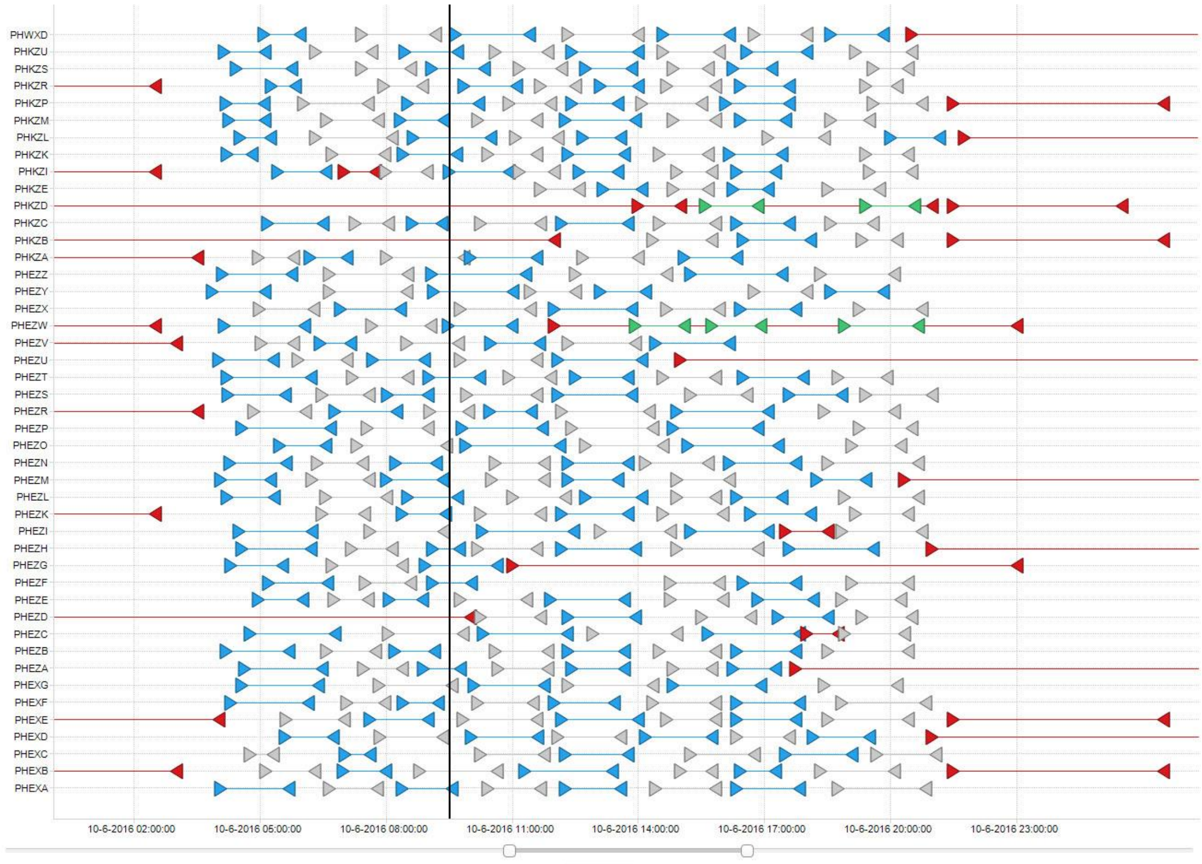


Table 14 Solutions to case 3 unsorted

Solution	*Affected Passengers*	*Aircraft Swaps*	*Subtype Swaps*	*Delay Minutes*	*Delays*	*Cancellations*	*Cost*
1	0	5	9	38	4	0	77.85
2	159	11	8	5	1	2	131.75
3	25	14	6	26	2	0	130.6
4	126	22	11	43	5	2	102.1
5	0	10	0	21	1	0	39.6
6	151	16	4	31	3	2	141.1
7	183	10	5	25	2	2	206
8	4	15	12	33	3	0	92.35
9	161	6	7	20	1	2	144
10	151	9	4	31	3	2	139.35
11	126	12	1	26	2	2	57.6
12	130	16	10	38	4	2	102.6
13	127	12	9	33	3	2	85.1
14	161	14	7	20	1	2	146
15	161	2	7	20	1	2	143
16	152	22	8	26	2	2	148.1
17	25	12	13	31	3	0	152.6
18	321	13	10	59	5	4	449.45
19	127	18	12	39	5	2	101.5
20	0	18	8	33	3	0	73.6
21	151	10	10	46	6	2	172
22	5	14	7	21	1	0	68.1
23	39	9	8	39	6	0	187.95
24	126	13	12	43	5	2	102.35
25	152	8	6	34	4	2	150
26	0	9	9	37	5	0	79.95
27	25	9	4	26	2	0	124.35
28	140	19	5	34	5	2	117.95
29	0	13	0	26	2	0	45.35
30	126	11	6	20	2	2	47.75
31	153	9	7	31	3	2	152.85
32	152	11	6	31	3	2	145.35
33	127	17	5	26	2	2	69.35
34	176	7	11	31	3	2	240.35
35	126	14	10	31	3	2	69.2
36	135	17	5	21	1	2	88.35
37	126	18	7	39	5	2	87.1
38	0	9	9	34	4	0	74.55
39	25	13	4	31	3	0	130.35
40	14	8	4	22	2	0	91
41	127	7	11	43	5	2	98.85
42	0	21	11	38	4	0	86.85
43	211	2	9	33	3	2	301.8
44	161	3	7	21	1	2	160.35
45	151	19	4	26	2	2	136.85
46	4	11	11	39	5	0	96.25
47	139	11	11	22	2	2	112.05
48	170	16	3	24	2	2	183.5
49	0	13	0	21	1	0	40.35
50	127	12	5	26	2	2	68.1

Fig. 30 Solution 5 to case 3