The Air Cargo Allocation Plan Recovery Problem

Recovering from disruptions on air cargo allocation planning for combination airlines

O.Y.F. Teng

Delft University of Technology

The Air Cargo Allocation Plan Recovery Problem

Recovering from disruptions on air cargo allocation planning for combination airlines

by

O.Y.F. Teng

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,

Version: 1.0
Publication date : July 29, 2020

Student number:	4735935	
Project duration:	September 2019 - July 2020	
Supervisors:	Dr. B.F. Lopes dos Santos,	TU Delft, supervisor
	Dr A. Bombelli,	TU Delft, daily supervisor
	H. Zwitzer,	KLM, supervisor
	Dr. ir. R. Vos,	External examiner

TU Delft

Preface

This report is created as the final work for my completion of the Master of Science in Aerospace Engineering. I was given the opportunity to research the field of air cargo and specifically focus on disruptions on air cargo allocation planning. Interestingly, a lot of research has been conducted for passenger airline optimization, but only a little for air cargo. This in particular spiked my interest to learn as much as possible on the topic. The risk following this new area of research was the unknown factors and uniqueness of the air cargo business. I would have to admit, that at multiple moments in time I thought the whole research was impossible due to the problems arising. Luckily, I was always able to find adequate guidance to achieve the necessary results to complete the research. The problem turned out to be both an interesting academical research subject and an applicable relevant problem from a practical point of view. In the end, I am very happy with the steep learning curve I traversed and with my personal development.

During the thesis I was given the opportunity to complete my work at Air France KLM Martin Air Cargo. Although the last few months were completed at home I am still grateful for all the support I was given. I specifically want to thank Hans Zwitzer for always being available for questions. As an expert in the air cargo business he always was able to give a realistic view on the topic and made sure the problem was solved as realistically as possible.

From the university I want to thank my supervisors Dr Bruno Santos and Dr Alessandro Bombelli. Although Bruno was only updated on the most recent works during the milestones of the project, he was always able to immediately give relevant feedback on the theoretical aspects of the problem. These views always gave me the correct direction to work towards. Secondly, the courses Airline Planning Optimization and Network Scheduling motivated me to pursue this field of research and without these relevant courses I would have not been able to complete this thesis. As my daily supervisor, Alessandro was always available, which I am very grateful for. Whether it was a meeting on Monday morning at 9:00 or a quick email. The theoretical feedback Alessandro was able to supply was a very big factor in how my research has formed to what it has now become.

Lastly, I want to thank my parents who always supported me through my student career. It was a long ride and I am grateful for their support mentally and financially.

O.Y.F. Teng
Delft, July 2020

Contents

1
Paper

The Air Cargo Allocation Plan Recovery Problem: recovering from disruptions on air cargo allocation planning for combination airlines

Olivier Teng

Delft University of Technology, Delft, The Netherlands

Abstract

In this paper, we introduce the Air Cargo Allocation Plan Recovery Problem, where we embrace the perspective of a combination airline that relies on belly space to transport cargo. We present a recovery model that can reallocate bookings that are offloaded because of disruptions in the demand (overbooking) or in the supply (aircraft swap of cancellation) side. The model is based on a set-partitioning mixed integer linear program formulation with an itinerary generation pre-processing step whose goal is to limit the number of possible itineraries per booking and hence the computational effort. The problem is solved for three different cases from a European airline. The three cases are, a flight cancellation, an aircraft swap, and an aircraft swap with ULD configuration. The results show that the model is adaptable to fit the time available, the operational cost of the initial solution can be improved and the larger amount of the objective function value comes from revenue loss.

1 Introduction

The air cargo transportation business has undergone drastic changes in recent times. As example, although air cargo demand has been constantly increasing in the last few years, the rate of increase has been decreasing instead. The rate of increase was 9.2% in 2017, 3.4% in 2018, and 2.0% in 2019 (IATA, 2020). From the supply side the capacity is outgrowing the demand with an increase of freighter capacity of 5.4% in 2018 (IATA, 2020). When the capacity of freighters outgrows the demand for cargo, it is inevitable that combination airlines, which transport passengers and cargo in their belly space, will phase out their freighter aircraft.

With the competitiveness of the air cargo business, fast and reliable deliveries are essential in order to avoid loss of clients. Creating a robust network for combination airlines is more difficult due to the uncertainty in the expected cargo to deliver and the fact that cargo transport is generally of second-tier importance with respect to passenger transport. For example, for most European airlines it is still possible to book cargo until shortly before take-off, making it hard to assess beforehand how much cargo will actually be booked for a flight. From another perspective, a penalty for a no-show does not exist (Wada et al., 2017). This created the common (mis)practice to overbook cargo on flights, which can create the possibility of assigning too much cargo to the aircraft. These uncertainties do not only arise from the demand side of combination flights, but also from the supply side. For instance, it is not known beforehand how much space is available for cargo in the aircraft. The reason is to be found, again, in the fact that for combination airlines passengers have priority. Hence, there might and usually will be fluctuations in the room required for passengers' luggage and the necessary fuel for the flight.

When a disruption occurs on a certain flight, the excess cargo must be reallocated to a different flight. The time available to find a solution is an important factor. In fact, the closer the disruption occurs with respect to the scheduled take-off time, the faster a solution should be found. This paper deals with disruptions that occur due to the uncertainties from the demand and the supply side for combination airlines. A recovery model is proposed to recover from these disruptions, by reallocating shipments that were booked on the disrupted flight. For many airlines, this is a process that is still executed manually by an experienced flight analyst. While manual execution can still lead to quasi-optimal solutions in some cases, we argue that mathematical optimization can complement, without necessarily replacing, flight analysts and help them design even better recovery actions. Although a lot of research work has been carried out for passenger airline recovery, this is the first recovery problem that addresses air cargo combination airlines to the best of our knowledge. As such, this is the first methodological contribution of our paper. In fact, given the totally different business rules and values of time, recovery models for passengers and cargo are intrinsically different. The second contribution regards the case studies we present. We were able to gather real flight disruption data from a partner airline, and will show how our model performs with respect to real implemented recovery operations.

The rest of the paper is organized as follows. In Section 2 the relevant literature is presented. Section 3 formulates the complete problem statement for the model. Section 4 discusses the complete formulation of the model. In Section 5, three cases of a European airline are used to test the model. Lastly, in Section 6 a conclusion is made and some recommendations are added for future research.

2 Literature Review

In order to classify the contribution of the problem, the relevant literature is discussed. We will analyze recovery problems dealing with air transport. Due to the lack of recovery models for air cargo, most of the inspiration is taken from recovery problems for passenger airlines. Then, we will briefly describe, to the best of our knowledge, the only recovery model that specifically addresses the air cargo business, and highlight similarities and differences with respect to our model. Finally, we will present an optimization model that tackles the full air cargo supply chain. Notwithstanding the fact that it is not a recovery model, it shares some similarities with our model in the way a set of feasible itineraries is pre-generated.

2.1 Passenger airline recovery models

In the case of a disruption for passenger airlines, multiple recovery actions are undertaken. The most influential are the aircraft recovery problem, passenger recovery problem and the air crew recovery problem. Promising methods are discussed for each problem.

2.1.1 Aircraft recovery

In the case of a disruption in the schedule of an airline, the aircraft recovery problem (ARP) has a large pay-off between minimizing the cost and complexity of the problem. In the process of solving the aircraft recovery problem, the goal is to decide what flights to delay or cancel and what aircraft to use for those flights.

Due to the simplicity of the ARP, compared to passenger and aircrew recovery, it is a more attractive problem to solve. However, it is still a popular research problem that has become more complex trying to reach more realistic solutions. One of the more recent works is by Khaled et al. (2018) where the authors created a multi-objective integer linear programming problem for tail assignment. When given a list of flights, the tail assignment is to assign an aircraft to each flight. The objectives are to minimize the operating cost and the deviation of the original solution. From a general repair model point of view, the authors propose four important characteristics that a recovery model must include. The most important characteristics for an air cargo situation are real time optimization and minimum deviation of the initial solution. The solution must be obtained before other modifications happen and the decision maker must then execute all these changes.

2.1.2 Air crew recovery

The complexity for recovery models increases when solving the air crew recovery problem (CRP). This is caused by the restrictions and regulations that arise when reallocating crew. For example, the maximum duration of a duty. The goal of the problem is to re-assign the necessary crew to the set of flights in order to minimize the additional costs.

Often the CRP is solved simultaneously with the ARP, however some work exists that specifically solves the CRP. Novianingsih et al. (2015) proposed a three stage heuristic, where all the options for crew swaps are analyzed and executed if possible. Then, if the swaps are not possible, an iterative random search is used to create new crew schedules. The model is tested for 214 flights with 48 crew pairings, which takes 90 seconds to solve. What is more important for a recovery model, is that the authors conclude that the problem can be solved in polynomial time.

2.1.3 Multiple integrated problems

Solving recovery problems sequentially can result in sub-optimal solutions. Therefore, approaches are taken to solve multiple problems in an integrated model.

Santos et al. (2017) solve the passenger and aircraft recovery problem with an integer linear programming model. The objective is to minimize the aircraft fuel costs as well as the additional passenger costs. The model is able to solve a schedule of a day within a few minutes.

A combination of the aircraft and crew recovery is covered by Maher (2016), where a column and row generation framework is used to improve the branch and price method. The recovery strategy the model uses is flight delays and cancellations. The model is applicable for both point-to-point and for hub & spoke networks.

Bisaillon et al. (2011) tackle the aircraft and passenger recovery problem. By applying a large neighbourhood search heuristic, the authors were able to get well performing solutions with limited computation effort. The heuristic has three phases: construction, repair and improvement. In the first two phases a feasible solution is created using the original flight schedule, then improvement moves will optimize the solution in the last phase.

During the construction phase, a randomness is added to explore multiple areas of the solution plain. These phases are iterated until a certain maximum in computational time is reached. This model was taken and improved by Sinclair et al. (2014) and after by Sinclair et al. (2016). In Sinclair et al. (2016) the authors added a mixed integer program after the heuristic was applied.

2.2 Air cargo recovery model

At this point, the only recovery model for air cargo applications is created by Delgado et al. (2020). The authors focus on solving the Air Cargo Schedule Recovery Problem (ACSRP). This problem is specified for cargo airlines that only use freighter aircraft. This gives them the freedom to completely reschedule the itineraries of the aircraft. Due to the uncertainty in the cargo, disruptions can occur. By rescheduling the flights and air crew the fluctuations in demand are accounted for. The problem is solved using three different crew management policies, which create different cost evaluations. The objective is to minimize the costs of rescheduling flights and crew, in order to still transport the bookings.

2.3 Air cargo optimization

A limited amount of research has been executed in the field of optimization problems for air cargo, especially compared to passenger airline optimization problems. However, Archetti and Peirano (2019) elaborate a model that addresses the full supply chain of freight forwarding transportation. They called the problem the Air Transportation Freight Forwarder Service Problem (ATFFSP). Their scope is from the origin of the shipper to the destination of the consignee. This includes picking up the shipment at the customer and bringing it to the airport. Here the shipment is flown to the airport in the destination country, where it is delivered at the final destination. The authors create a time-space network and solve the problem using a mixed integer linear programming model. The objective of the problem is to minimize the total cost to perform all shipments over a given time horizon.

Angelelli et al. (2020) continue with the ATFFSP and rely on a math-heuristic to solve the problem. The authors propose a column generation approach to generate a set of routes and the identification of the best solution with a set-partitioning mathematical model.

Another work covering air cargo optimization is from Tang et al. (2008). The model focuses on solving the multi-commodity flow problem for combination airlines and therefore, also includes cargo. Again, this model does not include any recovery decisions. The objective is to minimize the cost by using a hybrid mixed integer program and heuristic model. The heuristic is necessary due to the NP-hardness of the problem.

3 Problem Statement

The uncertainties that arise for air cargo transportation come from both the demand and supply side. From a supply perspective, the capacity for cargo on combination airlines is determined after calculating the capacity required for fuel and passenger luggage (Morrell, 2011). Apart from that, combination airlines can choose to swap aircraft or even cancel flights based on their model for the aircraft, passenger and crew recovery problem. Contrary to the air cargo schedule recovery problem, the schedules of the flights for combination airlines cannot be adjusted in order to accommodate the bookings. The unpredictability from the demand side is due to two factors. The first one is the time frame at which bookings are placed. From a passenger perspective, a seat can be booked long before take-off. Air cargo however, is used for fast delivery, which means bookings are often booked only short before take-off. Air cargo can typically be booked 10 days beforehand and the last 50% of the bookings is placed 4 days before take-off (Sandhu and Klabjan, 2006). The second unpredictable factor comes from the turn up rate of bookings and if it does show up it remains uncertain what the actual size of the bookings are (Amaruchkul et al., 2007). These issues arise due to the competition within the air cargo business. Clients are able to book the same booking at multiple airlines and without consequence or penalty they can cancel the booking at their own convenience (Wada et al., 2017). In the case of a disrupted flight, excess bookings must be reallocated to new itineraries. Furthermore, deciding which bookings are reallocated and which remain on the same itinerary must be defined. At most airlines this is decided and executed by an experienced flight analyst and can therefore not guarantee efficient solutions. The problem proposed is referred to as the Air Cargo Allocation Plan Recovery Problem (ACAPRP) and is approached by minimizing the additional costs generated by the reallocation strategy. Where the term "recovery" is borrowed from passenger airline side of the problem.

As a consequence of the dynamic nature of air cargo transportation, different recovery situations occur. These situations are determined by the time available before take-off and at what stage the allocation of the bookings is. Throughout the recovery process different phases are taken into account. Once the flight is opened for booking shipments, these bookings have not arrived at the warehouse. When approaching the take-off time of the flight, the booked shipments start arriving or are completely cancelled. Shortly before take-off, but depending on the flight, all the bookings have arrived at the warehouse and all the Unit Load Devices (ULD) are configured to be loaded onto the aircraft. As a recovery model is focused on creating a fast and high quality solution, the stage at which the allocation of the bookings is, should not make a difference. The recovery problem therefore considers if the bookings on a flight are only digitally booked or are physically at the warehouse fully packed in their intended ULD. If a disruption occurs in the first part of the process when the bookings have not arrived at the warehouse, this implies the booking can be reallocated to a different itinerary without any operational difficulty. The reallocation of a booking during the last phase of the process could have additional operational issues. Depending on the time the bookings arrive at the warehouse, the ULDs could already be consolidated at the time the disruption occurs. As a consequence, the handling steps required and operational cost to reallocate a booking increase.

An additional difference between cargo and passengers are the rules required for connections between flights. For passenger flights, the time between origin and destination is a constraining factor, which is not necessarily the case for cargo. Furthermore, unlike passengers, cargo requires operational handling in order to arrive at the connecting flights. As a consequence, a larger layover time between flights is necessary for cargo. Based on the flights in the network the reallocation of a booking to a different flight will often incur an allocation to a completely different itinerary.

The main contributions of the work are defined as follows:

- An integrated framework is created to solve the air cargo allocation plan recovery problem, by reallocating bookings while minimizing additional costs
- The variation in phase of completion of the allocation process are taken into account. It does not matter whether the bookings have not arrived or if all the ULDs are already packed
- Both demand and supply uncertainty are considered as a disruption source, which from a cargo perspective is a problem that has not yet been addressed
- Consider the viable itineraries available for each booking

4 The Air Cargo Allocation Plan Recovery Problem

4.1 Definitions and notation

The airline transports a set of bookings $k \in \mathcal{K}$, each of which has an individual origin and destination. The available travel options a booking can follow is defined by the total set of itineraries $i \in \mathcal{I}$. Each itinerary is defined by an origin and destination, a combination of arcs and the departure and arrival time. The alternate itineraries for a booking are determined based on the origin, destination and departure time of the itinerary $i \in \mathcal{I}^k$. All the itineraries are created using the available arcs $f \in \mathcal{F}$. All arcs are defined by a flight or a truck between two airports, with a specific departure and arrival time. In the problem it is not necessary to make a distinction between an actual flight and a truck because trucks have the same properties as a flight but with a longer duration of a trip and larger capacities. The arcs are divided into two sets. The first set comprises the disrupted flight and the equivalent flights on that flight leg $f \in \mathcal{F}^E$, where an equivalent flight is defined as a flight with the same origin-destination combination as the disrupted flight. The second set comprises all the connecting arcs to and from the disrupted flight $f \in \mathcal{F}^C$. Each of the airports within the network have stations $m \in \mathcal{M}$ where all the cargo is handled before it is loaded onto a flight. These stations are served by the arcs within the network. The problem has an initial solution, in which each booking has an initial itinerary. Therefore, all the bookings are assigned a specific set of arcs they initially traverse. From the other perspective, when analysing an arc, it is known which bookings are assigned to that arc $k \in \mathcal{K}^{f^-}$. As a consequence, these bookings could potentially be reallocated to different arcs when selecting an alternative itinerary. Therefore, each arc also has a set of bookings that could potentially be allocated to that arc $k \in \mathcal{K}^{f^+}$. Depending on the phase of the allocation process of the disrupted flight, some bookings might already be packed in a ULD. In the situation of a disruption during this phase the packed bookings are divided over the set of configured ULDs $h \in \mathcal{H}$. This does not necessarily mean that all the bookings are packed and every ULD is configured. Each of the completed ULDs h has an initial solution with a set of bookings that has been packed into the ULD $k \in \mathcal{K}^{h^-}$. The new solution can potentially be reallocate bookings to a different ULD on the same flight, which means there exists a set of bookings that could be reallocated to ULD h, $k \in \mathcal{K}^{h^+}$. When addressing the problem from a booking perspective, each booking has an initial ULD and a set of ULDs it can be reallocated to $h \in \mathcal{H}^{k^+}$. All the sets are described in Table 1.

Table 1: Sets for the ACAPRP

Set	Description
$\mathcal{K}$	Set of all the bookings
$\mathcal{I}$	Set of all the itineraries
$\mathcal{F}$	Set of all the arcs
$\mathcal{M}$	Set of all the stations
d	The disrupted flight
$\mathcal{H}$	The set of configured ULDs in the disrupted flight

The objective of the problem is to minimize the additional operational costs and revenue loss, to make sure the capacity of the flights is not exceeded. This is achieved by finding new itineraries for bookings. To achieve this objective a set of parameters is necessary. SC_m denotes the station handling cost per kg that is incurred if a booking is handled at station m. FC_f is the fuel cost per kg incurred when a booking traverses arc f.

One of the key features of air cargo is fast deliveries. When a disruption occurs, bookings are reallocated to new itineraries and could arrive later than the client expected. Depending on the bookings priority this can have consequences. For example, fresh produce is not fresh if it arrive a week later. To take into account this factor for priority items, a revenue loss is incurred when the booking arrives later than its initially scheduled time. RC_i^k is the revenue loss when booking k is assigned to itinerary i. V^k and W^k are the volume and weight of booking k. $VCap_f$ and $WCap_f$ are the total volume and weight capacity of arc f. As mentioned, a distinction is made between the disrupted flight, including the equivalent flights, and the connecting flights. For the connecting flights the parameters BV_f and BW_f are the booked volume and weight of arc f. Similar to the arcs, $VCap_h$ is the total capacity of ULD h. Furthermore, when a ULD is configured and bookings are reallocated to a different flight, additional operational handling costs must be incurred due to opening the ULD. These costs are referred to as DC_h, which is the cost of opening ULD h. Due to the disruption, the quantity of ULD positions available on the flight will differ. ULD_{pos_n} is the amount of ULD positions available for ULD type n. For the case of a disruption all the positions available will change to zero. All the parameters are summarized in Table 2.

Table 2: Parameters for the ACAPRP

Parameter	Description
SC_m	Airport handling costs at station m
FC_f	Fuel cost for arc f
RC_i^k	Revenue loss for booking k when selecting itinerary i
V^k	Volume of booking k
W^k	Weight of booking k
$VCap_f$	Volume capacity of arc f
$WCap_f$	Weight capacity of arc f
BW_f	Booked weight on arc f
BV_f	Booked volume on arc f
$VCap_h$	The volume capacity of ULD h
DC_h	Disassembling cost for ULD h
ULD_{pos_n}	Number of ULD positions for ULD type n available on the disrupted flight

For the Mixed Integer Linear Program (MILP), binary variables are used to formulate the problem. The variable $U := \{U_i^k : k \in K \wedge i \in I^k\}$ denotes whether a booking k is assigned to a new itinerary i. If the itinerary of a booking remains unchanged the variable $Q := \{Q^k : k \in K\}$ is used. Both these variables are associated to complete itineraries and not arcs, the variable $u := \{u_f^k : k \in K \wedge f \in F^k\}$ denotes the link between a booking k and an arc f, which is necessary for the compliance to the capacity constraint for each arc. For the ULD reconfiguration multiple decision variables are added. Based on the parameter ULD_{pos_n}, a new selection of ULDs is made. The decision variable determining if ULD h actually remains on the disrupted flight is $Y := \{Y_h : h \in H\}$. As an additional cost is incurred when opening a ULD, $Z := \{Z_h : h \in H\}$ is a binary variable that determines per ULD h if it is opened. The last variable linked to the ULDs is whether a complete assembled ULD is reallocated to a different equivalent flight, this is determined by the decision variable $X := \{X_{fh} : h \in H \in f \in F^{h^+}\}$. Additional decision variables are necessary to determine if and where the bookings are configure in the ULDs. Similar to the Q variable, if the booking is initially located inside a ULD, the variable $s := \{s^k : k \in K^{d^-}\}$ determines whether a booking remains in its initially assigned ULD. Oppositely, $r := \{r_h^k : k \in K \wedge h \in H^{k^+}\}$ determines whether a booking k remains on the same flight, but is reassigned to a different ULD h. The decision variables can be found in Table 3.

Table 3: Decision variables

Decision Variable	Type	Description
U_i^k	binary	1 if booking k is reallocated to itinerary i, 0 otherwise
Q^k	binary	1 if booking k remains on its initial itinerary, 0 otherwise
u_f^k	binary	1 if booking k is assigned to arc f, 0 otherwise
Y_h	binary	1 if ULD h is loaded onto the disrupted flight, 0 otherwise
Z_h	binary	1 if ULD h is opened, 0 otherwise
X_{fh}	binary	1 if ULD h is allocated to arc f, 0 otherwise
s^k	binary	1 if booking k remains in its initial ULD, 0 otherwise
r_h^k	binary	1 if booking k is reallocated to ULD h, 0 otherwise

4.2 Math-heuristic column generation itineraries

In the occurrence of a disruption where all the bookings traverse one flight leg and have the same destination, it is easy to determine what route they should use. However, for a hub & spoke air cargo airline almost all the bookings must take multiple arcs towards multiple destinations. As a first step in the model, using the set of bookings and arcs, all the possible itineraries are generated with a math-heuristic. This set of itineraries can be reused and appended every time a disruption occurs. Additionally, depending on the airline and the disruption the heuristic can easily be adapted for each situation.

For the generation of itineraries, multiple factors must be taken into account. When a booking itinerary includes a layover, the booking must be transported from the arrival gate to the departure gate. Therefore, enough layover time must be included between two arcs for the airport handling team to move the booking. Another factor comes from the origins and destinations of the bookings. An itinerary is only generated when at least one booking from the set of bookings has the same origin and destination of the itinerary. Multiple arc itineraries can only have a fixed amount of feasible combinations of truck and flight arcs. These can be categorized depending on whether the disruption occurs on an inbound or an outbound flight from the hub. If the disruption occurs on an outbound flight, the set of itineraries is created using the connecting arcs to the hub and departing arcs from the disruption destination. If the disruption occurs on an inbound flight, the set of itineraries is created using the connecting arcs towards the origin of the disrupted flight and the departing arcs from the hub. The different combinations can be seen in Figure 1. Once all the itineraries are generated, each booking is matched with its feasible itineraries.

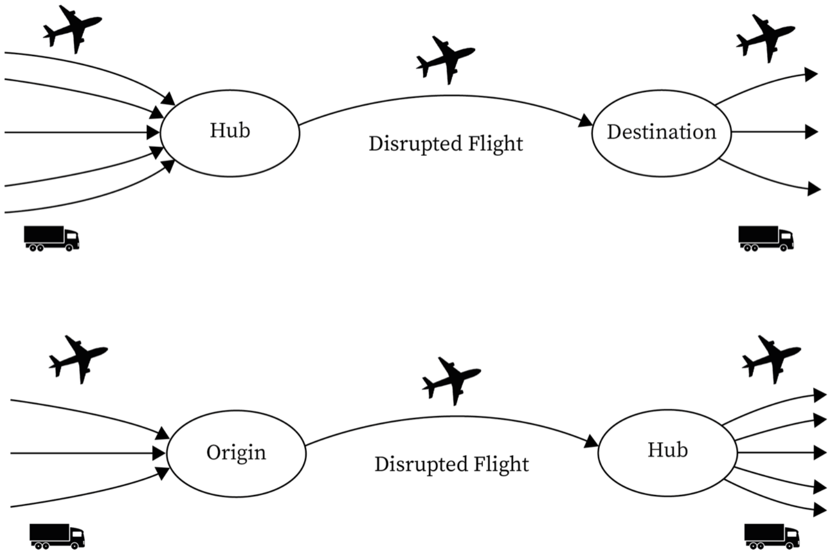


Figure 1: Feasible itinerary options. Upper is outbound and lower is inbound.

4.3 Mathematical Formulation

The ACAPRP can be formulated as a MILP that minimizes the function $\mathcal{J}$, which is defined as follows:

$$
\begin{aligned}
\min \quad \mathcal{J} = & \sum_{k\in K}\sum_{i\in I^k}\left(\sum_{m\in M^i} SC_m W^k + \sum_{f\in F^i} FC_f W^k + PC_i^k\right) U_i^k && \text{(1a)}\\
& + \sum_{k\in K}\left(\sum_{m\in M^{Q^k}} SC_m W^k + \sum_{f\in F^{Q^k}} FC_f W^k\right) Q^k \\
& - \sum_{k\in K}\left(\sum_{m\in M^{Q^k}} SC_m W^k + \sum_{f\in F^{Q^k}} FC_f W^k\right) + \sum_{h\in H} DC_h Z_h
\end{aligned}
$$

s.t.

$$Q^k + \sum_{i\in I^k} U_i^k = 1 \qquad \forall k \in K, \tag{1b}$$

$$\sum_{f\in F^{Q^k}} u_f^k \geq |F^{Q^k}| Q^k \qquad \forall k \in K, \tag{1c}$$

$$\sum_{f\in F^i} u_f^k \geq |F^i| U_i^k \qquad \forall i \in I^k, k \in K, \tag{1d}$$

$$\sum_{f\in F^{E^k}} u_f^k = 1 \qquad \forall k \in K, \tag{1e}$$

$$\sum_{k\in K^f} V^k u_f^k \leq VCap_f \qquad \forall f \in F^E, \tag{1f}$$

$$\sum_{k\in K^{f+}} V^k u_f^k + \sum_{k\in K^{f-}} V^k (u_f^k - 1) \leq VCap_f - BV_f \qquad \forall f \in F^C, \tag{1g}$$

$$\sum_{k\in K^f} W^k u_f^k \leq WCap_f \qquad \forall f \in F^E, \tag{1h}$$

$$\sum_{k\in K^{f+}} W^k u_f^k + \sum_{k\in K^{f-}} W^k (u_f^k - 1) \leq WCap_f - BW_f \qquad \forall f \in F^C, \tag{1i}$$

$$
\begin{aligned}
&u_d^k = s^k + \sum_{h \in H^{k^+}} r_h^k && \forall k \in K^{d^-}, && \text{(1j)}\\
&u_d^k = \sum_{h \in H^{k^+}} r_h^k && \forall k \in K^{d^+}, && \text{(1k)}\\
&\sum_{k \in K^{h^-}} s^k \geq |k^{h^-}|(1 - Z_h - \sum_{f \in F^E, f \neq d} X_{fh}) && \forall h \in H, && \text{(1l)}\\
&\sum_{h \in H^n} Y_h \leq ULD_{pos_n} && \forall n \in N, && \text{(1m)}\\
&\sum_{k \in K^{h^-}} V^k s^k + \sum_{k \in K^{h^+}} V^k r_h^k \leq VCap_h Y_h && \forall h \in H, && \text{(1n)}\\
&\sum_{k \in K^{h^-}} u_f^k \geq |K^{h^-}| X_{fh} && \forall h \in H, f \in F^E, f \neq d, && \text{(1o)}\\
&\sum_{k \in K^{h^+}} r_h^k \leq |K^{h^+}| Z_h && \forall h \in H, && \text{(1p)}\\
&\sum_{k \in K^{h^+}} r_h^k \leq |K^{h^+}|(1 - \sum_{f \in F^E, f \neq d} X_{fh}) && \forall h \in H, && \text{(1q)}\\
&U_i^k \in \{0,1\}, && k \in K, i \in I^k, && \text{(1r)}\\
&Q^k \in \{0,1\}, && k \in K, && \text{(1s)}\\
&u_f^k \in \{0,1\}, && k \in K, f \in F^k, && \text{(1t)}\\
&Y_h \in \{0,1\}, && h \in H, && \text{(1u)}\\
&Z_h \in \{0,1\}, && h \in H, && \text{(1v)}\\
&X_{fh} \in \{0,1\}, && h \in H, f \in F^{h^+}, && \text{(1w)}\\
&s^k \in \{0,1\}, && k \in K^{d^-}, && \text{(1x)}\\
&r_h^k \in \{0,1\}, && k \in K^{d^+}, h \in H^{k^+} && \text{(1y)}
\end{aligned}
$$

The objective function 1a minimizes the difference in costs compared to the initial solution. If a booking differs from its itinerary, the handling costs and fuel cost of the initial itinerary will not be incurred, instead the operational costs for the new itinerary is selected. Furthermore, a revenue loss is incurred if a high priority booking arrives later than its initial itinerary. Depending on the configuration of ULDs additional costs can be incurred if a ULD is opened.

The constraints 1b-1i consist out of the constraints required for the reallocation of bookings to new itineraries. Constraint 1b ensures that one and only one itinerary is selected for each booking, whether it is its initial itinerary Q^k or a new itinerary U_i^k. Given that itineraries could consist of multiple arcs, if an itinerary is selected, constraint 1c and 1d make sure the booking is assigned to each arc of that itinerary. As a consequence of the formulation of constraint 1c and 1d, the model could potentially allocate bookings to multiple equivalent flights. To avoid this constraint 1e is added, where each booking can be allocated to exactly one equivalent flight.

The problem size is limited by only selecting the bookings from the disrupted flight and its equivalent flights. There are two reasons for taking this approach. The first reason is to avoid a large chain of bookings that must be reallocated. Air cargo is a highly dynamic transportation method, if the flight analyst is forced to reallocate a large set of bookings, by the time all the adjustments are made the solution might not be relevant anymore. The second motive is associated with the runtime of a recovery problem. The problem occurs shortly before take-off, forcing the flight analyst to react quickly with an alternative solution. When analysing the bookings from the disrupted flight, each booking has an individual origin and destination. Therefore, bookings have alternative connecting arcs. If the full network were to be selected it would be necessary to take into account all the bookings in the connecting arcs, creating a snowballing effect because all of those bookings also have different connecting flights, origins and destinations. Only selecting the bookings on the disrupted flight and equivalent flights has a direct influence on the determination of the capacity constraints of the arcs. Constraint 1f and 1g take into account the volume capacity of the disrupted flight and the connecting arcs. For the disrupted flight and the equivalent flights the sum of all the bookings is smaller than the actual capacity of the flight (constraint 1f). Constraint 1g does not consider the volume of the arc but, the volume that is available at the time of the disruption. This includes the bookings that are initially planned on that arc. In the scenario that a booking is reallocated onto the disrupted flight or an equivalent flight, the booking will not contribute to the capacity of its initial flight in constraint 1f. If the similar situation occurs for a connecting arc (constraint 1g) the booking will decrease the amount of booked volume and therefore increase the free capacity of the arc. The same approach is taken for the weight constraints 1h and 1i. Figure 2 illustrates the difference between the capacity constraints of equivalent flights and the connecting flights.

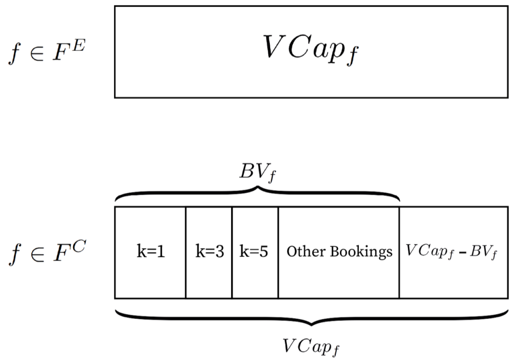
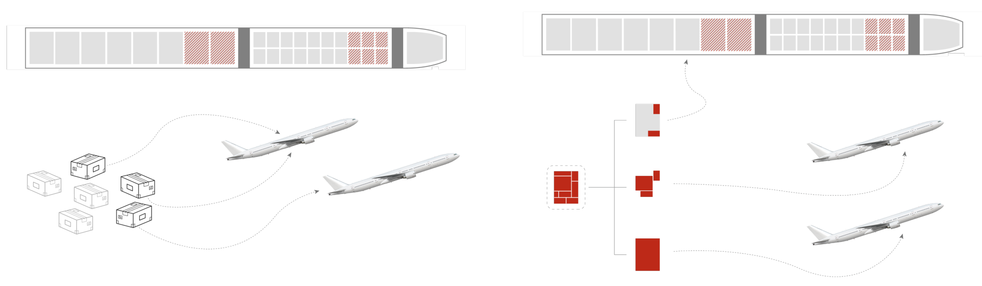


Figure 2: Capacity of flight compared to capacity available

Depending on the flight, if a disruption occurs long before the departure of the flight, the ULDs have not yet been configured and bookings have not been loaded. Gradually, as the departure date and time comes closer, more ULDs are loaded with bookings. When a disruption occurs and a booking must be removed from the flight, but it was already loaded onto a ULD, additional operational handling actions must be accounted for. To solve this, a cost is incurred when a ULD is opened in order to remove a loaded booking. An additional reallocation option is to completely reallocate the configured booking onto a different flight, in that case no disassembly cost is necessary. Constraints 1j-1q refer to the disassembly or reallocation of a ULD. When a booking is allocated to the disrupted flight and already loaded onto a ULD, constraint 1j ensures that it will remain in its initial ULD, or be allocated to a different ULD, or completely reallocated to a different flight. Constraint 1k is similar to constraint 1j, but it considers bookings that are not loaded onto a ULD. These bookings could be allocated to the disrupted flight, but are still in the warehouse. Therefore, the possibility exists where these bookings are also added to one of the existing ULDs, forcing the ULD to be opened. If a booking is is reallocated to a different flight or ULD constraint 1l forces the opening of that ULD, except if all the bookings loaded onto that ULD are reallocated to the same flight. With the reallocation of all the bookings in the ULD, the complete ULD is reallocated to the other flight without any disassembly actions. Constraint 1m is the capacity constraint to confirm the amount of ULD positions available for ULD type n. To confirm that the capacity of a ULD is never exceeded, constraint 1n is created for each ULD h. As mentioned, when all the bookings of a certain ULD are reallocated to a new flight, the complete ULD is reallocated. The link between the bookings in ULD h and the complete reallocation to a new flight is made in constraint 1o. Constraints 1p and 1q are required for bookings that could potentially be reallocated to ULDs. If no bookings are removed from a ULD but a booking is reallocated to that ULD, it must still be opened or is not allowed to be reallocated completely to a new flight.

All the different reallocation options are displayed in Figure 3.



(a) Reallocation of bookings without ULDs **(b)** Reallocation of bookings with ULDs

Figure 3: Booking reallocation options

5 Case studies

5.1 Description of the airline

For a large European hub & spoke combination airline, three cases are addressed that focus on different problems in terms of disruption type, time occurrence of the disruption, and location of the disruption. These different cases were selected on purpose to analyze how the model performed under considerably different situations. The two hubs are in Amsterdam Schiphol airport (AMS) and Paris Charles de Gaulle airport (CDG), where the airline has narrow-body aircraft flying through Europe and wide-body aircraft flying intercontinental. The transportation within Europe the turnaround times of aircraft are too short to load cargo, the airline therefore chooses to transport cargo within Europe using trucks, the cargo is then transported through the hubs to intercontinental destinations. The airline has a network of 76 destinations outside Europe, with a flight frequency of 402 flights a week. The network reaches North and South America, Africa and Asia. The truck network consists of 58 destinations. For the generation of new itineraries the airline has a specific set of rules. The first one is the differentiation between small shipments and large shipments. Due to their size, it is easier to transfer small shipments, therefore, the airline assigns a layover time of at least 2.5 hours. Large shipments are assigned a layover time of at least 5 hours. When analyzing the possible arc connections for an itinerary, the airline has certain options for outbound and inbound flights. The available itinerary options for outbound and inbound disruptions can be seen in Table 4 and Figure 4. Another specification of the airline is the use of multiple priority levels for bookings. Each booking is assigned a priority value ranging from 1 to 9, where 9 and 8 are low priority and 7 to 1 are high priority. The distinction between different priority categories is incorporated in the model using the revenue loss. For the low priority bookings no revenue loss is ever incurred. This decision can be made because no consequences are involved if the booking arrives late at the client. The next priority category has a revenue loss of exactly the revenue of the booking, but following that category all revenue loss has an added multiplier ranging from 2 to 4. The multiplier is added to the model to make a distinction in the priority of the different categories.

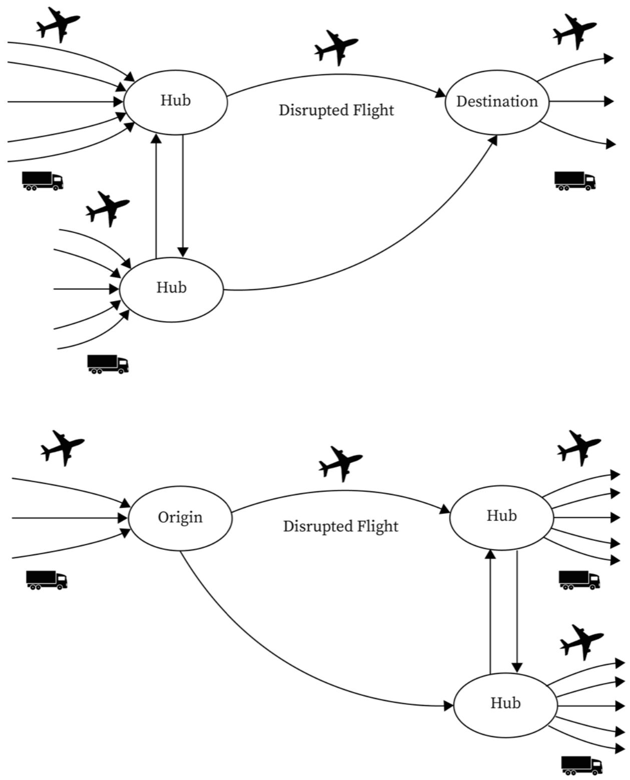


Figure 4: Feasible itinerary options for the airline. Upper: outbound, Lower: inbound

Table 4: Inbound/outbound itineraries

Itinerary options	Arc Quantity
Outbound	
hub - disruption destination	1
connecting arc - hub - disruption destination	2
hub - disruption destination - connecting arc	2
hub - hub - disruption destination	2
connecting arc - hub - disruption destination - connecting arc	3
Inbound	
Disruption origin - hub	1
Connecting arc - disruption origin - hub	2
Disruption origin - hub - connecting arc	2
Disruption origin - hub - hub - connecting arc	3
Connecting arc - disruption origin - hub - connecting arc	3

5.2 Case 1: Flight cancellation for inbound flight

The first case is built around a flight cancellation. The flight was planned from Hong Kong airport (HKG) to AMS at 5:35 on 10-11-2019. The disruption occurred at 00:31 on 10-11-2019, shortly before departure. When a disruption occurs a few hours before departure, the probability is high that all the bookings are loaded onto ULDs. However, due to data unavailability, it was unknown if and where the bookings were loaded. Therefore, for this case the assumption is made that none of the ULDs are configured and none of the bookings are loaded onto ULDs.

This case is an inbound flight, which means a lot of the bookings will have their own destination but come from the same origin. Divided along the 31 bookings on the original flight are already 10 different destinations, but only one origin. The summation of the volumes of these bookings is 79.5 m^3. With a volume capacity of 90.8 m^3 the flight had a high load factor. However, in comparison to an aircraft swap, the model does not have to assess what bookings remain on the flight because all the bookings must be reallocated. For the selection of the equivalent flights, 9 additional flights are added, again both to AMS and CDG. This totals to an amount of 144 bookings all coming from HKG and destined to a total of 31 destinations. These additional flights create a time horizon from 5:35 10-11-2019 to 15:30 14-11-2019. By adding all the connecting arcs, a total of 1,478 flights are used to create itineraries.

Due to the large time frame selected the possibilities for itineraries will also increase. As time is still a key factor, a sensitivity analysis is applied for this case. By varying two parameters the trade-off between solution quality and time can be assessed. The first parameter that is adjusted is the layover time. When looking at a time frame of four days, one of the generated itineraries is a flight on the first day to the hub but a connecting arc four days later. Preferably this is to be avoided for two reasons. The first reason is the lack of inventory cost in the model, by keeping the time at the warehouse short this can be turned into an even value for the itineraries. From a business perspective, air cargo is all about fast deliveries. Keeping the bookings in the warehouse for a long time will defeat the purpose of air cargo. By varying the maximum layover time, the amount of possible itineraries will decrease and the time to actually generate them too. The second parameter to vary is the time horizon of the model, specifically varying the selection of equivalent flights. From a theoretical point of view, the least amount of equivalent flights possible is linked to the volume that must be reallocated to other flights. If enough volume capacity exists divided among the equivalent flights, then a solution is possible. However, when selecting more equivalent flights, we increase the solution space and, potentially, the chances to obtain a better solution.

5.3 Case 2: Aircraft swap for outbound flight

The second case is from an aircraft swap for a flight from CDG to São Paulo airport (GRU), implying it is an outbound flight. The disruption occurred at 16:57 on 08-11-2019 and the departure time of the flight was at 23:20 on the same day. With a time until departure of 6 hours, a lot of the bookings of the disrupted flight were already in the warehouse in CDG. This also means most of the ULDs were already configured and loaded with bookings. However, the assumption is still made that the bookings had not been loaded onto a ULD. Due to this assumption the set H contained no ULDs and only constraints 1b to 1i are necessary. In addition, the cost of opening a ULD had not been activated.

The available volume capacity of the flight was almost halved from 112 m^3 to 63 m^3. With the flight being fully booked with 31 bookings, the volume capacity is occupied with 98 m^3. After the disruption is occurred at least 35 m^3 worth of bookings must be reallocated. A lot of the bookings must be reallocated, but what is important about this case is that the model must assess which bookings remain on the same flight and which are reallocated. The bookings of the disrupted flight all originated from CDG, except for one booking that came from Lyon (LYS). The destination airports for the bookings on the disrupted flight were to GRU, except for one booking that was destined for Belo Horizonte International Airport (CNF).

As mentioned in the model description, not only bookings from the disrupted flight are selected to be reallocated, but also bookings from equivalent flights. When selecting the equivalent flights to the disrupted flight, both flights from CDG to GRU and AMS to GRU are selected because the airline flies from both hubs to the destination. For this case, four additional flights are added, two from CDG and two from AMS. The decision to select only four additional flights is based on three factors. The first is that at least four were necessary to achieve enough free capacity to reallocate the $35m^3$. The second factor was linked to the arrival time of the bookings. When selecting more flights the time horizon increases and the bookings could potentially be allocated to an itinerary that arrives multiple days later. The time horizon of the disrupted flights and the equivalent is from 23:20 08-11-2019 to 9:55 10-11-2019. All bookings can therefore arrive around two days later. The last factor is linked to the execution time of the model. When adding more flights, additional bookings and arcs are added. This will increase the time necessary to create the itineraries. These additional flights added another 154 bookings to allocate that are also coming from different airports. Apart from CDG and AMS, the bookings came from 28 different locations mostly inside Europe and are destined for 5 airports apart from GRU. Using these bookings a set of 471 flights are used to create the initial and new itineraries.

5.4 Case 3: Aircraft swap for outbound flight with ULD configuration

To assess the effects of the configuration of ULDs on a disrupted flights, the aircraft swap case was adapted by adding the full ULD configuration. Initially the flight from CDG to GRU had a capacity of 112 m^3, which breaks down to 11 pallets with a volume of 10 m^3 and 2 bulk containers with a volume of 1 m^3. This configuration together with the bookings on the disrupted flight are used to create an initial ULD configuration. The initial configuration is displayed in Table 5. Due to the size of some of the bookings, these not fit inside one ULD. These bookings are split into part shipments by dividing them into two shipments. The model is not capable of creating part shipments and therefore, the bookings are addressed as two separate bookings. Furthermore, they cannot be split up into even smaller shipments. As an addition to the model for this case, all the broken down part shipments of a booking are forced to take exactly the same itinerary.

After the disruption, the capacity reduced to 63 m^3, which translates to 6 pallets and 3 bulk containers. The handling cost of opening ULDs is determined in communication with the airline. In order to open a pallet the net and plastic shield must be removed, this leads to an opening cost of €50,-. The bulk containers only need to be opened, which has an operational cost of €20,-. However, in addition to the initial case the model will have to assess which ULD is disassembled and which ULD remains unopened.

Apart from the ULD reconfiguration, everything remains the same in comparison to the initial aircraft swap case.

Table 5: ULD load configuration

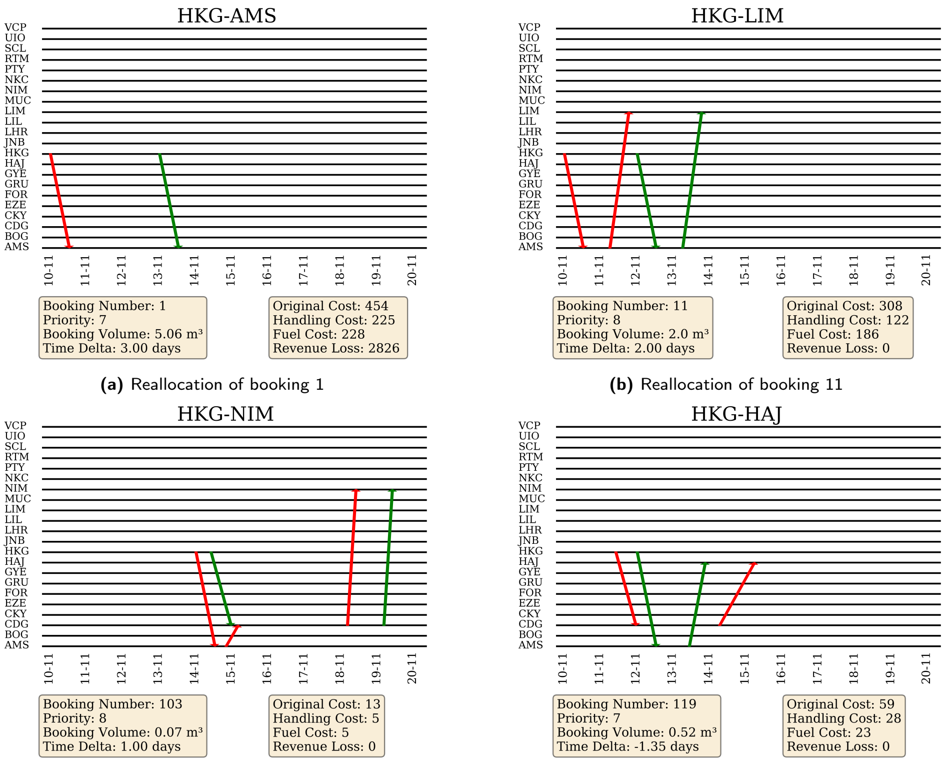
ULD Number	ULD Type	Booking [Booking number: Volume in m^3]	Total
1	Pallet	[1.1: 7], [7: 0.58], [9: 1.09], [16: 0.02], [18: 0.09], [20: 0.27], [21: 0.05], [22: 0.52], [30: 0.06]	9.68
2	Pallet	[2.1: 10]	10
3	Pallet	[4: 9.14], [26: 0.64]	9.78
4	Pallet	[3: 0.01], [5: 0.10], [12: 5.36], [13: 0.01], [14: 0.03], [15: 0.02], [19: 0.27], [23: 3.71]	9.51
5	Pallet	[17: 8.95], [24: 0.98]	9.93
6	Pallet	[10: 1.88], [25: 6.64], [31: 0.93]	9.45
7	Pallet	[27.1: 10]	10
8	Pallet	[6: 1.47], [8: 0.22], [11: 0.97], [28: 0.26], [29: 0.07], [1.2: 7]	9.99
9	Pallet	[2.2: 10]	10
10	Pallet	[27.2: 10]	10
11	Pallet	-	0
12	Bulk	-	0
13	Bulk	-	0

5.5 Computational Results

The following solutions were all obtained using a computer with an i5-6300U 2.4GHz 2 Core processor, which was supported by 8GB of RAM. On this computer the software used was Python v3.7 in combination with Gurobi optimizer v9.

5.5.1 Case 1: Flight cancellation for inbound flight

When considering a flight cancellation for an inbound flight, the different destinations for bookings cause a large set of itineraries. With a time frame of the problem ranging from 10-11-2019 to 21-11-2019, a set of 1,922 arcs is used to create 459,758 itineraries. When running the model the objective function value found was €32,763.15,-, with a runtime of 58 minutes. Due to the large time horizon and great amount of equivalent flights it was possible to divide fairly evenly the bookings among the set of alternative itineraries. This led to an additional operational cost of €-495.04,-. Often bookings that used more than two flight legs were reallocated to an itinerary using fewer flight legs. The high overall costs came from the revenue loss, which added up to €33,258.18,-. The high value is due to the binary implementation of the revenue loss. When a booking arrives later than its initial arrival time, the revenue loss is incurred. In the case of a flight cancellation to an inbound flight it is difficult to find another itinerary that arrives earlier. Especially when the disruption occurs very shortly before departure. In total 44 bookings were reallocated, of which 22 incurred a revenue loss. The other bookings were either non priority or arrived earlier than their initial itinerary. Four different type of reallocation possibilities were used to solve the problem and are displayed in Figure 5. In Figure 5, the red arrows are the initial itineraries and the green arrows are the new itineraries. Where each line indicates a different airport and the x-axis indicates the time. The first reallocation possibility is seen in Figure 5a, where a booking is reallocated to a different equivalent flight, arriving later at the destination and also incurring into a revenue loss. The second reallocation approach is illustrated in Figure 5b, where the booking arrives later but is a non-priority booking and therefore no revenue loss is incurred. In the third situation, the booking initially uses a cost inefficient itinerary with a transfer between the hubs. The new solution selects an itinerary where the booking only uses two arcs, which is displayed in Figure 5c. For the last reallocation option in Figure 5d, the booking uses a new itinerary that relies on the other hub and additionally arrives earlier. As a consequence, no revenue loss is incurred.



(a) Reallocation of booking 1

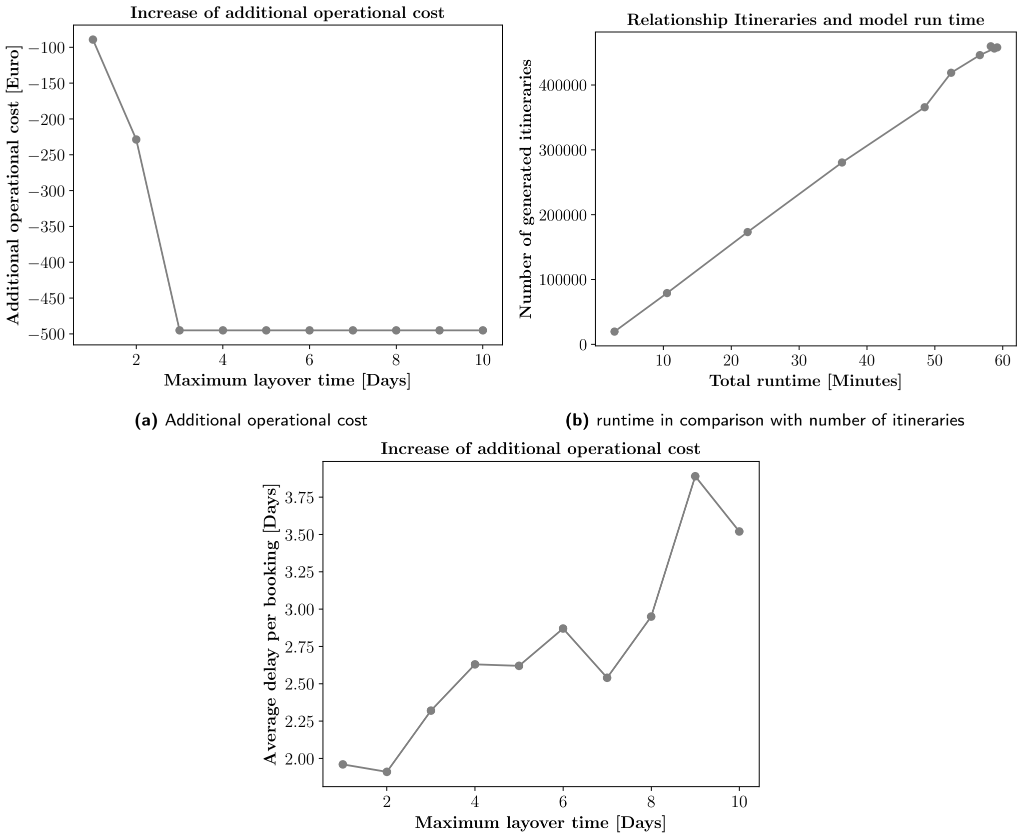
(b) Reallocation of booking 11

(c) Reallocation of booking 103

(d) Reallocation of booking 119

Figure 5: Booking reallocation options

Due to the large size of the problem the runtime of the model takes 58 minutes. Following the procedures of the airline, the decision making of the recovery actions must take 10 minutes. By executing two sensitivity analyses, it is possible to assess which factors influence the runtime of the model and magnitude of the objective function value. The first factor that is varied is the maximum layover time between two arcs. The maximum layover time is varied from a maximum of 1 day layover time to a maximum of 10 day layover time. Although the amount of itineraries is decreasing by changing the maximum layover time, it is only marginally decreasing for the cases with a maximum layover time between 6 and 10 days. For these cases, the quantity of different itineraries changes from 459,758 to 418,875. As seen in Figure 6b, the quantity of itineraries that must be generated is proportionally linked to the runtime of the complete model. Therefore, when analysing the cases with a maximum departure time between 6 and 10 days, the total runtime also only decreases from 58 minutes to 52 minutes. Due to the time horizon of 11 days of the problem, most of the generated feasible itineraries have layover times smaller than 6 days. As a consequence, when varying the maximum layover time from 5 to 1 day the quantity of generated itineraries decrease from 365,706 to 19,636 itineraries. Linked to that, the runtime decreases from 49 to 2.82 minutes. The variation of the maximum layover time does not have any influence on the revenue loss. For every case the revenue loss remains €33,258.18,-. This is due to the binary application of the revenue loss function. Bookings that arrive later will still arrive later because otherwise there would have existed an itinerary to make the booking arrive earlier in the largest case too. Bookings can only arrive earlier if the new itinerary has short enough layover time to beat the initial arrival time, which means the bookings that arrive earlier will arrive earlier in every case. The variation in the objective function value is fully determined by the additional operational cost value. This also remained the same value up to a maximum layover time of 3 days, after which it increased drastically, as seen in Figure 6a. However, for each case the additional operational cost remained negative. The model was able to find a more efficient solution even in the situation of a disruption. A value that is not included in the additional cost is the inventory costs. Preferably the bookings will remain at each warehouse as short as possible, this is also linked to the difference of the time of arrival between the initial and the new itinerary of each booking. If this delay is large it would imply that the booking was situated at one of the warehouses for a longer amount of time before being transported. The delay between the initial and new arrival of a booking increases when increasing the maximum layover time. The average delay per booking per case is displayed in Figure 6c. Therefore, from a cost perspective a maximum layover time of 3 days would be the most preferred solution because of the low additional operational cost and shorter average delay per booking. From a recovery point of view, the runtime of 22 min for this case would take too long.



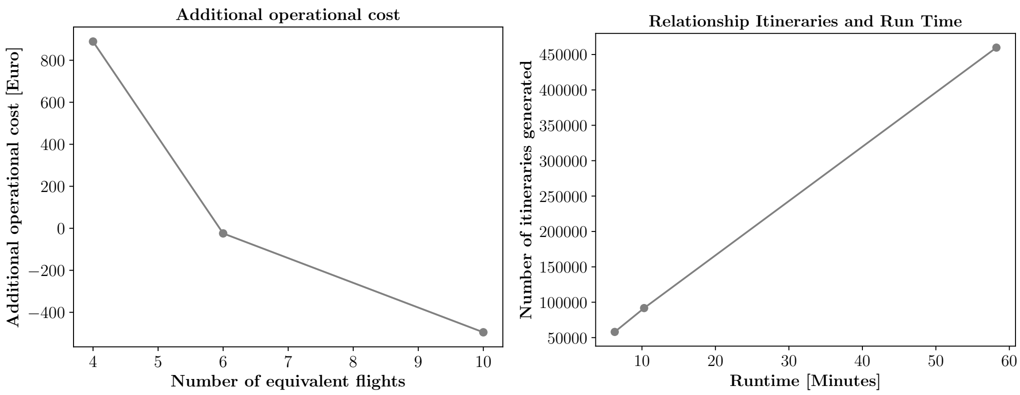
(a) Additional operational cost

(b) runtime in comparison with number of itineraries

(c) Increase of arrival time difference per booking

Figure 6: Sensitivity analysis by varying the maximum layover time

For the second sensitivity analysis, the number of equivalent flights was varied and along with it the amount of bookings. When removing or adding equivalent flights, the time frame of the problem increases or decreases. The time frame can be decreased until there is not enough capacity to allocate all the bookings. By removing equivalent flights from the problem, the overall problem size decreases. However, this could lead up to bad solutions if not enough flights are available to spread out the bookings from the disrupted flight. This is the case, when solving the problem with a total of four flights including the disrupted flight. From the other two cases, with a higher quantity of equivalent flights, the additional operational cost is below €0,-, which can be seen in Figure 7a. For all the cases the revenue loss remained the same and equaled €33,258.18,-, exactly like the layover variation cases. For the case with four flights, the additional operational cost is €889.16,-. In Figure 7b the relationship between runtime and itineraries is confirmed again. Per case, the number of itineraries that are generated decreases due to the decrease in bookings and the decrease in the number of flights, as seen in Table 6. As a consequence of the smaller problem with a smaller time horizon the average delay per booking also decreases from 3.52 days to 1.2 days. Lastly, when the number of bookings that must be reallocated does not change, but the number of equivalent flights does, the average load factor per equivalent flight will increase more. When looking at the complete problem with 10 equivalent flights in Table 6, the volume load factor is 39%, which is much lower than the 85% of the smallest case.



(a) Additional operational cost **(b)** runtime in comparison with number of itineraries

Figure 7: Sensitivity analysis by removing equivalent flights

Table 6: Booked volumes and weights before and after recovery

Number of equivalent flights	Average arrival time difference	Number of bookings	Number of arcs	Total time horizon	Initial Volume Load Factor	New Volume Load Factor	Initial Weight Load Factor	New Weight Load Factor
	[days]	-	-	[days]	[%]	[%]	[%]	[%]
10	3.52	144	1,922	11	34.4	39	38	43.5
6	2.31	129	1,234	7	52.3	66	54.4	67.2
4	1.2	114	1,158	6.5	68	85	72.6	88.1

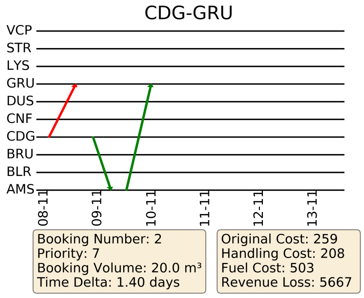
5.5.2 Case 2: Aircraft swap for outbound flight

As a result of the disruption on an outbound flight with a time frame of 5.7 days and a set of 748 arcs, the total amount of itineraries that were generated and usable was 7,574. Although there are 185 bookings, a lot of the bookings are already situated at one of the hubs. Therefore, most of the bookings have the same origin and destination. The similarity of origin and destination causes a smaller amount of itineraries compared to the aircraft cancellation case. Each booking is matched with a set of feasible itineraries. The additional cost that is incurred due to the disruption is €6,721.75,-, which was found in 2.4 minutes. However, the actual operational additional cost only adds up to €1,053.95,-. The remaining cost comes from the revenue loss of one booking with priority, seen in Figure 8a. This booking was in one category above general cargo, but is characterized by a high weight, which added up to a revenue loss of €5,667,-. Initially, the booking was scheduled to arrive on 09-11-2019 and was rescheduled to arrive at 10-11-2019 at the destination GRU. All the other bookings either did not arrive later than the initial itinerary or were not priority bookings and it was not an issue to arrive later. In total, 21, of the 185, bookings were reallocated to a different itinerary. Figure 8b and Figure 8c show examples of bookings arriving on time and do not have a revenue loss or arrive later but are low priority and also do not have a revenue loss. In some cases the new schedule implied an earlier arrival than the initial schedule. This occurs as example in Figure 8d where a booking from a two arc schedule is reallocated to a schedule with one leg. In total 12 bookings arrived later than their initial itinerary, all but one were general cargo bookings. The delay per booking due to the new solution varied from -1 to 1.4 days.

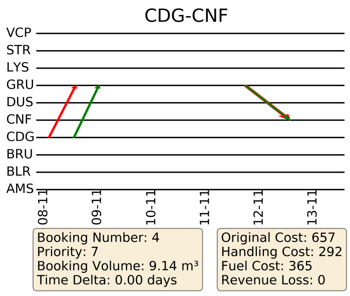
In Table 7 we report the new and old load factors of the disrupted flight and equivalent flights. When comparing the load factors of the volume and the weight, it is clear that the volume is the most constraining factor. Furthermore, in this case flight 3 was overbooked too and a lot of the bookings from flight 1 and 3 were moved to flight 4 and 5.

Table 7: Booked volumes and weights before and after recovery

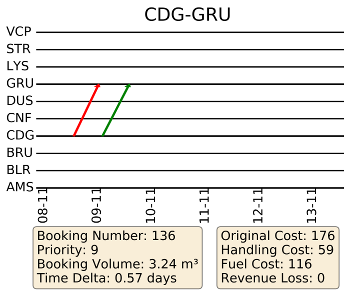
Flight	Initial Volume Capacity	Initial Volume Booked	New Volume Capacity	New Volume Booked	Initial Weight Capacity	Initial Booked Weight	New Weight Capacity	New Booked Weight
	$[m^3]$	$[m^3]$	$[m^3]$	$[m^3]$	$[kg]$	$[kg]$	$[kg]$	$[kg]$
1	112	98.34	63	62.95	31,385	15,127.1	13,872	11,264.7
2	112	108.37	112	111.98	31,222	25,668.9	31,222	26,374.3
3	112	135.7	112	111.97	31,857	25,148.71	31,857	24,558.11
4	119.4	79.66	119.4	111.36	24,704	14,060.5	24,704	16,266.9
5	115.1	89.55	115.1	113.36	24,477	18,954.4	24,477	20,495.6



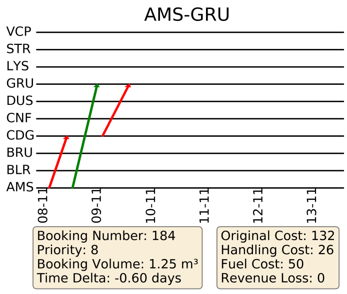
(a) Reallocation of booking 2



(b) Reallocation of booking 4



(c) Reallocation of booking 136



(d) Reallocation of booking 184

Figure 8: Booking reallocation options

5.5.3 Case 3: Aircraft swap for outbound flight with ULD configuration

In a situation where all the ULDs are already configured the model will preferably open the smallest amount of ULDs because of the added cost. This was reflected in the quantity of bookings that were reallocated, namely 14 bookings compared to the 21 of the previous case. With the additional ULD configuration the model took 3 minutes to solve the complete problem, with an objective function value of €15,196.23,-. The main reason for the higher cost is due to the reallocation of booking 25. When the configuration is unknown the only capacity constraint is linked to the total capacity of the aircraft. When the ULDs must be configured, it adds additional capacity constraints. Booking 25 is a priority booking and has a high priority (category 5), but also a large volume 6.64m^3. As a consequence, the solution of the problem without the ULD configuration applied to this case would be infeasible. Additionally, this is also linked to the fact that the model cannot further divide the bookings that were already split into part shipments. With a configuration of 6 pallets it becomes impossible to assign booking 1.1, 1.2, 17, 12, 25, 27.1 and 27.2, as these all take up more than half a pallet. The revenue loss created by reallocating booking 25 is €8,244.6,-. When subtracting this value from the objective function value the additional cost that remains is €6,951.63,-. Also included in this value is the reallocation of booking 2, that uses exactly the same itinerary as in the case without ULD configuration. For this reallocation, ULD 2 and 9 are left unchanged and completely reallocated to an equivalent flight, which does not cause any additional operational cost. The actual additional operational cost equals €1,283.83,-, which is only €229.88,- more than the non-ULD configuration case.

Included in the additional operational cost is the cost of opening three pallets, this adds up to €150,-. The ULDs that were opened are ULD 3, 4 and 6. ULD 4 was only opened to take out bookings and add bookings from other ULDs. ULD 3 and 6 were completely emptied and removed from the flight. The other ULDs that were removed from the flight were ULD 2, 9 and 11. ULD 11 did not contain any bookings, causing no additional cost of removing the ULD. The complete reconfiguration of the ULDs can be seen in Table 8.

Table 8: ULD load configuration

ULD Number	ULD Type	Booking [Booking number: Volume in m^3]	Total	Action
1	Pallet	[1.1: 7], [7: 0.58], [9: 1.09], [16: 0.02], [18: 0.09], [20: 0.27], [21: 0.05], [22: 0.52], [30: 0.06]	9.68	remains unchanged
~~2~~	Pallet	[2.1: 10]	10	Moved to flight 5
~~3~~	Pallet	~~[4: 9.14]~~, ~~[26: 0.64]~~	9.78	Is opened and removed
4	Pallet	[3: 0.01], [5: 0.10], [12: 5.36], [13: 0.01], [14: 0.03], [15: 0.02], [19: 0.27], ~~[23: 3.71]~~, **[10: 1.88]**, **[26: 0.64]**, **[31: 0.93]**, **[165: 0.61]**	9.86	Is opened
5	Pallet	[17: 8.95], [24: 0.98]	9.93	Remains unchanged
~~6~~	Pallet	~~[10: 1.88]~~, ~~[25: 6.64]~~, ~~[31: 0.93]~~	9.45	Is opened and removed
7	Pallet	[27.1: 10]	10	Remains unchanged
8	Pallet	[6: 1.47], [8: 0.22], [11: 0.97], [28: 0.26], [29: 0.07], [1.2: 7]	9.99	Remains unchanged
~~9~~	Pallet	[2.2: 10]	10	Moved to flight 5
10	Pallet	[27.2: 10]	10	Remains unchanged
~~11~~	Pallet	-	0	Removed
12	Bulk	-	0	Remains unchanged
13	Bulk	-	0	Remains unchanged
14	Bulk	-	0	Remains unchanged

6 Conclusion

In this paper we presented, to the best of our knowledge, the first recovery model that specifically focuses on air cargo allocation for combination airlines. The model is called the Air Cargo Allocation Plan Recovery Model (ACAPRP) and can cope with disruptions both on the demand and the supply side by reallocating bookings to new itineraries. In terms of temporal dimension, the model can both cope with bookings that have not been configured into Unit Load Devices (ULDs), or with ULDs that are already configured if the disruption occurs close to the take-off time.

In order to keep the computational burden within a reasonable range, given the recovery nature of the problem and the necessity to have a recovery scheme that can be implemented, a selection of bookings is selected whose itinerary uses the same flight as the disrupted one. Bookings whose itineraries can be entirely modified are divided between the disrupted flight and each equivalent flight. Where an equivalent flight is a flight with the same origin-destination combination as the disrupted flight. These bookings, and their origins/destinations, are also used in a pre-processing step to create all the feasible itineraries using a math-heuristic. Then these itineraries are used as input to formulate the ACAPRP as a mixed integer linear program. The objective is to minimize the additional operational cost and revenue loss by reallocating bookings onto new itineraries. A revenue loss is incurred when a booking is high priority and the new itinerary arrives later than the initial itinerary.

Three disruption cases from a European airline are used to validate the model. The airlines hub & spoke network is built around two hubs, creating more options for different itineraries with the same origin and destination. Given the proximity of the two hubs, ground transportation between them is also considered to add flexibility in the generation of the available itineraries.

The first case is a flight cancellation on an inbound flight from HKG to the hub AMS. A large set of equivalent flights is initially used to solve the problem with a accurate solution in 58 minutes. The large set of flights causes a large set of bookings and a large time frame of the problem. This has two downsides, the runtime is long and the amount of bookings that must be reallocated is large. Two sensitivity analyses are executed to control the runtime and the amount of reallocated bookings. The maximum layover time between arcs is decreased and the amount of equivalent flights is decreased. In both cases, the runtime decreases and a direct link can be made between the runtime and the amount of generated itineraries. Adding that the quality of the solution does not deteriorates too much, hence there is a good trade-off.

The second case considers an aircraft swap on an outbound flight from CDG to GRU. When an aicraft swap is made, the capacity for cargo on that flight can decrease. In this case, the volume capacity decreased from $112m^3$ to $63m^3$. The model must therefore assess what bookings remain on their initial itinerary and what bookings are reallocated. The achieved solution is found in 2.4 minutes, which is a runtime that is compatible with a recovery model. The revenue loss takes up a substantial amount of the overal cost, while operational costs do not play a major role in the objective. This is caused by a high priority booking that is delivered late, and on the associated high priority.

The third case is an addition to the second case, where the allocation of bookings to ULDs was artificially generated, since the airline did not have the required data to create a case with ULD configuration. Hence, the previous case was selected and the bookings on the disrupted flight were each configured into a ULD. The model was run in a similar time as the previous case but the revenue loss increased. The increase in revenue loss was caused by the configuration of the individual ULDs. The same solution obtained for the case without ULD configuration was no longer feasible because not all the bookings could fit in the available ULDs at the same time.

This work presents revenue loss as a method to make sure priority bookings arrive on time. However, this is implemented in a binary way. Either the revenue loss is incurred or it is not. A potential extension of this work is to implement a penalty function that is, either linearly or with a different function, to the delay time of the new itinerary. This will force the model to make sure every booking arrives as early as possible. Furthermore, the cost of new itineraries is determined based on the fuel cost and the station handling cost. Another improvement could be to add an inventory cost, which will also steer the solution towards earlier arrivals of bookings.

Further extensions to the research could be to build a complete MILP based on a time space network. In this case, the pre-processing step to generate the itineraries is not necessary. However, this will increase the problem size.

Lastly, the model does not take into account that a booking can be composed of multiple shipments. A booking can actually consist out of multiple pieces that can be loaded onto different ULDs. From a business perspective it is not preferable to split the booking too much, but it is more realistic to add this possibility.

We believe this model can be of great relevance for combination airlines to better manage their cargo operations in the occurrence of disruptions (to solve the infeasibility caused by the disruption) as well as for normal operations (to improve load factors or choose better itineraries that manual matching might miss).

References

Amaruchkul, K., Cooper, W. L., and Gupta, D. (2007). Single-leg air-cargo revenue management. *Transportation Science*, 41(4):457–469.

Angelelli, E., Archetti, C., and Peirano, L. (2020). A matheuristic for the air transportation freight forwarder service problem. *Computers & Operations Research*, 123:105002.

Archetti, C. and Peirano, L. (2019). Air intermodal freight transportation: The freight forwarder service problem. *Omega*, page 102040.

Bisaillon, S., Cordeau, J.-F., Laporte, G., and Pasin, F. (2011). A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR*, 9(2):139–157.

Delgado, F., Sirhan, C., Katscher, M., and Larrain, H. (2020). Recovering from demand disruptions on an air cargo network. *Journal of Air Transport Management*, 85:101799.

IATA (2020). Iata annual review 2019. `https://www.iata.org/en/publications/annual-review/`. Accessed: June 2020.

Khaled, O., Minoux, M., Mousseau, V., Michel, S., and Ceugniet, X. (2018). A multi-criteria repair/recovery framework for the tail assignment problem in airlines. *Journal of Air Transport Management*, 68:137 – 151. JATM-Multiple Criteria Decision Making in Air Transport Management.

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

Morrell, P. (2011). *Moving Boxes by Air: The Economics of International Air Cargo*. Ashgate.

Novianingsih, K., Hadianti, R., Uttunggadewa, S., and Soewono, E. (2015). A solution method for airline crew recovery problems. *International Journal of Applied Mathematics and Statistics*, 53:137–149.

Sandhu, R. and Klabjan, D. (2006). Fleeting with passenger and cargo origin-destination booking control. *Transportation Science*, 40(4):517–528.

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

Sinclair, K., Cordeau, J.-F., and Laporte, G. (2014). Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233(1):234 – 245.

Sinclair, K., Cordeau, J.-F., and Laporte, G. (2016). A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. *Computers & Operations Research*, 65:42 – 52.

Tang, C.-H., Yan, S., and Chen, Y.-H. (2008). An integrated model and solution algorithms for passenger, cargo, and combi flight scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 44(6):1004 – 1024.

Wada, M., Delgado, F., and Pagnoncelli, B. K. (2017). A risk averse approach to the capacity allocation problem in the airline cargo industry. *Journal of the Operational Research Society*, 68:643–651.

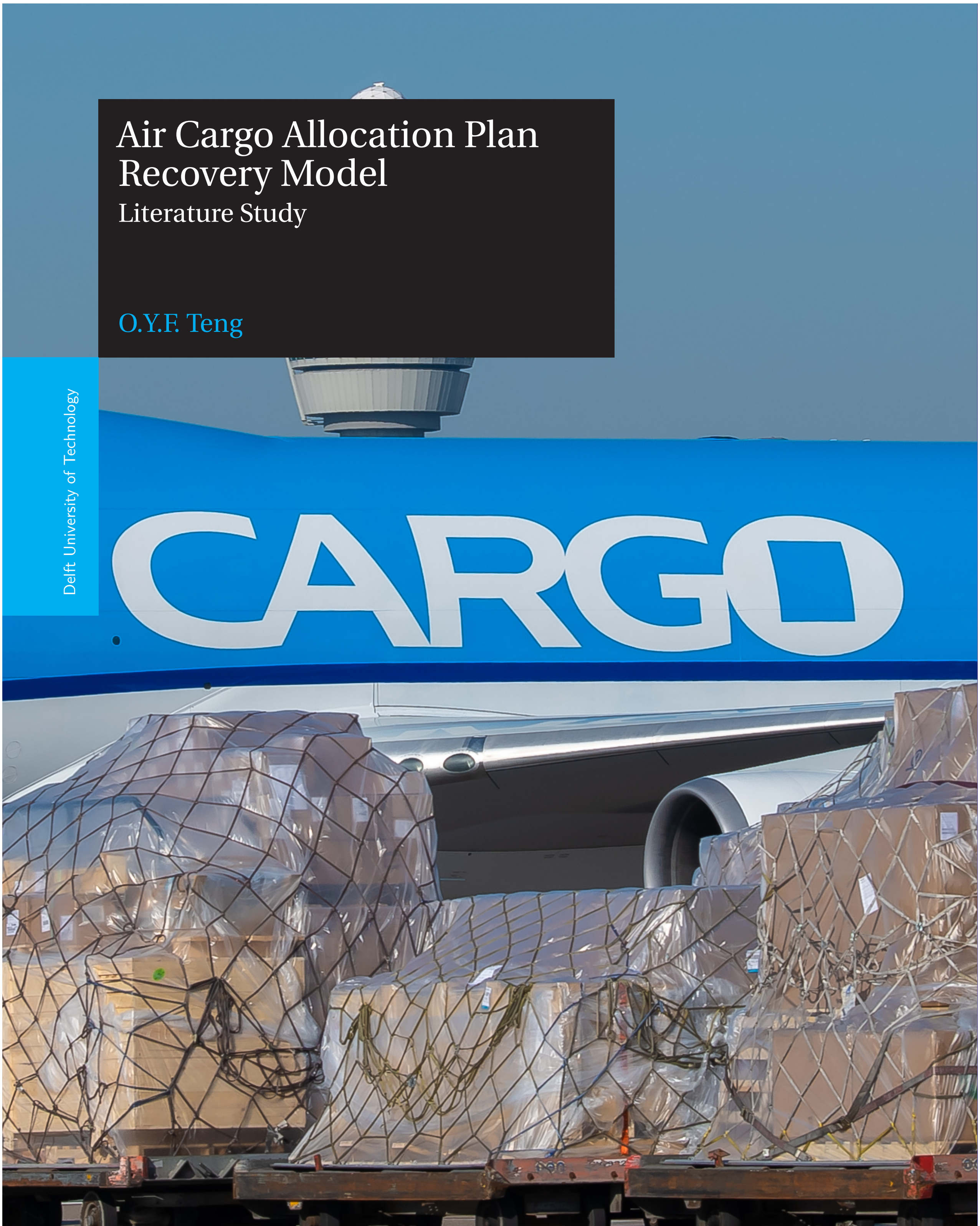
A
Appendix A: Literature Study (Previously graded under AE4020)

Air Cargo Allocation Plan Recovery Model

Literature Study

O.Y.F. Teng

Delft University of Technology



TUDelft Delft University of Technology

Challenge the future

Air Cargo Allocation Plan Recovery Model

Literature Study

by

Olivier Teng, 4735935

In partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology

Project Duration:	6 September 2019 - 3 December 2019	
Thesis Supervisors:	Dr. B.F. Lopes dos Santos	Supervisor
	Dr. A. Bombelli	Daily Supervisor

Abstract

Air cargo is a type of transportation which is far from conventional, both compared to other means of cargo transportation and to passenger flights. The possibilities of last minute changes to the initial capacity create disruptions. At the moment of writing, no recovery model has been created to solve these disruptions. As a foundation for a recovery model the relevant literature is studied. The literature that is reviewed take the overall process of air cargo airlines, the general bin packing problem, packing requirements for air cargo, the uncertainties that arise, existing bin packing models and recovery models from other fields into consideration. Finally, the investigation creates a scope on how to initiate the development of a recovery model.

List of Abbreviations

1D-BPP	One Dimensional Bin Packing Problem
2D-BPP	Two Dimensional Bin Packing Problem
3D-BPP	Three Dimensional Bin Packing Problem
ACLPP	Air Cargo Load Planning Problem
ACP	Aircraft Configuration Problem
AFKLMP	Air France KLM Martinair Cargo
AoD	Airport of Destination
AoL	Airport of Leaving
APP	Air Cargo Palletization Problem
ATA	Air Transport Association
ATFFSP	Air Transportation Freight Forwarder Service Problem
BSP	Build-up Scheduling Problem
GA	Genetic Algorithm
MBSBPP	Multiple Bin Size Bin Packing Problem
OCC	Operations Control Center
SA	Simulated Annealing
TS	Tabu Search
ULD	Unit load device
WBP	Weight and Balance Problem

CONTENTS

1
Introduction

1.1. Problem Definition

Air cargo is a type of transportation which is far from conventional, both compared to other means of cargo transportation and to passenger flights. For this reason, general models cannot easily be applied and other solutions must be developed. One of the main differences is the uncertainty in the expected cargo to deliver. For example, at Air France KLM Martinair Cargo (AFKLMP) it is still possible to book cargo up until shortly before take-off, making it hard to assess beforehand how much cargo is actually booked for a flight. The logical solution is to overbook flights, which can create the possibility of assigning too much cargo to the aircraft. These uncertainties do not only arise from the demand side, but also from the supply side. For instance, it is not known beforehand how much space is allocated for cargo in the aircraft. The reason being the fluctuations in room required for the baggage of the passengers. A more impacting change from the supply side are last minute changes in the aircraft type, as these are determined by the passenger side of operations. Leaving the choice which cargo will and will not be loaded onto the aircraft. Furthermore, even if the cargo is assigned to a certain flight the loading strategy still remains unknown.

The loading strategy opens another difficulty of air cargo transportation. From a packing perspective, air cargo distinguishes itself compared to other means of transportation because of the multiple shapes of the loading devices and the extensive amount of constraints. For example, orientation, stability and fragility of the items. Combining these extra constraints will make the, already challenging, general bin packing problem even more difficult in an air cargo situation. Beyond that, it is important to bear in mind that the packing problem can only be solved once the dimensions of the packages are known, which is short before take-off, making computational time a great issue.

If the cargo cannot be allocated to the preferred flight, it must be rescheduled to another flight. How to determine which secondary flight has enough capacity, and is the best option, is another unknown.

1.2. Research Objective

The objective of the research is to create a recovery model for the disrupted cargo by assessing the possibility of repacking the initial aircraft and/or selecting the correct flight on which it can be loaded. Disruptions are considered as cargo that was initially planned on a specific flight, but could last minute not be loaded because of uncertainties. This can range from one pallet that must be reconfigured because it won't fit, to a complete shipment that must be reallocated due to the cancellation of a flight. As this is a very volatile time window in the process, it could mean that the best solution changes after a short period of time due to new information. Therefore, the model will not be solved only once when the disruption occurs, but multiple times in a certain time period. Optimally speaking the model is executed as late as possible, however that would be impossible because the solution still has to be executed. This creates a limit to the amount of times a new solution can be created and the time it takes to run the model. Making time a key factor in creating the recovery model, as the disruptions are not known up until shortly before take-off.

2
Air Cargo Operations

Given the unique features of the air cargo transportation system it is easy to imagine they have an impact on the general process. This is clarified by creating an overview of the process as a whole. Starting at the booking of the order, and ending with the delivery of the item at the final recipient, also referred to as the consignee. When this process is elaborated, it becomes clear at which point the recovery model is applied and how it will influence the preceding or following steps. Additionally, it will become easier to visualize where adjustments can and cannot be made, which is key in the creation of the recovery model. Before this can be explained, it is important to have an understanding of the equipment that is used for air cargo.

2.1. Unit Load Devices

To reduce the effort to pack the cargo into the aircraft, the individual items are packed into larger containers or onto pallets. In air cargo transportation these containers and pallets are referred to as unit load devices (ULDs).

2.1.1. Containers

An air cargo container distinguishes itself from a regular box because it is specifically designed for use in an aircraft, taking into account the curvature of the fuselage. It can be compared to a rectangular box from which one edge, or two, has been cut out, as seen in Figure 2.1. The dimensions of containers are fixed and each is identified with its own Air Transportation Association (ATA) code. Thus, in support of standardization, a finite amount of container types in different sizes and shapes are available and used by the airlines. Considering the fact that the model will be used for cargo placed in the belly of passenger aircraft, only ULDs designed for the belly will be incorporated in the model. The ULDs available for placement in the belly are: LD-1, LD-2, LD-3, LD-6, LD-8, LD-9, LD-11, LD-26, LD29 and LD-39 [Uldcare, 2019].

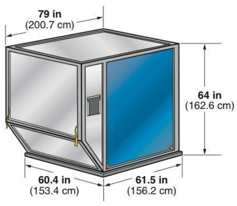


Figure 2.1: *Lower deck container example LD-3*

2.1.2. Pallets

Besides containers it is also possible to load packed pallets onto the aircraft, which have to meet certain requirements. When they are loaded in the warehouse, the items placed on the pallet must not exceed specified dimensions, meaning they have virtual borders. Once a pallet is filled to a desired level it is enclosed under a plastic sheet and a net. When this step is completed the pallet must remain untouched until it is to be unpacked at the destination.

2.2. PROCEDURE

To get a clear view of the procedure used to ship air cargo, an interview was executed with an AFKLMP employee [Zwitzer, 2019].

The first step in the process is when the order is placed by the client. At the placement of the booking, the process of air cargo already distinguishes itself from the process in passenger flights. In order to complete a booking for a passenger ticket the client must pay upfront, for the booking of air cargo this is not the case. Therefore, the clients will most likely place multiple orders, with different airlines, for the same shipment and then choose the best option. The only information the client is obliged to communicate is an estimation of the weight and volume of the shipment. The order is then booked into the system for a particular flight. The earliest a booking can be placed is 14 days before the take-off of the flight. Another difference in the process of air cargo compared to passenger flights, where clients can easily book up to 6 months before flight.

As a consequence of the uncertainties behind the size of the orders an analysis is made estimating the actual size of the shipments. For example, regular clients will probably have a constant type of order that is placed every month. Therefore, it will be easier to take into account what their order will be. The exact order will actually remain unknown up to shortly before the take-off. Using this information the quantity of necessary ULDs for a specific flight is determined. The most recent estimation of the revenue management is handed over to the operations at a short time span before take-off, this is referred to as the handover point. It is important to mention that this does not mean the booking of orders has been stopped. Even after the handover point clients can book orders to fly that flight. In the mean time operations has already started with packing the ULDs at the warehouse.

Finally, just before take-off all ULDs must be packed and ready to be loaded onto the aircraft. If unexpected disruptions occur a choice is made by operations to repack a certain ULD or move the cargo onto another flight.

2.3. PLANNING

The planning of efficiently transporting the booked shipments is not considered as one specific process, but as multiple. Brandt and Nickel [2018] has a very extensive review on the complete Air Cargo Load Planning Problem (ACLPP) and has standardized the terminology used. Each of the individual parts answers a question required to complete the planning. The four decision questions used are:

1. What ULDs should be built?
2. When should the ULDs be built?
3. How to pack items on the ULDs?
4. Where to load the ULDs inside the aircraft?

The first step in the ACLPP is the Aircraft Configuration Problem (ACP), where the quantity of ULDs and which type are determined for a specific flight. This amount is based on the capacity that is expected to be available on the flight. This will answer the **what** in the four decision questions.

The next step is the Build-up Scheduling Problem (BSP). When it is known how much ULDs must be packed, it can be determined **when** each ULD is built in the warehouse.

One of the most difficult parts of the ACLPP is the Air Cargo Palletization Problem (APP). In the third step the individual items are assigned to a ULD. Depending on the objective function the optimization will assign the items to ULDs based on the best distribution. Which answers the question on **how** the items should be packed. Often three dimensional bin packing models are created to do this optimally, but these are very difficult to solve within a the desired amount of time.

Once the items are placed in the ULDs the last step is executed, the placement of the ULDs in the aircraft itself. This step is referred to as the Weight and Balance Problem (WBP). The fuel efficiency of aircraft is partly determined by the position of the center of gravity [Brandt and Nickel, 2018]. For a cargo flight the loading positions of the ULDs in the aircraft can have a big influence on the center of gravity making this an important step. This step is also important for safety reasons because each aircraft has a maximum take-off weight which cannot be exceeded. This last step answers the **where** decision question.

When considering the recovery model the ACP and BSP have already been completed, which means last minute adjustments have to be made in the APP and WBP.

3
Bin Packing Problem

The optimal way of packing items into containers has been researched for multiple decades. In the field of operations research this problem is referred to as the bin packing problem. Given a set of items and a finite number of bins, the goal is to use the least amount of bins to fit in as many items as possible [Korf, 2002]. While it sounds like a simple problem to solve, it is actually a computationally heavy problem to solve optimally.

For every single bin packing problem it is proven they are NP-hard [Korte and Vygen, 2006]. As a consequence, it is possible to solve the problem for small instances, but when increasing the problem size the time to solve does not increase linearly.

The bin packing problem can be solved in multiple dimensions. The most basic version of the bin packing problem is the one dimensional problem. An example of the one dimensional bin packing problem (1D-BPP) is to load items, from which only the volume is known, into bins. This would work with liquids because they do not have specific dimensions. The only information known is the fixed volume of the bins and the volumes of the individual items. The bins have a volume of 100 and the set of items {5,12,25,42,54,58}. The sum of the individual items is 196, meaning at least 2 bins are required. A possible solution is to put {42,58} into the first bin and {5,12,25,54} into the second bin utilizing the least amount of bins possible. This is similar to the knapsack problem in which one bag is used and a maximum amount of weight [Kellerer *et al.*, 2004].

Apart from filling up bins with liquids, or with data, the 1D-BPP does not have a lot of practical applications. For instance, when covering a certain surface area with boxes, multiple dimensions are needed. When covering a certain area inside bins, the width and length of the boxes are required in order to determine the best coverage. This changes the problem from a one dimensional case to a two dimensional bin packing problem (2D-BPP) [Ma and Zhou, 2017].

Once items with specific dimensions must fill up bins with defined dimensions, the three dimensional bin packing problem (3D-BPP) must be applied [Martello *et al.*, 2000]. The bin packing problem becomes a lot more difficult when making it three dimensional because more constraints arise in order to make it practically applicable. The addition of the length, width and height and additional conditions make it extremely hard to solve optimally.

3.1. Different loading approaches

Even when considering the 1D-BPP, the computational times can ramp up quickly because of the NP-hardness. For practical applications, optimality is generally sacrificed in favor of sub-optimal solutions that can be computed with a limited computational cost through ad-hoc algorithms. One of these approaches is to apply a heuristic algorithm. As a step up to more complex 3D-BPP heuristics it is interesting to assess loading heuristics for the 1D-BPP. In Rieck [2009] they explain different loading approaches for the 1D-BPP. Using these approaches and fictitious data they are elaborated. All the examples discussed will be using a random loading order that is illustrated in Figure 3.1 and these will be loaded onto bins that have a capacity of 5.

The first and most simple loading heuristic is the next-fit method. The principle is to only look at the current bin to check whether the next item can be placed. If the item does not fit into the bin a new bin is added and the item is placed there. The biggest drawback of this heuristic is that it does not look at the previous bins to check whether they have space available to accommodate the item. An example of the loading of the next fit method can be seen in Figure 3.2. Seven bins were necessary to make sure all the items were packed.

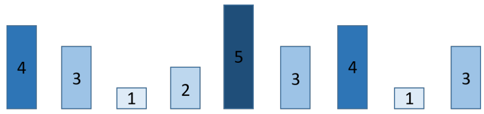


Figure 3.1: *Loading order*

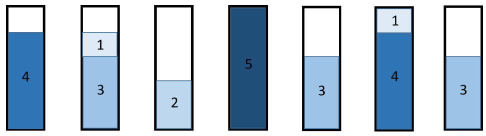


Figure 3.2: *Next fit*

The second and more advanced loading heuristic is the first-fit method. In the first-fit approach the algorithm checks the first bin if the item fits and otherwise continues to the next. It checks each bin consecutively if the item fits in a bin, until the item fits fits or all bins have been checked. If the item does not fit into any of the bins, then a new bin is opened and the item is placed there. For the given loading order the solution using the first fit approach is illustrated in Figure 3.3. Since this heuristic can look back at the previous bins, it can assess more loading options for the selected item. As seen in Figure 3.3 it only needs six bins compared to the seven bins used for the next fit approach.

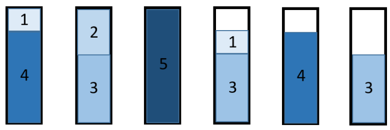


Figure 3.3: *First fit*

The best-fit heuristic also looks at previous bins when choosing where to load the item. This method does not specifically choose the first bin it will fit into, but it will check in which bin it will fit the best. The best bin will be the bin that is filled up as far as possible once the item is placed. If the item does not fit in any of the bins a new bin is opened. The best-fit heuristic is applied to the sample data, the result can be seen in Figure 3.4. As a consequence of filling up the bins when possible, the result will contain bins that have a high utilization and some with a low utilization. For example, in Figure 3.4 four of the bins are fully utilized and two only contain 3 out of the 5 available capacity.

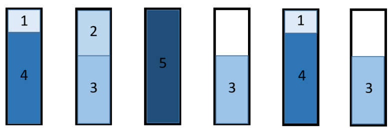


Figure 3.4: *Best fit*

The last placement heuristic is the worst-fit approach. When an item is selected to be placed, each bin is assessed and the bin with the most capacity available is selected. The results of this heuristic can be seen in Figure 3.5. Compared to the best-fit, the worst-fit heuristic has a more uniform distribution of the volume utilized per bin. Depending on the objective of the optimisation different methods are preferred.

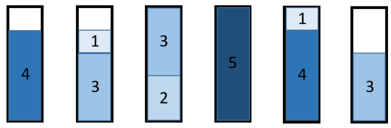


Figure 3.5: *Worst fit*

3.2. LOADING ORDER

The loading sequences used to determine when an item is loaded into a bin are not bounded to the one dimensional bin packing heuristics, but are also used for multi-dimensional bin packing heuristics. The best loading order for a heuristic approach for each instance cannot be defined. However, it is interesting to analyse the different options. The two KPIs mostly used to decide the order, are the weight or the volume of the items. They are selected because these values are used to determine the capacity of the bins. Each of the options can be applied in two different ways. Either in an ascending order, where the smallest item is placed first and the largest item is placed last, or in a descending order, where the largest item is placed first and the smallest item is placed last. As seen in van Aken [2019] there is not a substantial difference in using an increasing or decreasing order of the weight or volume. What can be observed is that volume ordering works better than weight ordering.

Another possibility in loading order is to normalize the volume and weight, to then add these values together. By dividing the volume of the item by the total volume of the bin a normalized value for the volume can be determined. When this is also calculated for the weight of the item and the maximum weight capacity of the bin, these two values can be added. For the different loading approaches this method seems to be the most efficient in van Aken [2019]. Specifically, the high to low ordering got the best results.

Lastly, it is made clear that the most efficient ordering method has not been determined yet. This is reflected in the fact that the random ordering still works as one of the best loading approaches [van Aken, 2019].

4
Requirements for Air cargo allocation

From all the different planning problems, the Air Cargo Palletization Problem is the toughest problem to solve. Therefore, Brandt and Nickel [2018] also points it out as the most prominent part of the ACLPP.

The APP problem can be solved by implementing an advanced 3D-BPP model. As mentioned in chapter 3, the 3D-BPP on its own is already a time consuming problem. When it is applied to solve the APP, it becomes increasingly difficult. This is caused by the specific constraints that must be implemented for an air cargo situation. At this point it is not possible to create a model that solves optimally, for all the requirements, within the available time.

Although it is impossible to implement all the available constraints into one model, it is important to get a grasp of the different possibilities available.

The following two constraints can be found in Pollaris *et al.* [2015].

Stacking: To fill up the ULD as optimal as possible the items must be stacked. Before doing this, each item is assessed whether it is safe to place other items on top. If too much weight is stacked on top of an item it might damage the product. By applying stacking constraints (also denoted as load-bearing strength constraints or fragility constraints) this can be avoided. The maximum pressure an item can hold is referred to as the load-bearing strength. With multiple orientations the maximum load-bearing strength can differ. For example, one side might be stronger than another. When considering fragile items it might be prohibited to stack other items on top, this can also be accounted for in the constraint.

Complete-shipment constraints: When a client places an order for a multiple item shipment, theoretically the items could be shipped individually. However, as this is inconvenient, it can be specified that either the complete order is shipped or none at all.

The following five constraints can be found in Brandt and Nickel [2018].

Item grouping: Similarly to the complete-shipment constraint, when a client has placed an order, it is preferable to keep all the items together. Therefore, the order must be divided over the least amount of ULDs as possible. This reduces the handling procedures and the risk of making mistakes. This is also beneficial for the overall optimization because the items with the same destination will often be packed together in the same ULDs.

ULD weight limit: Even though the main goal is to utilize as much volume of the ULD as possible. It must be taken into account that the maximum weight threshold of the ULD cannot be exceeded.

Item weight limit: For safety reasons, some dangerous substances can have an associated weight limit. For example, dry-ice, radioactive materials, chemicals and explosives.

Item compatibility: Another safety critical constraint is when certain materials are not allowed to be loaded in the same ULD or close to each other on the aircraft.

Item availability: At the workstation where the ULDs are packed the items arrive at different times. Logically the workers do not wait for the correct item to arrive and pack as good as they can. Therefore, the items can only be loaded if they are available at the time that the ULD is packed.

Orientation Stability: It is possible that items are fragile and therefore must be loaded onto the ULD in a certain orientation. Often these items have a fixed vertical orientation and are labelled with a arrow sticker saying "This way up" [Bortfeldt and Wäscher, 2013]. However, it also occurs that the item must be accessed from a certain side, like a pallet. If this is the case the item is assigned a fixed horizontal orientation.

Must Fly: With air cargo some orders are obliged to fly exactly that flight. Even if it is possible to deliver the cargo earlier, it is not allowed. This type of cargo is referred to as must fly cargo. It must fly on that specific flight linked to a specific tail number. An example for must fly conditions is given in Virgin Atlantic Cargo [2016].

Complete ULD: Completed ULDs can be delivered and constrain the possibility of adding more cargo to the ULD. Additionally, the ULD is not to be unpacked either, which leaves fewer possibility to optimize the shipment.

Packing constraint: For a pallet a tolerance exists in which items may protrude slightly from the sides. All items that are packed inside a container are obliged to fit inside the dimensions of the container. This is obvious from a realistic perspective because the containers are closed boxes. However, it must specifically be incorporated in the model.

5
UNCERTAINTIES

As mentioned in Koch *et al.* [2019] and Wu [2011], a lot of uncertainties can cause for a disruption. In order to create an extensive model, that will take the correct choices, it is important to get a grasp of the available disruptions.

5.1. DEMAND SIDE

From the demand side, air cargo works completely different from passenger cargo. The clients of the air cargo companies are not forced to exactly define the size of their booking. This can have two consequences, either the client could show up with no shipment at all or they will show up with more than they initially booked. In the latter case, more shipments are available than the capacity of the aircraft. The reason why this is possible is because the clients will pay after the shipment is delivered. What forwarders will generally do is place orders at multiple cargo companies and select which company gives the best offer. This also means it is a closed market and it is not known what an exact price would be to ship item X from A to B. As the companies that arrange the orders are forwarders themselves, they are unsure how the shipment will precisely look like. The information they will provide, most of the time, is a weight and a volume [Koch *et al.*, 2019]. This causes an uncertainty when considering the actual dimensions of the shipments. More information can only be determined once the actual shipment arrives.

5.2. SUPPLY SIDE

A passenger aircraft has a certain amount of capacity reserved for air cargo [Murph, 2019]. However, it is not a guarantee that this capacity can be allocated to cargo. On passenger flights the passengers have the priority, this also holds for the belly of the aircraft. It could be possible that passengers arrive with a lot more baggage than expected. In this case, the room that initially was given to cargo is now replaced with passenger baggage.

Another disruption from the expected cargo capacity is the change in aircraft type. This is not a choice that the cargo operations has influence on, but it does have large consequences. For example, the lower deck bulk volume capacity for a 777-200 is 162 m^3 and for a 777-300 is 216 m^3 [Modern Airliners, 2017].

For reasons unknown it might also be possible that a flight is cancelled. The decision of cancelling a flight is fully determined by the passenger side of operations, forcing the cargo operations to find new solutions. In that case all the cargo planned for that flight most be relocated to different flights. Luckily, the highest priority of aircraft recovery is to avoid cancelling a flight because this is often the most expensive option. In 2018, 99,093 passenger flights where cancelled in the US, which accounts for 1.83% of the total amount of flights [Bureau of Transportation Statistics, 2019].

6
Bin Packing Models

The goal of the recovery model is to create a generic model that can be tuned to fit different airlines needs. In order to find the correct moves, to recover an initially planned solution, different bin packing models are assessed. These models are chosen because at the time of writing no air cargo recovery models exist in the literature. Therefore, it might be possible to create recovery moves by analysing bin packing models that have an emphasis on time.

6.1. Comparison between mixed integer program with heuristics approach

In section 6.1 a comparison is made between two models which solve the three dimensional bin packing problem for variable bin sizes in a same situations. The first uses a mixed integer approach to solve the problem optimally and the second solves it using a two-phase heuristic. These models are elaborated because they both solve a very elaborated version of the bin packing problem considering the most amount of air cargo constraints.

6.1.1. A mixed integer programming formulation for the three-dimensional bin packing problem

When solving a bin packing problem, the solution of the exact formulation provides the best possible solution, but is not computable within a reasonable time limit. In Paquay *et al.* [2016] the multiple bin size bin packing problem (MBSBPP) is solved, using a mixed integer programming formulation. The applicability regarding the air cargo allocation model and recovery model are assessed.

Including all the available constraints for the air cargo allocation problem is not possible in one model, which means certain sacrifices are made. When comparing different models, as analyzed in Brandt and Nickel [2018], it can be observed that the model described in Paquay *et al.* [2016] is a very complete one. This model accounts for orientation stability, stacking, ULD weight limit and weight distribution.

It is obvious that an optimal solution for the bin packing problem is preferable. This is achieved when applying a mixed integer programming approach, which solves towards optimality. However, this does come with drawbacks because the problem is computationally heavy to solve. As described in Paquay *et al.* [2016], the runtime limit is set for 1 hour, and the model is able to solve up to 1 ULD x 12 boxes within that time. For instances with 1 ULD x 18 boxes the max time is reached and the optimality GAP is still 50% [Paquay *et al.*, 2016]. In this situation, only one type of ULD is available, but when different types are added the complexity of the problem increases. As a consequence, this method is only applicable for small instances with a maximum of around 12 boxes. The minimum amount of items used on actual cargo flights is around 300 items.

Although this model cannot be used for real life scenarios, it can be used as reference for newly developed models that are faster. For example, using heuristics or a combination of integer programming and heuristics.

6.1.2. A TAILORED TWO-PHASE CONSTRUCTIVE HEURISTIC FOR THE THREE-DIMENSIONAL MULTIPLE BIN SIZE BIN PACKING PROBLEM

As mentioned in subsection 6.1.1 a full mixed integer programming formulation is time consuming and only realistic for small cases. The time is therefore the most restricting issue of this model, which opens the door for different solution approaches. In Paquay *et al.* [2018] a two-phase constructive heuristics method is applied. This model incorporates the same objective and constraints as in subsection 6.1.1, which opens the door to directly compare the results and performance of both models. For the two-phase heuristic certain moves are assigned to pack the ULDs. These moves are split up into two phases, in which the first phase is a packing algorithm that packs a set of items into one ULD and the second phase will use packed ULDs to create several multiple-bin situations.

Packing Algorithm, Phase 1:

In the first phase a packing algorithm is created. This algorithm uses designated moves to determine where to place items into a ULD. In the process of placing these items, specific constraints are taken into account. The orientation constraint creates the possibility of placing an item in all the 6 different orientations, but also taking into account that some orientations are prohibited because of fragility of the item. In a ULD, corner points are created as potential positions to place an item. These points are referred to as extreme points. When an item is placed, all the extreme points are assessed using a merit function and the best option is selected. For fragile items it is preferable to not remain any left over top space because no other item is allowed to be placed on top of the item. This process is also split up into two parts. In the first part all the non-fragile items are placed and some of the fragile items. Fragile items are only placed in part I when a maximum amount of top space is surpassed. Once all the items are placed, or put in the remaining fragile item list, part II is initiated. In part II the threshold is removed and all the remaining volume is filled up with the left over fragile items.

Multiple ULDs, Phase 2:

In the second phase the option of having multiple ULD types is assessed. The method is based on the one-dimensional cases used in Kang and Park [2003], but extended to be applicable for a three dimensional case. In the second phase the packing algorithm is used to pack for multiple ULDs. In the first case, all the items are placed in the same ULD type. Once this solution is created, a new solution is created by changing the ULD type of one of the ULDs. The ULD that is switched is selected based on the volume utilized and the overal solution is assessed. This is repeated one by one for all the ULDs and the best solution is selected. Lastly in post processing, a correction for the weight distribution in the ULD is executed.

The biggest drawback of the model in subsection 6.1.1 is the time it takes to even solve for big instances. With the model presented in Paquay *et al.* [2018] the time to solve is decreased drastically, being able to pack 100 boxes in 12 seconds and it still meets up to the same constraints. The model is able to solve for larger instances, but it logically creates packing options that are sub optimal. This is also reflected in the results of the model, as it only achieves load factors of around 50% [Paquay *et al.*, 2018].

6.1.3. REMARKS REGARDING THE COMPARISON OF THE MODELS

Making a comparison between models is only possible if they are solving exactly the same problem, which includes the way the models are setup. For example, the constraints must be configured in a similar manner. When looking at Paquay *et al.* [2016] and Paquay *et al.* [2018], this is the case. Creating the possibility to compare the computational time of the models directly. In general this is already an issue for the 3D-BPP since it is NP-hard. However, for a recovery model this characteristic must be emphasized even more because the men in the workstation do not have time to wait for a solution and are forced to pack quickly to get the cargo on time for take-off. When looking at the computational times for both models, Paquay *et al.* [2016] can solve for instances with a maximum of 18 boxes within an hour and Paquay *et al.* [2018] can already pack 100 boxes in 12 seconds. As time is one of the main factors of the recovery model it can be concluded that further investigations must be taken in the field of heuristics.

Although these methods are very elaborate and are taking into account many of the constraints, they cannot be used as a basis for a recovery model. Both models start from scratch when packing the ULDs, which is different for the case of a recovery model. In that case the model is supplied with a non-feasible solution and will use that solution to create a feasible solution. The bridge is easily made towards improvement heuristics and analysing how these are applied to the 3D-BPP, but also other situations like aircraft recovery models.

6.2. Improvement Heuristics

The main driver behind a recovery model is to improve the situation which is non-feasible at first. In the case of air cargo transport, the initial allocation of the items in ULDs is not possible. A solution must be created to make the allocation of the items feasible again. Although recovery models specifically for air cargo do not exist, it does not mean general bin packing algorithms should be left out. Coming from section 6.1, the overarching idea is to use models that use as input a low-quality or infeasible solution (due to the unplanned disruptions), and that focuses on how to improve such solution. These type of models are referred to as improvement heuristics and in Zhao *et al.* [2016] multiple models are assessed.

6.2.1. Genetic Algorithm

Genetic algorithms (GA) are a traditional approach in heuristics. The first application to the bin packing problem date back to 1994 in Hemminki [1994]. Looking at more recent models Wu *et al.* [2010] makes a comparison between an exact model and a GA model. The authors approach consists of making chromosomes where they consider two factors. The volume and the orientation of each item. In genetic algorithms the chromosome are properties that together create a solution, also known as a population. Each of the chromosomes are ordered using the volume and then a random orientation is added. The reproduction is then executed based on roulette wheel selection. For better solutions the best option at that point is added to the reproduction pool. Furthermore, two mutation forms are applied, a sequence mutation and a position mutation. In the sequence mutation two bits in a chromosome are selected and switched to form a new one. In position mutation multiple bits are selected in the chromosome and rotated to different positions. Lastly, a fitness evaluation is made, in order to choose between available extreme points. The fitness evaluation is calculated using a packing index, which includes the relationship between the volume that the item takes up at that position, the height and the utilized volume up to that point. The extreme point that achieves the highest index is selected for the item.

6.2.2. Simulated Annealing

The only recent approach using simulated annealing (SA) can be found in Mack *et al.* [2004]. The authors approach the bin packing problem for a single bin with multiple models. The first heuristic technique they apply is a simulated annealing approach. This model was initially designed to simulate the physical cooling process of a solid metal. Later it was also applied in combinatorial problems. In general, the first step is to create an initial solution, which does not have to be an efficient one. By changing one of the parameters, a neighbourhood solution is created. This change is based on randomly changing an index in the packing sequence and then defining a new entry for this position. If the solution quality is improved, it will be selected as the new optimal solution. If no improvement is established, a probability is added to still select this solution. This choosing probability is created using the difference of the new solution compared to the old solution and the 'temperature'. The temperature is adjusted every iteration by a reduction function. This essential step is important to make sure the model does not end up in a local minimum. The model appears to work better than a tabu search method. However, the model showed higher complexity in the selection of the steering parameters.

6.2.3. Tabu Search

In Crainic *et al.* [2009] a two stage tabu search (TS) method is applied. The first stage assigns the items to containers and the second stage packs the assigned items in the container. The advantage of this method is that the second stage can be adjusted according to the constraints required, while keeping the first stage unmodified.

With the first level, it is created an initial solution which is used to create a set of solutions called neighbourhood. Every time a better solution is selected, a new neighbourhood is created and extended around that solution. This is repeated for k amount of steps. Within this heuristic, a diversification phase is added in order to escape local minima and explore new regions of the solution space. New neighbourhoods are created using two adjustments, the 1-swap move, in which the items of a bin are swapped with the items of another bin and add-drop move, in which one item of a bin is loaded onto another bin.

6.2.4. Space defragmentation

In Zhu *et al.* [2012], the authors use two improvement techniques which are unique for the bin packing problem. Initially they use an extreme point approach to pack for an initial solution. After creating an initial solution, they start to improve it. The first improvement they make is to look at the disconnected empty volumes in the bin, then they reposition the packed items so they create one larger empty volume that is filled with a new item. This is the defragmentation process and an example can be seen in Figure 6.1. It is clear that enough volume is available to accommodate item number 3. However, because of the placement of item 1 and 2 the empty volumes are disconnected. By defragmenting the items in the bin it is possible to load item 3.

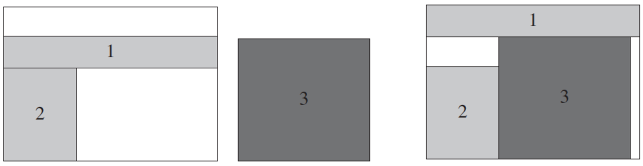


Figure 6.1: *Defragmentation process [Zhu et al., 2012]*

6.2.5. Variable Neighbourhood Search

In Zhu *et al.* [2012], two different improvement heuristics are incorporated. The first is mentioned in subsection 6.2.4, while in the second part they implement a iterated local search method which they refer to as a bin shuffling method. By re-shuffling items from a readily made solution it might be possible to decrease the amount of bins used and improve the solution. The bin with the lowest utilization is selected and emptied. Then by shuffling the remaining bins it is assessed whether the non allocated items can be allocated. Preferably all the items are relocated to a new bin and the solution requires one bin less. However, it is often observed that this is only possible for large solutions. When increasing the amount of bin,s it becomes more useful to apply bin shuffling. For small instances, this stratgy does create denser packed bins. For a recovery model this is an interesting move, because it can help translate a non feasile solution into a feasible one. For example, if items are not allocated in the initial solution, modifying the loading strategy such that bins are more densely loaded will create space to potentially pack the items.

Another model applying variable neighbourhood search is created by Parreño *et al.* [2010]. The authors solely rely on a variable neighbourhood search approach for the single container loading problem. First, they generate a feasible initial solution by filling up the so called maximal spaces. These are parallelepipeds created by cutting the remaining space in the container. Once an item is loaded, new maximal spaces are created. Once the initial solution is created new neighbourhood solutions are generated by applying five specific movements. These movements are: layer reduction, column insertion, box insertion and emptying the container and filling it again using two different procedures. Once a large neighbourhood is created the best solutions are assessed, this repeats until the maximum amount of iterations is reached. The best solution found up until that point is saved. This is referred to as the variable neighbourhood descent approach. The basic variable neighbourhood search is also applied. This also contains a stochastic move called shaking which makes sure the simulation does not stay in a local minimum.

6.2.6. Ant Colony optimization

In Yuan *et al.* [2010], an ant colony optimization (ACO) algorithm is applied. Ant colony optimization is a swarm intelligence method, where the algorithm tries to find the optimal path in a similar way ants do. When an ant finds a food source, it will free a specific chemical that reacts as a pheromone. The ant will walk back to the nest and other ants will react on the pheromones. A strong smell is created along the path of the food source because all the ants that react will gather food and release pheromones too. After a certain amount of time the shortest path will have the strongest pheromones because it is the most traveled route. For the model a set of solutions is made and the shortest path will have the strongest pheromones. This solution is then explored further. The different solutions are made using a layer building algorithm. That builds different layers using items and places these layers into the bins.

6.2.7. Quasi-Monte-Carlo Tree Search for 3D Bin Packing

A completely different approach is used in Li *et al.* [2018] in which the authors adapt a monte carlo tree search to work for the single bin packing problem. They refer to the model as a quasi monte carlo tree search. Monte carlo tree search methods are best known to simulate and beat certain games like chess and poker. A first solution is created, which is called the root-node. Each node exists out of an evaluation value which is the volume utilization. From the root-node, successive child-nodes are added up until the leaf-node. In the application for the 3D-BPP multiple child-nodes can be created as working-nodes. A leaf-node is reached once all the child-nodes in the path have a value no smaller than their respective Top-K values. The Top-K values are values in a table that represent an upper confidence bound. When better solutions are found in the path these are further expanded. This way MCTS can expand the tree towards the best working moves.

6.2.8. Hybrid Simulated Annealing and Tabu Search

In Mack *et al.* [2004] a hybrid simulated annealing and tabu search algorithm is created. The initial solution is created by combining items to create multiple larger blocks. The blocks are placed in the bins by making an evaluation using the residual space obtained for that arrangement. For the TS, small and large neighbourhoods are distinguished. Both are generated by making exactly one a change on a significant index position in the packing sequence. The difference between the small and the large neighbourhoods is that the large neighbourhoods can modify their index positions within an iteration, while the small neighbourhoods have pre-determined indices per iterative step.

A hybrid model is created using two different methods, in which the goal is to implement the advantages of both heuristics. In the first method an SA model is applied as described in subsection 6.2.2. Then as a post processing step a TS phase is added, this process is iterated multiple times. The other approach is to run the SA phase multiple times and then use the best solution to apply a TS post processing phase. The hybrid models for both methods are measured to be effective.

6.2.9. Hybrid biased random-key genetic algorithm and variable neighbourhood descent

Zudio *et al.* [2018] is an interesting model because the authors create a hybrid model using a biased random-key genetic algorithm and variable neighbourhood descent. This model uses the same maximal space method to create an initial solution as in Parreño *et al.* [2010]. If the item does not fit into the bin, a new bin is opened where the item is placed. Then a genetic algorithm is applied and individuals (chromosomes) are created. The individuals are made either elite or non-elite, which happens random for the first iteration (generation). At random, individuals can be prioritized to elite every move. New generations are created using features of a variable neighbourhood descent approach. The combination of variable neighbourhood descent and genetic algorithm will steer the model into optima, while trying to avoid local minima. The big downside of the model is the fact that it only works with fixed orientations.

7

RECOVERY MODELS IN OTHER FIELDS

As a consequence of the lack of recovery models for an air cargo application, recovery models of other fields are analysed. When analysing the heuristics used in these models, certain methods might be applicable for an air cargo situation.

For a competition called the 2009 ROADEF challenge, the goal was to create a model to solve the aircraft and passenger recovery problem. The competition was won by the model that was created in Bisaillon *et al.* [2011]. By applying a large neighbourhood search heuristic, the authors were able to get well performing solutions with limited computation effort. The heuristic has three phases: construction, repair and improvement. In the first two phases a feasible solution is created using the original flight schedule, then improvement moves will optimize the solution in the last phase. During the construction phase, a randomness is added to explore multiple areas of the solution plain. These phases are iterated until a certain maximum in computational time is reached.

In Sinclair *et al.* [2014], the authors took the model from Bisaillon *et al.* [2011] and improved it. They expanded the three phases to become more complete. Using the instances from the 2009 ROADEF challenge they were able to yield very good solutions.

The last update made to the initial model presented in Bisaillon *et al.* [2011], is performed in Sinclair *et al.* [2016]. The authors solve the instances given in the ROADEF problem using the heuristic from Sinclair *et al.* [2014]. This solution is then entered in a mixed integer program and optimally solved using column generation approach.

In Zhao and Chen [2018] the authors create a solution and then assign weights to the aircraft for each flight in a time-space model. At first, these weights are a constant and after each iteration they are updated. They are updated based on the extreme values of the cost of that iteration and the best extreme of the whole population. For the experiment data, the model guarantees that a cancellation is not needed and the performance in runtime is very good as well.

In Hu *et al.* [2017], the aircraft recovery problem is solved using a large neighbourhood search. From operations control center (OCC) multiple objectives are given, which makes the problem more complex. In order to solve the optimization problem in a limited time, the model is built considering one objective, while the others are added as constraints. Upper limits are added to these constraints which are referred to as ϵ-constraints. By setting values for these limits in the constraints, the other objectives can also be taken into account.

The model applied in Chang [2012] is a crew recovery model. If a disruption occurs, the planning of the crew must be recovered taking into account all the required constraints. In Chang [2012] the model takes the non feasible solution and uses a genetic algorithm to create feasible solution. Although this is not a bin packing problem it is still interesting because it uses a non feasible solution.

8
FREIGHT FORWARDING OPTIMIZATION

In the situation of a disruption it might be more cost effective to find an alternative flight for a completely packed ULD. When the ULD is reallocated the different options and costs must be assessed. This can be seen as a second model within the complete recovery model.

Similar to the reconfiguration of ULDs, no recovery models exists for the reallocation of complete ULDs. However, in Archetti and Peirano [2019] they elaborate the complete freight forwarder transportation. They have called the problem the Air Transportation Freight Forwarder Service Problem (ATFFSP). As seen in Figure 8.1, their scope is from the origin of the consignee to the destination of the consignee. When regarding the allocation of the ULDs it is interesting to analyse the steps between the Airport of Leaving (AoL) and the Airport of Destination (AoD). The authors create a time-space network and solve the problem using a mixed integer linear programming model.

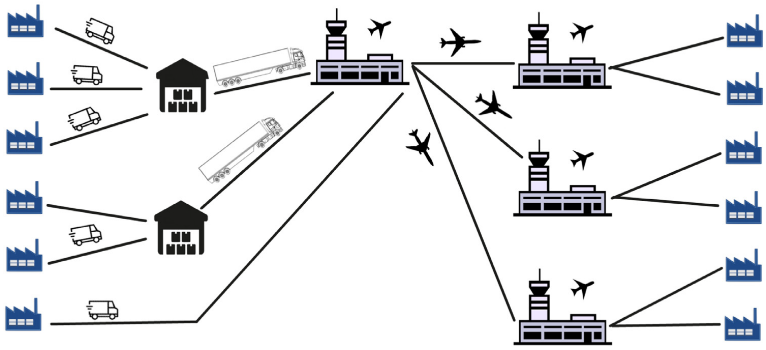


Figure 8.1: *Complete Network ATFFSP [Archetti and Peirano, 2019]*

9

Conclusion and Further Research

9.1. Conclusion

Coming to a conclusion, air cargo is a very complex transportation method, but with a lot of room for improvement. As mentioned in chapter 5, a solution must be generated for the disruptions caused by the uncertainties from the supply and demand side.

By creating a recovery model, these disruptions can be resolved. Important factors are the time it takes to find a solution and the amount of times the model is run to create new solutions. As a result of time being a constraint a sacrifice is made in the quality of the solution and it is not a given that the global solution is found. This is confirmed in the comparison of the bin packing models assessed in chapter 6.

The general loading of items into containers is already a complex problem to solve. As found in chapter 4, applying the bin packing problem to air cargo makes it even more complex and very specific requirements arise. From chapter 6, a mixed integer approach will take too long to solve for a recovery situation. This results in a model that incorporates a heuristics approach. Actual recovery models for air cargo do not exists, but bin packing models with a focus on time do. An analysis of these models in chapter 6 show the extensive amount of options for heuristics moves which can potentially be used for the recovery model.

In order to get a better view on recovery models, other fields were analysed in chapter 7. Models for the aircraft and passenger recovery, aircraft recovery and air crew recovery problems were assessed. These also showed similar heuristic methods that were used in the bin packing models, which shows interest in further elaborating these methods for a specific recovery model.

Lastly, the reallocation of complete ULDs is researched in chapter 8. The model found is not a recovery model, but a complete optimization model of the Air Transportation Freight Forwarder Service Problem. They apply a mixed integer programming model to solve the problem, which is also interesting for the reallocation of the ULDs.

9.2. Further Research

The literature review has given insights on the existing literature for an air cargo recovery model. However, not all subjects were raised in order to directly solve this problem. None of the bin packing models were specifically designed for recovery purposes, which means the exact application must be created.

Furthermore, all the heuristic models are intended to pack items into ULDs, but when a disruption occurs some of the ULDs are fully packed and must be assigned to a different flight, as mentioned in chapter 8. The allocation of the ULDs onto different flights is a very promising recovery move. However, to make this feasible, the cost of different options must be analysed. For example, the handling cost of bringing the ULD back to the warehouse or the trucking cost of flying it from a different airport.

When implementing the recovery model the ULDs that do not fit on the aircraft must be divided into the two parts of the model. The ULD is either taken apart and the individual items are loaded onto the ULDs in the aircraft or the complete ULD is reallocated to a different flight. The first model will be referred to as the ULD reconfiguration model and the second model as the ULD reallocation model. Eventually these two parts are merged into one large model.

In order to create a complete model, and solve the research objective, the following main research questions will be answered:

- How disruptions to the air cargo allocation plan should be addressed, while minimizing the resulting costs incurred to the airline?

The sub questions of the main research question can be divided into three parts. Questions linked to the ULD reconfiguration model, to the ULD reallocation model and to the complete model.

The question that is assembled for the ULD reconfiguration model is:

- What heuristic method can be used to reconfigure the ULDs and pack more items?

The question that is assembled for the ULD reallocation model is:

- What method can be used to reallocate fully packed ULDs that do not fit on the planned flight?

The questions that are assembled for the complete model are:

- How can the cost of the available recovery options be quantified?
- What are the requirements of the model in terms of time efficiency?
- For a specific ULD, subject to disruption, what are the decision criteria guiding the selection of either the ULD reconfiguration model or the ULD reallocation model?

BIBLIOGRAPHY

Uldcare, *Uld types,* (2019), https://www.uldcare.com/uld-tool-solutions/uld-types/#1471477848238-e095d776-30d6 .

H. Zwitzer, Personal Interview (2019).

F. Brandt and S. Nickel, *The air cargo load planning problem - a consolidated problem definition and literature review on related problems,* European Journal of Operational Research **275**, 399 (2018).

R. E. Korf, *A new algorithm for optimal bin packing,* in *Eighteenth National Conference on Artificial Intelligence* (American Association for Artificial Intelligence, Menlo Park, CA, USA, 2002) pp. 731–736.

B. Korte and J. Vygen, eds., *Bin-packing,* in *Combinatorial Optimization: Theory and Algorithms* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 426–441.

H. Kellerer, U. Pferschy, and D. Pisinger, *Introduction,* in *Knapsack Problems* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004) pp. 1–14.

N. Ma and Z. Zhou, *Mixed-integer programming model for two-dimensional non-guillotine bin packing problem with free rotation,* in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (2017) pp. 456–460.

S. Martello, D. Pisinger, and D. Vigo, *The three-dimensional bin packing problem,* Oper. Res. **48**, 256 (2000).

B. Rieck, *Basic analysis of bin-packing heuristics,* (2009).

M. van Aken, *An efficient bin-packing algorithm applied to packing groceries in a fulfillment center,* TU Delft repository (2019).

H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg, *Vehicle routing problems with loading constraints: state-of-the-art and future directions,* OR Spectrum **37**, 297 (2015).

A. Bortfeldt and G. Wäscher, *Constraints in container loading – a state-of-the-art review,* European Journal of Operational Research **229**, 1 (2013).

Virgin Atlantic Cargo, *Generic product sheet,* `https://www.virginatlanticcargo.com/content/dam/cargo/pdf/GenericProductSheets.pdf` (2016), accessed: 2019-11-25.

M. Koch, A. Bombelli, and B. Santos, *A forecast and optimization tool for uld packing in the air cargo industry,* (2019).

Y. Wu, *Modelling of containerized air cargo forwarding problems under uncertainty,* Journal of the Operational Research Society **62**, 1211 (2011).

D. Murph, *Behind the scenes at delta's cargo,* `https://thepointsguy.com/news/behind-the-scenes-delta-cargo/` (2019), accessed: 2019-11-28.

Modern Airliners, *Boeing 777 specs, what makes this giant twin work?* `http://www.modernairliners.com/boeing-777/boeing-777-specs/` (2017), accessed: 2019-11-28.

Bureau of Transportation Statistics, *On-time performance - flight delays at a glance,* `https://www.transtats.bts.gov/HomeDrillChart.asp` (2019), accessed: 2019-11-28.

C. Paquay, M. Schyns, and S. Limbourg, *A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application,* International Transactions in Operational Research **23**, 187 (2016), https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12111 .

C. Paquay, S. Limbourg, and M. Schyns, *A tailored two-phase constructive heuristic for the three-dimensional multiple bin size bin packing problem with transportation constraints,* European Journal of Operational Research **267**, 52 (2018).

J. Kang and S. Park, *Algorithms for the variable sized bin packing problem,* European Journal of Operational Research **147**, 365 (2003), fuzzy Sets in Scheduling and Planning.

X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowsland, *A comparative review of 3d container loading algorithms,* International Transactions in Operational Research **23**, 287 (2016), https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12094 .

J. Hemminki, *Container Loading with Variable Strategies in Each Layer* (University of Turku, 1994).

Y. Wu, W. Li, M. Goh, and R. de Souza, *Three-dimensional bin packing problem with variable bin height,* European Journal of Operational Research **202**, 347 (2010).

D. Mack, A. Bortfeldt, and H. Gehring, *A parallel hybrid local search algorithm for the container loading problem,* International Transactions in Operational Research **11**, 511 (2004), https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-3995.2004.00474.x .

T. G. Crainic, G. Perboli, and R. Tadei, *Ts2pack: A two-level tabu search for the three-dimensional bin packing problem,* European Journal of Operational Research **195**, 744 (2009).

W. Zhu, Z. Zhang, W.-C. Oon, and A. Lim, *Space defragmentation for packing problems,* European Journal of Operational Research **222**, 452 (2012).

F. Parreño, R. Alvarez-Valdes, J. F. Oliveira, and J. M. Tamarit, *Neighborhood structures for the container loading problem: a vns implementation,* Journal of Heuristics **16**, 1 (2010).

J. Yuan, W. Xiong, and B. Jiang, *Hybrid binary ant colony algorithm for container loading problem,* in *Proceedings of the 29th Chinese Control Conference* (2010) pp. 5247–5251.

H. Li, Y. Wang, D. Ma, Y. Fang, and Z. Lei, *Quasi-monte-carlo tree search for 3d bin packing,* in *Pattern Recognition and Computer Vision*, edited by J.-H. Lai, C.-L. Liu, X. Chen, J. Zhou, T. Tan, N. Zheng, and H. Zha (Springer International Publishing, Cham, 2018) pp. 384–396.

A. Zudio, D. H. da Silva Costa, B. P. Masquio, I. M. Coelho, and P. E. D. Pinto, *Brkga/vnd hybrid algorithm for the classic three-dimensional bin packing problem,* Electronic Notes in Discrete Mathematics **66**, 175 (2018), 5th International Conference on Variable Neighborhood Search.

S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin, *A large neighbourhood search heuristic for the aircraft and passenger recovery problem,* 4OR **9**, 139 (2011).

K. Sinclair, J.-F. Cordeau, and G. Laporte, *Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem,* European Journal of Operational Research **233**, 234 (2014).

K. Sinclair, J.-F. Cordeau, and G. Laporte, *A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem,* Computers & Operations Research **65**, 42 (2016).

T. Zhao and X. Chen, *A weight-table based heuristic algorithm for aircraft recovery problem,* in *2018 37th Chinese Control Conference (CCC)* (2018) pp. 2242–2246.

Y. Hu, H. Liao, S. Zhang, and Y. Song, *Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft,* Expert Systems with Applications **83**, 283 (2017).

S.-C. Chang, *A duty based approach in solving the aircrew recovery problem,* Journal of Air Transport Management **19**, 16 (2012).

C. Archetti and L. Peirano, *Air intermodal freight transportation: The freight forwarder service problem,* Omega , 102040 (2019).