Development of a support tool to limit workload at a hub for Air Cargo combination Airlines via dynamic loading advices at outstations

B.W. (Wouter) van der Wal

Development of a support tool to limit workload at a hub for Air Cargo combination Airlines via dynamic loading advices at outstations

by

B.W. (Wouter) van der Wal

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on the 13th of December 2021 at 10:00h.

Student number:	4290712	
Project duration:	January 24, 2021 – December 13, 2021	
Thesis committee:	Dr. Ir. B.F. Lopes dos Santos	TU Delft, Chair & Supervisor
	Dr. A. Bombelli	TU Delft, Supervisor
	Dr. J. Sun	TU Delft, Examiner

This thesis is confidential and cannot be made public until December 13, 2023.
Cover: KLM Royal Dutch Airlines

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

Working on your thesis in 2021 means that you also had to deal with the side effects of the corona pandemic, which included working at home, having online meetings with your supervisors and a curfew to prevent you from seeing friends in the evening. However, I would say that I did not experience many difficulties in simultaneously dealing with both thesis work and the lockdowns. This can mainly be attributed to the enormous support from my supervisors, close friends and family. This thesis forms the closing chapter of my Master in Aerospace Engineering and my life as a student at TU Delft. It has been a long and challenging journey but I am very thankful to the ATO department, the staff there managed to trigger my interests which resulted in me being where I am now.

In particular I would like to express my thanks to my supervisor Alessandro Bombelli, who was always available on the short term if I had questions or needed feedback. Even though it had to be done online, it felt like really comfortable meetings. Additionally, I also would like to thank my other supervisor Bruno Lopes dos Santos, who was there right there from the start to always provide constructive feedback and challenging questions. I really appreciate your directness in providing feedback, which in my opinion only works positively.

Next to this I am very grateful for the opportunity to do my thesis at a company. The warm welcome there and the large number of colleagues contributing to the project and my well-being really pulled me through this project. In special I thank Rutger-Jan Pegels who offered me this position as a graduate intern and for the many concept discussions we were able to have despite of his busy agenda. Also thanks to Mark Starrenburg for guiding me through the initial phase of the research and thanks to Bob Bokern and Mouhsine Bouhbouh, who took over from Mark and who were always available to discuss the project and to have a laugh when we were allowed back in the office.

Last but not least, I would like to thank my parents for the constant support, emotionally and financially, during the whole duration of my studies. Without you I would not even been able to study at TU Delft and let alone be able to finish it.

Wouter van der Wal
Rotterdam, November 28, 2021

Contents

List of Figures

List of Abbreviations

BD	Breakdown - process of disassembling ULDs at the hub
BPP	Bin Packing Problem
BU	Buildup - process of assembling ULDs at the hub
DAP	Delivered as Promised - % of cargo delivered on time
LF	Load Factor
M-ULD	Mixed ULD
RCS	Received Cargo from Shipper
T-ULD	Through ULD
ULD	Unit Load Device

Introduction

Air transport is one of the major means of transport for intercontinental cargo transport. In comparison to other means of transport, like sea or rail transport, air transport has one major advantage. This advantage is the average transit time, as airplanes travel much faster than trains and ships. The downside of air transport however, is that it is more expensive and there are significant constraints in terms of weight and volume. The above mentioned properties make air cargo transport especially useful for four types of cargo, according to [12]. These four types are dangerous, urgent, perishable and valuable cargo. The types have in common that transit time is more important than the high costs associated to air transport. This is illustrated by the ratio between value and weight transported by air. Whereas air cargo only accounts for 1% of the intercontinental weight transport according to KiM [24], it accounts for 35% of the value according to IATA [35]. Therefore, airlines that transport cargo have the goal to build a network that is able to transport cargo across the globe as fast as possible. In addition to that, airlines want to deliver as promised, as this is a major motivator for shippers to choose a specific airline.

A lot of airlines operate a hub and spoke network, this is especially the case for cargo airlines that also transport passengers. These airlines are called combination airlines because they combine the passenger business with a cargo business. Cargo is then transported by both cargo-only aircraft (full freighters) as passenger aircraft. In the case of passenger aircraft the lower deck, beneath the passenger section, is used to transport cargo. In the hub of a cargo airline, both cargo from intercontinental flights as cargo coming from trucking stations all over the continent are connected onto their next flight or truck, which makes it one of the most critical handling blocks in the network. If the hub is not able to handle all incoming cargo, shipments will miss it's connection and will therefore not be delivered as promised. This results in a loss of customer satisfaction as well in unused capacity of flights outbound from the hub. It is therefore beneficial if airlines would have the possibility to adjust the workload to the workforce and vice versa. This research focuses on how the workload can be minimized by optimizing pallets at the outstations; the spokes of the network.

This research provides an unique insight and approach for both an academic as an airlines' point of view. In the academic world a lot of research has been done on the optimization of Unit Load Devices (ULDs), which is the name of the air transport equivalent to sea containers. This is done from a revenue management perspective, which has the goal to sell just as much space as the capacity. In order to determine the available space, models are used to determine the space utilization, ranging from one to three dimensional models. These models also often include detailed constraints like stackability, item orientation and physical stability. However, these models assume that all cargo for a specific flight is available at the same place at the same time to build the ULDs optimally. This is unfortunately not the case in the operation, where cargo is delivered at a large number of outstations at different times. The result of optimizing ULDs at outstations could be a significant reduction of workload in the hub. At outstations, pre-built ULDs could be made for specific flights for example, or ULDs containing cargo with similar connection times could reduce the peak workload. Given this current situation, an optimization from the outstation's perspective could result in significant reductions of the (peak) workload at the hub.

Similarly, airlines also researched and developed these kind of models to optimize the assignment of items to ULDs from a revenue management perspective with the same limitations. However these palletization solutions from revenue management are not useful in the operation due to the nature of the operation where cargo doesn't always show up as booked and where shipment dimensions are not required. Instead, all shipments are planned onto specific loading groups by hand by experienced planners, who apply a set of rules of thumb that aims to reduce the workload. This planning is then issued to outstations as a Loading Advice. A downside of this situation, where planners apply a set of rules is that it does not consider shipment and network characteristics and the rules remain the same in cases that a shortage of workforce in the hub is forecasted. In the case of a shortage of workforce in the hub it could be chosen for example to sacrifice capacity on flights with a low load factor and a low revenue to favour flights with a high load factor and a high revenue. Besides this, smart and efficient clustering of items can also improve the spread of the workload by reducing the high peaks, which are typical for airlines due to the typical arrival and departure banks of airlines at their hub.

This thesis report is split into parts. In Part I, the scientific paper is presented. Part II contains the relevant Literature Study that supports the research.

I

Scientific Paper

Development of a support tool to limit workload at a hub for Air Cargo combination Airlines via dynamic loading advices at outstations

Wouter van der Wal,*

Delft University of Technology, Delft, The Netherlands

Abstract

The planning of air cargo across an airlines' network is proven to be a complex problem consisting of multiple subproblems. Currently, all these subproblems are solved sequentially and by hand, resulting in partial solutions instead of an integrated solution. Two of these subproblems are the palletization of items onto ULDs and the scheduling of a workforce at a cargo terminal. This research proposes an implementable tool that tries to find a more integrated solution between these two subproblems. The tool provides loading instructions to ground handlers on how to combine cargo at the outstation. It makes use of both a dynamically determined volume threshold for ULDs containing cargo for one connecting dated flight (T-ULD) as a K-means clustering algorithm to combine cargo based on the connection time in the hub. A 1D Bin Packing Problem and Breakdown Scheduling model was created to study the effects of the new loading instructions. The dynamic threshold showed a predictable behaviour in the reduction of workload while achieving a baseline workload reduction of 1.4%. The tool further opens the opportunity to reduce workload even further with 24.9%, but this is associated with a risk and additional costs. The K-means clustering algorithm did not show an improvement to the baseline but it did offer the opportunity to cluster cargo based on multiple properties besides connection time at the hub.

1 Introduction

In intercontinental cargo transport there are mainly two types of transport, which are transportation by ship and by aircraft. Even though transport by air is much more expensive, it accounts for about 35% of the value of all intercontinental cargo transport [IATA, 2018]. This can be devoted to the unique properties of air cargo, which gives it an advantage over other means of transport. In comparison to other modalities, air cargo is a fast and safe way of transport. This makes it particularly interesting for urgent, perishable, valuable and dangerous cargo [Brandt and Nickel, 2019]. Therefore, it is the primary goal of airlines to build a network that is able to transport the cargo as fast and safely as possible across the globe, as the percentage of cargo Delivered As Promised (DAP) is a major motivator for a shipper to choose a specific airline. The DAP is by industry standard defined as the percentage of cargo delivered on time by an airline [Cargo-IQ, 2019].

For an airline, the effect of the DAP on its revenue can be split into two parts. The first part is that a lower DAP score results in a lower achieved Load Factor (LF), as a part of the capacity is not used because cargo that was initially booked on that flight has been delayed onto a next flight. This means that with a lower DAP score, the Revenue Management department can sell less cargo per flight, resulting in less revenue generated per flight. The second part that can affect the airlines' revenue is customer satisfaction. It can be argued that a shipper is willing to pay more for an airline with a high DAP score than for an airline with a low DAP score. Shippers choose air transport over other modalities because of its low transit time and therefore a higher chance of cargo being delivered on time might be worth the higher price tag. This makes that an airline with a high DAP score might be able to set a higher fare in comparison to the competition, which in turn results in more revenue generated per flight.

Three types of cargo airlines can be distinguished. The first type is a full freighter airline which only transports cargo from airport to airport. The second is an integrator airline, which provides transport from shipper to consignee via air, this includes the complete shipment journey. The last one is the combination airline, which transports both cargo as passengers. Combination airlines carry cargo in full freighter aircraft or in the cargo hold of passenger aircraft, which is positioned below the passenger deck. The combination airline is the focus of our research. Most combination airlines use a hub and spoke network where all cargo is collected and connected

*MSc. Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology to a next flight. This makes the hub a critical piece of the network as all cargo has to pass through the hub. In case of a shortage of handling capacity, cargo will miss its connection. This in turn will result in a lower DAP score. In literature, a lot of research has been done on the topic of load planning for air cargo. This is mostly solved from the perspective of the Revenue Manager who has the task of determining whether there is still space to accept a certain booking. At one point, the responsibility of the cargo flight is transferred to the operation who have the goal to transport the sold cargo to its destination, preferably on time. Here, a range of operational constraints come into play, such as the large variety of delivery times and delivery stations. In this complex playing field the operation continuously tries to match the workforce to the workload.

Air cargo is assembled on special pallets or containers, which are called Unit Load Devices (ULDs). In Figure 1 the handling process of ULDs in the hub is illustrated. For simplicity reasons the cargo that is directly delivered at the hub is neglected and only cargo coming from outstations going to outstations is shown. The cargo inbound to the hub is transported in both aircraft as trucks. Trucks are mostly used for continental or short distance legs because narrow body aircraft, which are deployed for short to medium flight legs, have a small cargo hold. Furthermore, it is easy to increase the cargo capacity of trucking legs by scheduling extra trucks, which makes trucking more flexible. Intercontinental legs are always flown by aircraft, which are either wide-body passenger aircraft or full freighter aircraft.

From the hub's perspective, both flights as trucks deliver two types of ULDs. A Through-ULD (T-ULD) contains cargo for only one connecting dated flight. The Mixed-ULD (M-ULD) contains cargo for multiple connecting flights. As is illustrated by Figure 1, only the M-ULD has to go through the handling process. In the breakdown process the M-ULD is disassembled and all shipments are transported to an area where the connecting flight is prepared. The next process is the buildup process, where shipments are consolidated onto ULDs for departure to a next destination. The T-ULD does not require any handling and can be directly moved to the connecting flight upon arrival. Potential delays in the hub therefore often occur for shipments that are on a M-ULD. A first measure to take for an airline in case there is not enough handling capacity, is to decrease the number of M-ULDs, which can be achieved by consolidating more shipments onto a T-ULD. A second measure for airlines is to improve the distribution of handling deadlines. If all M-ULDs have similar deadlines there is a high peak in workload, which might result in not handling all cargo on time. If an airline can achieve a good spread of handling deadlines the peak workload can be lowered to possibly avoid missed connections.

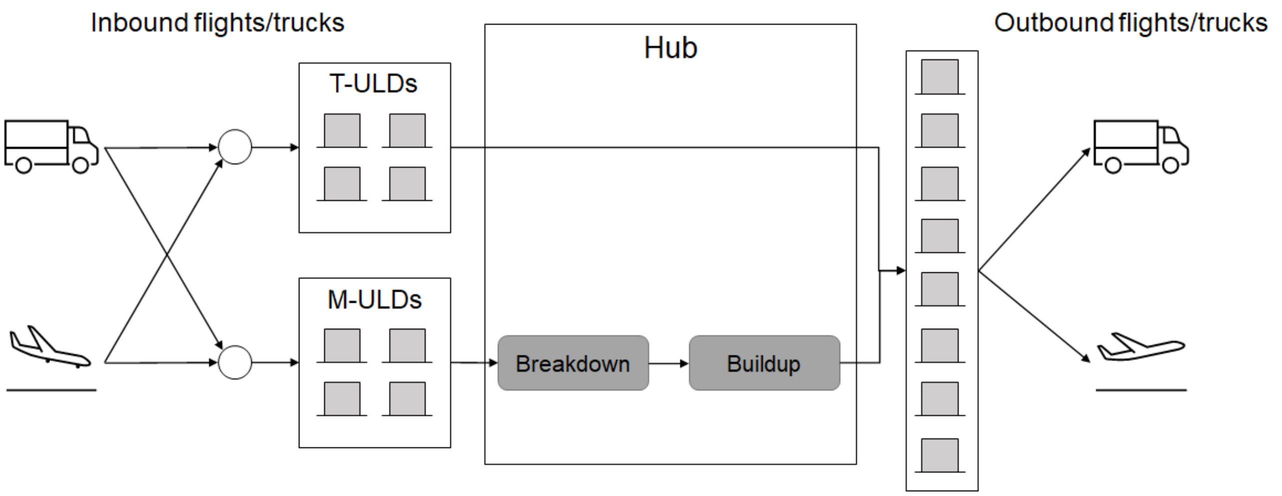

Figure 1: Illustration of the ULD handling process for inbound and outbound flights in the hub

Multiple cargo airlines have rules of thumb in place to address the above mentioned measures. This includes a volume threshold to determine whether a specific connection has enough cargo to build a T-ULD. Furthermore, M-ULDs are usually grouped by connection times by using fixed thresholds, which is defined as the time between the arrival and departure of a shipment in the hub. These rules of thumb are still applied by hand, usually by planners but sometimes also by the ground handlers themselves. A downside of these rules of thumb is that they are generic and do not consider specific flight and hub characteristics. An example is that the T-ULDs on a flight with a low LF do not have to be fully filled, as there is plenty of space left. The opposite is the case for flights with a high LF, which require fully filled T-ULDs as the flight is almost limited on capacity. Another consideration for a cargo airline could be the value of the flight because for an airline highly profitable flights are important to prioritize as they contribute a lot of value. A last consideration is the balance between workload and workforce in the hub. If there is a shortage of handling capacity in the form of workforce, it is desirable to reduce the workload by generating more T-ULDs, which results in less M-ULDs to be handled. So, there is potential to replace these rules of thumb with more data-driven decisions that are more tailored to the status of the network, such as hub capacity or flight capacities.

The objective of this research is to develop and evaluate a decision support tool for planners to improve the balance between workload and workforce at a hub of a combination airline. This is done by generating loading advices for outstations, resulting in different solutions for assigning items to ULDs. These advices will be generated based on the current situation of the network. Two approaches have been researched, the first is a mechanism to create more T-ULDs, especially for low load factor and low value flights. This will reduce the overall workload. Another approach is a more appropriate way of determining the M-ULD groups, such that the deadlines are better spread, resulting in lower peaks in the workload.

This paper is structured as follows, in the next section (section 2) the related work on air cargo in literature will be reviewed. This will be followed by the methodology in this research section 3. Then in section 4 the case studies for the model will be explained and discussed, before presenting the results in section 5. The results are discussed in section 6 and the Conclusions are provided in section 7 followed by some recommendations for future research in section 8.

2 Literature Review

The complete process from a set of shipments to consolidated ULDs loaded in an aircraft is referred to as the Air Cargo Load Planning Problem (ACLPP) in literature [Brandt and Nickel, 2019]. This problem consists of four unique subproblems, which are;

- The Aircraft Configuration Problem (ACP) about how many of which types of ULDs to load per aircraft.
- The Build-up Scheduling Problem (BSP), which addresses when to schedule which ULD for build up at a cargo terminal.
- The Air Cargo Palletization Problem (APP), that solves which items to load onto which ULD.
- The Weight and Balance Problem (WBP) that tries to find an optimal assignment of ULDs in an aircraft such that an optimal aircraft trim is found.

The set of subproblems are all interrelated which means that the outcome and decisions in one of the subproblems could influence the other subproblems. The subproblems are already difficult to solve on their own as at least the BSP, APP and WBP are known to be NP-hard [Brandt and Nickel, 2019, Lurkin and Schyns, 2015]. In literature therefore there are no models known yet that solve multiple parts of the ACLPP, let alone the whole problem from origin to destination. The current sequential solving of the subproblems leads to partial solutions whereas a more integrated solution could be more beneficial for the whole process. Additionally, most work is still done by hand by highly experienced planners or ground handlers. Brandt and Nickel (2019) devote this current situation to the lack of decision support systems [Brandt and Nickel, 2019]. The lack of decision support systems in itself can be attributed to the fact that the data required for the models is only gradually becoming available in the last few years and that the data quality is not reliable enough yet. Shippers are only required to provide complete shipment weights and volumes, while individual item weights, volumes and dimensions would be required to find realistic, feasible and good solutions. This research does focus on the interconnection between two subproblems where the question is whether the decisions for the APP at an outstation could be changed in order to positively influence the BSP in the hub. The state of the art of both subproblems will be discussed in section 2.1 and section 2.2. The focus of this research will be on creating a first workable version that could aid the highly experienced planners. Therefore, the current loading advices issued by planners will be used to make different decisions at outstations. A good option to make M-ULD groups are clustering techniques, which involve the distribution of data points over clusters. Section 2.3 will therefore focus on clustering techniques.

2.1 Air Cargo Palletization Problem (APP)

The Air Cargo Palletization Problem involves the assignment of items to ULDs with the goal to either maximize the used space in a given number of ULDs or to minimize the required number of ULDs to pack all items [Bortfeldt and Wäscher, 2013]. The maximization approach is usually referred to a Knapsack Problem (KP) and the minimization approach as a Bin Packing Problem (BPP). Both the KP as the BPP can be solved for one, two or three dimensions, where both space (in volume or 3D orientation) and weight are the main constraints. Specifically for the Air Cargo Palletization problem additional constraints were introduced like item orientation, stacking constraints and weight distribution [Brandt and Nickel, 2019]. In recent literature for the APP the three dimensional Bin Packing Problem was addressed. Paquay et al. (2016) formulated a Mixed Integer Programming model [Paquay et al., 2016] for the APP. Because of the high computational times with only small instances, further research focused on efficient heuristics to approach optimal solutions within a reasonable amount of time [Paquay et al., 2018a, Paquay et al., 2018b]. However, according to Brandt and Nickel (2019) both heuristic approaches are less optimal than what airlines are already able to solve nowadays [Brandt and Nickel, 2019]. Furthermore, they also stated that the data quality in practice is not good enough yet to solve a 3D problem, due to missing dimensions or inaccurate dimensions.

2.2 Build-up Scheduling Problem (BSP)

In other industries there are problems that are very similar to the Build-up Scheduling Problem. Van den Bergh et al. (2013) published an extensive review of present literature on the topic of personnel scheduling problems [Van den Bergh et al., 2013]. Specific for the air cargo case there are several papers that presented models to address these problems [Yan et al., 2006b, Yan et al., 2006a, Yan et al., 2008a, Yan et al., 2008b, Emde et al., 2020]. The problem can be solved with different goals, the first goal is to estimate the workforce to be scheduled, both in the form of personnel as machinery. The second goal is to create a roster of when to schedule which task by which team. Brandt and Nickel (2019) stated that recent models are suitable for the first goal of estimating the size of the workforce, but that due to practical reasons the models are not yet suitable for the second goal to accurately plan ULD build ups [Brandt and Nickel, 2019]. The focus on this research however is to study the effects of different palletization solutions on the Breakdown & Build-up workload in the hub. This indicates present models should suffice in estimating the required workforce. Additionally, breakdown is a bit more straightforward than build up. The breakdown department receives a complete ULD and has the task to separate all shipments on a connecting flight level and do not have to account for item availability. For build up this is different, as their output is a complete ULD and therefore the teams can only start working if all shipments for that ULD are delivered.

2.3 Clustering Algorithms

The assignment of bookings to specific M-ULD groups is a process that could be performed by a clustering algorithm. The purpose of clustering is to find a tool or algorithm that can predict the class of a set of items [Saxena et al., 2017]. Clustering algorithms can be categorised as unsupervised learning methods because the classes are unknown at the start and the algorithm predicts the classes by itself. Saxena et al. (2017) made an extensive review of all clustering algorithms [Saxena et al., 2017]. There are two main branches, which are hierarchical clustering methods and partitional clustering methods. Partitional methods are most suited to this research. It provides a set of clusters k and the respective assignment of data points to a cluster. Here two options are particularly applicable, which are distance-based partitional clustering methods [MacQueen et al., 1967, Kaufman and Rousseeuw, 2009, Ng and Han, 2002] and density-based clustering methods [Ester et al., 1996]. The first and best-known method is the distance-based K-means by MacQueen et al. (1967) [MacQueen et al., 1967]. The method works by determining centroids for every cluster. The centroid of a cluster corresponds to the average position of all data points belonging to that specific cluster. The process of determining clusters and centroids is repeated until new centroids no longer move, indicating an optimal solution is reached.

The advantage of a density based method is that it can recognise shapes in data, recognise outliers and it determines the optimal number of clusters k by itself. However, for distance-based methods there are options to determine the optimal number of clusters by using additional methods. Examples of these methods are the elbow method [Bholowalia and Kumar, 2014] or a silhouette value analysis [Rousseeuw, 1987]. Both methods are built on running multiple iterations of the clustering algorithm and value each iteration on the best fit. The best fit can then be selected as the optimal number of clusters k.

3 Methodology

This research focuses on combining considerations from two subproblems to find a more integrated solution between the palletization problem at outstations and the work scheduling problem at the hub. A modelling framework was created that combined three models. The three models run sequentially and their input is provided with the previous' model output. The modelling framework with the three models and the interaction between each model is shown in Figure 2. The first model is the Loading Advice model, which is the support model for planners to support them in issuing loading advices. The second model is an adaptation of a conventional 1D Bin Packing Problem. This model uses the Loading Advice from the previous step to simulate the effects on the palletization of items onto ULDs. The Breakdown scheduling model is the third and last model, which used the output of the 1D Bin Packing model as an input. The Breakdown Scheduling model is an adaptation of models currently addressing the Build Up Scheduling Problem. This model was created to determine the required handling capacity on a shift level to breakdown all ULDs on time. The current loading advice structure was maintained instead of providing exact item to ULD assignments because of the low quality of the available data. It would also not be a large change in procedures for cargo airlines, which makes it easy to implement in the operation. So to summarize, the loading advice model is the support model for planners and the two other models are simulation models to study the effects of the loading advices.

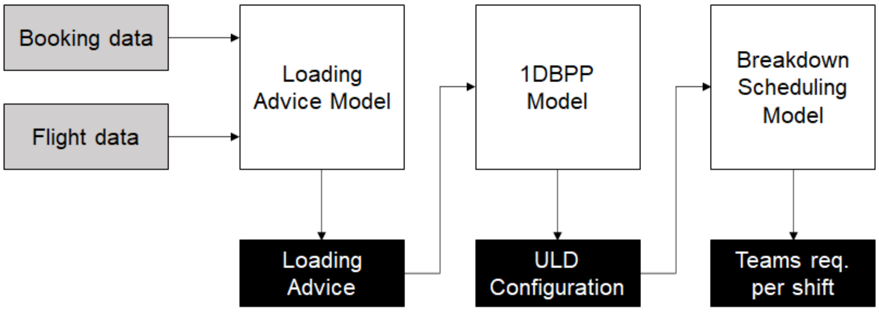

Figure 2: Visualization of the Modelling Framework showing the interrelation between the three models.

3.1 Loading Advice model

An illustration of the Loading Advice model is shown in Figure 3. The loading advice model produces a loading advice for every outstation and for every flight or truck. As mentioned in the introduction, two measures were considered that aimed to reduce the overall workload (upper side of the figure) and the peak workload respectively (lower side of the figure). The first measure is to determine a volume threshold for every outbound flight from the hub. This threshold corresponds to the minimum volume of cargo for a T-ULD and will be referred to as the T-threshold from now. After the flight specific T-threshold is determined, it is checked whether there are enough shipments in the booking list for the selected outstation and flight/truck to satisfy the T-threshold. If this is the case, all bookings for that specific connecting flight are added to the list of potential T-ULDs. The second measure, which is for reducing peak workload, determines the best M-ULD cluster for all bookings using a K-means clustering algorithm [MacQueen et al., 1967]. The input for the clustering algorithm is the connection time in the hub and a second factor can be selected that corresponds to the priority of a booking. Both the dynamic T-threshold as the clustering algorithm will be elaborated on further in this section.

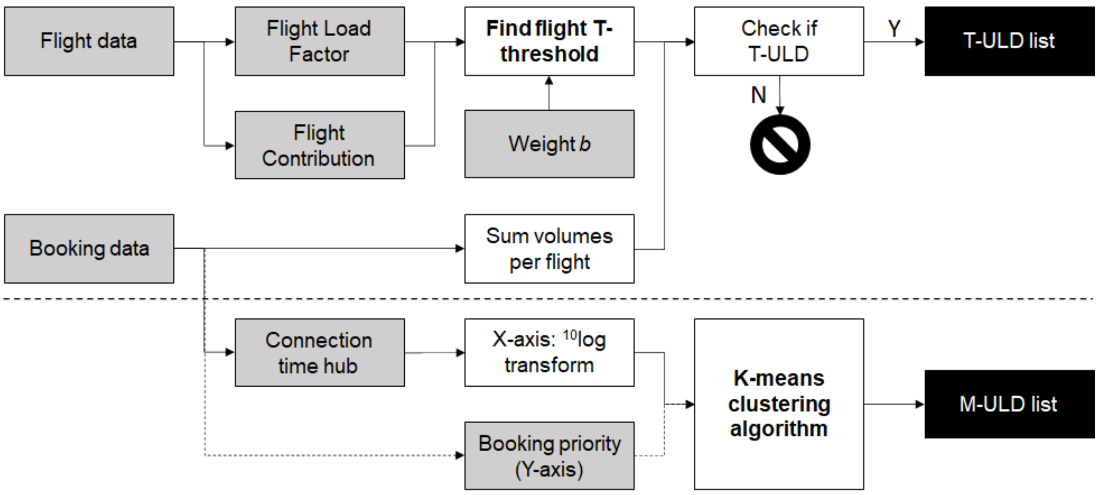

Figure 3: Visual representation of the Loading Advice model

Two sets are used in the model to determine the loading advice (Table 1). The first is a set of flights F, in this set every flight has a Flight number f and a departure date d. This is because in practice airlines often use the same flight number for the same leg, resulting in a non-unique flight number. Therefore the date is added to make every flight unique. The other set is a set of bookings J.

The used parameters are given in Table 2. Important to note here is that LF_{fd} is a volume load factor. This is because the T-threshold is also set as a volume threshold. In the table, two values of Contribution Cb are mentioned as well. Contribution is defined as the revenue minus the sum of variable costs. On a flight level Cb_{fd} this is relatively straightforward, but for individual shipments Cb_j the variable costs are set as a fraction of the flight's variable costs. This fraction is determined by the shipment's weight and volume with respect to

Table 1: Sets for the Loading Advice model

Set	Description
F	Set of outbound flights from the hub, which is a tuple $\{f, d\}$ of a flight number f and a departure date d
J	Set of all bookings on that specific segment

all other shipments on that flight. Even though the definition of the Contribution might not necessarily unique, it is a good indication of the value of a flight or shipment. The three weights b, w_{LF} and w_{Cb} are inputs for the dynamic T-threshold and will be explained later.

Table 2: Parameters for the Loading Advice model

Parameter	Description
b	The weight for correcting the T-threshold, equals a maximum change in m^3
LF_{fd}	Load factor of flight f on departure date d in %
Cb_{fd}	Contribution of flight f on departure date d in €
Cb_j	Contribution of booking j in €
V_j	Volume of one piece of booking j in kg
T_j	Connection time of booking j in the hub
O_j	Outbound flight f on date d of booking j from the hub
w_{LF}	The weight of the flight Load Factor for making a correction in the T-threshold
w_{Cb}	The weight of the flight contribution for making a correction in the T-threshold

3.1.1 Dynamic T-threshold

The flight specific T-threshold is given by Equation 1. This threshold is determined for every outbound flight from the hub. Here, it was chosen to set a base threshold given by the load factor of the flight. This makes that the ULD should have at least the same LF as the flight. The second term is the correction mechanism, which is used to artificially reduce the workload in case of a handling capacity problem. In the equation b corresponds to the maximum reduction of the T-threshold in m^3. A reduction of the T-threshold means more T-ULDs and less M-ULDs, resulting in less work in the hub. So the larger the capacity problem in the hub, the larger the value of b needs to be to reduce the workload. An additional term $\Delta(\{f, d\})$ was introduced to prioritize certain flights over other flights. This term takes a value between 0 and 1. The higher the value, the more the flight is compromised to reduce the number of M-ULDs. For example let's assume we have two flights with a base T-threshold of $8m^3$, one with a value of $\Delta(\{f, d\}) = 0.6$ and the other with a value of $\Delta(\{f, d\}) = 0.2$. If the calibration parameter is set to $b = 4$, one flight will have $T_{thresh} = 5.6m^3$ and the other has $T_{thresh} = 7.2m^3$. Here, the second flight is clearly prioritized higher than the first flight and the magnitude is dependent on the value of b. So, the higher the value of b, the lower the T-threshold. This in turn results in an increase of the number of T-ULDs as the minimum volume required to build a T-ULD on a flight is lowered. This makes that there are less shipments to build M-ULDs with, which results in less M-ULDs and therefore less work in the hub. A side effect of a lower T-threshold is that the overall number of ULDs will increase because of the lower LF on T-ULDs, which for continental outstations results in more required trucks to transport all ULDs.

$$T_{thresh}(\{f, d\}) = VCap_{ULD} * LF_{fd} - b * \Delta(\{f, d\}) \tag{1}$$

The prioritization term $\Delta(\{f,d\})$, which determines the priority of a flight, is provided by Equation 2. The use of the parameter b combined with the term $\Delta(\{f,d\})$ is associated with a risk, because it results in lower LFs on individual T-ULDs than the overall LF of the flight. Because of the presence of this risk, the flights to compromise have to be chosen carefully and as an airline it is not desired to compromise high valued cargo to high LF destinations. Therefore, two main parameters are taken into account. The first is the average load factor of the next three days. If the average LF is high, delayed cargo because of setting a too high value for b could potentially not be recovered. In essence, the potential of recovering missed connections is covered by considering the average LF. The second part is the contribution of the flight. The flight's contribution is compared to the flight with the highest contribution on that day resulting in a value between 0 and 1. This ensures that the high contribution flights are protected as these connections deliver much value to the cargo airline. Together, both terms are weighed against each other using the weights w_{LF} and w_{Cb}. The function is normalized by dividing it by the same weights and then subtracted from 1. This makes that high LF and high Contribution flights will result in a value close to 0, meaning the T-threshold for these flights will not decrease as much as for low LF and low Contribution flights.

$$\Delta(\{f,d\}) = 1 - \frac{w_{LF} * (\frac{1}{3}\sum_{d=d}^{d+3} LF_{fd}) + w_{Cb} * (Cb_{fd}/max(Cb))}{w_{LF} + w_{Cb}} \tag{2}$$

3.1.2 M-ULD clustering

The clustering of bookings into loading groups for M-ULDs is performed using a K-means clustering algorithm. Distance-based clustering algorithms are more suited to this application than density-based clustering algorithms. The unique advantages of density-based methods like recognising patterns in data and recognising outliers are not really applicable to this case and it turned out difficult to find the right input calibration parameters for density-based clustering algorithms. The used clustering algorithm uses two inputs. The first input is the connection time T_j of the shipment in the hub. However, from a practical point of view, close differences in connection times on one ULD are much more important for small connection times than long connection times. This is because the time windows for small connections are very narrow and it is therefore desirable to only handle cargo that has to be broken down at that moment and not a few hours later. Therefore the connection time is put through log_{10} function to artificially enlarge the difference in small connection times with respect to long connection times. A second input could be added such as the LF of the outgoing flight or the value of the shipment. The option exists to cluster based on value and priority, such that the hub could prioritise certain ULDs in case of a shortage of capacity. For this research however it is chosen to only use the first input of connection times to be able to compare it with the current situation. The K-means algorithm is following the schematic in Figure 5.

An additional complexity with the case of this research is that the clusters are required to contain a certain volume of bookings. As can be seen in Figure 4 there is a large variety of booking volumes. As an output, each cluster is preferred to have at least the size of one ULD in terms of volume. In this way, clusters do not have to be mixed which could have a negative impact on peak workload. In literature there are methods that use extra constraints to force the algorithm to find certain cluster sizes. However, to our knowledge, this is not possible to implement using standard techniques and therefore a different approach is taken. As mentioned in section 2, K-means uses a centroid to determine the center of a cluster. Therefore as shown in Figure 5, extra datapoints are added for large volume bookings, resulting that the centroids are pulled towards large volume bookings, which could ensure that the resulting M-ULD clusters are large enough to fill a complete ULD.

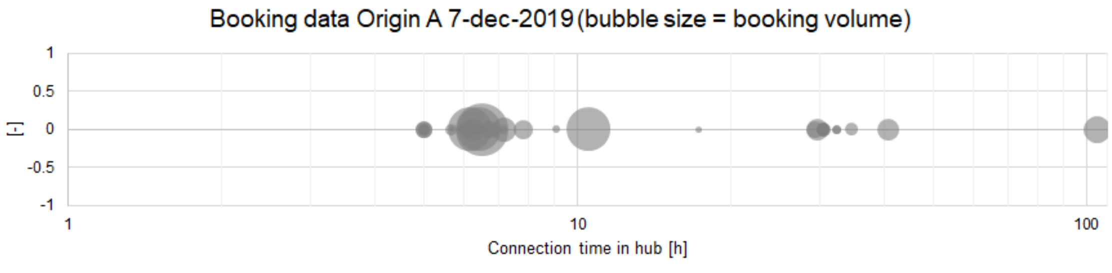

Figure 4: Input of the clustering algorithm

The number of clusters k are determined by performing a silhouette value analysis [Rousseeuw, 1987] as also depicted in Figure 5. The analysis is performed for multiple settings of the number of clusters k, after which the cluster setting with the best silhouette value is selected. The minimum number of clusters is set to 2 and the maximum number is dependent on the total volume of cargo. Here, the maximum number is defined as the volume of cargo for M-ULDs divided by the capacity of a ULD. The value is rounded up to the closest integer value with a maximum of 5 clusters. The maximum is set to 5 to make the loading advice manageable for outstations.

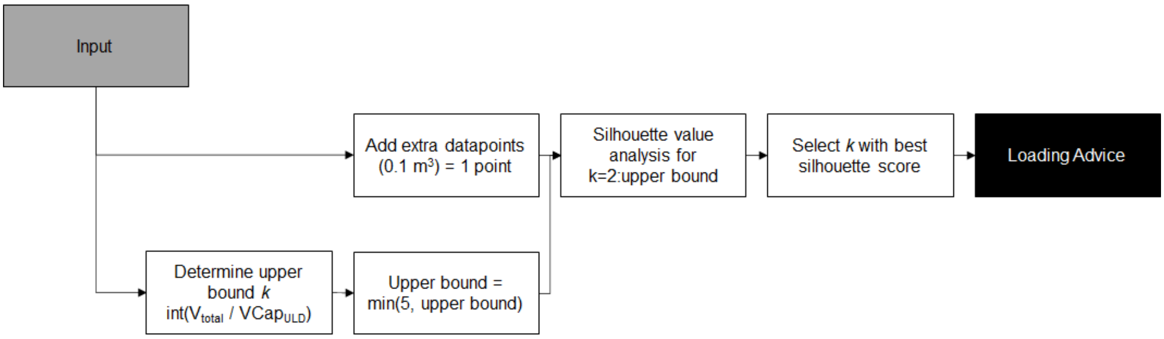

Figure 5: Visual representation of the Clustering implementation

3.2 1D Bin Packing model

For the Bin Packing model a two-phased system is used. The first phase is a heuristic that aims to pre-build all T-ULDs. The second phase is a MILP model that optimizes the required number of ULDs for the rest of the (unpacked) cargo. Three main sets can be distinguished for the Bin Packing model, which are given in Table 3. The number of ULDs for set I are estimated by dividing the total volume of cargo by the volume capacity of one ULD. This is then rounded up and a 50% safety margin is added to ensure that there are enough ULDs to pack all shipments. It is also important to note that every T-ULD group t in the set T contains a subset of bookings j from J. This is the result of the preceding Loading Advice model.

Table 3: Sets for the Bin Packing model

Set	Description
I	Set of all ULDs
J	Set of all bookings on that specific segment
T	Set of all advised T-ULD bookings from the loading advice, each element is a subset of j bookings from J

The parameters used from the sets are given in Table 4. This includes properties of a booking such as the number of pieces N_j, weight of one piece W_j and volume of one piece V_j. Furthermore there are weight and volume capacities, respectively $WCap_i$ and $VCap_i$, for ULDs. Other parameters require some additional explanation. In the model every ULD i gets a pre-assignment of a mix-cluster, which is given by SLR_i. So beforehand every ULD gets a number SLR_i which corresponds to a M-ULD cluster and every booking has a number SLR_j which corresponds to a M-ULD cluster. For example, if ULD 2 gets $SLR_2 = 1$, bookings with $SLR_j = 1$ will be assigned to this ULD as much as possible. This approach is taken in combination with a soft constraint because a balance has to be found between minimizing unused space and following the loading advice as much as possible. In order to move this balance, the "compliance" penalty p_c can be altered. The same can be done for the "spreading" penalty p_s, which is a penalty for spreading a booking over multiple ULDs.

Table 4: Parameters for the Bin Packing model

Parameter	Description
$VCap_i$	Volume capacity of ULD i in m^3
$WCap_i$	Weight capacity of ULD i in kg
V_j	Volume of one piece of booking j in m^3
W_j	Weight of one piece of booking j in kg
N_j	Number of pieces in booking j
SLR_i	Mix group cluster assigned to ULD i
SLR_j	Mix group cluster assigned to booking j
p_c	Penalty value for not following the loading advice
p_s	Penalty value for spreading one booking across multiple ULDs
$C(j_1, j_2)$	Compatibility matrix for special handling products, if two bookings j_1 and j_2 are compatible, the value of $C(j_1, j_2) = 1$, and otherwise 0

Three different decision variables are used in the model and given in Table 5. It is chosen to have two variables for item to ULD assignment. This is because in practice one booking could have multiple items and sometimes would even require multiple ULDs to pack all items from that booking. Therefore the variable y_{ij} is introduced. For the item segregation and compatibility constraints there is z_{ij}.

Table 5: Decision Variables for the Bin Packing model

Decision Variable	Description
x_i	1 if ULD i is built, 0 otherwise
y_{ij}	Number of pieces from booking j loaded into ULD i
z_{ij}	1 if at least one piece of booking j is loaded into ULD i, 0 otherwise

The Bin Packing model consists of two main elements. The first is the T-ULD building heuristic. In this heuristic, the bookings assigned to T-ULD groups from the Loading Advice are consolidated onto T-ULDs. This approach is taken as the building of T-ULDs is relatively straightforward, and presolving it could save some runtime. The inner functioning is described by Algorithm 1.

When this phase is finished, the items that are packed on T-ULDs are removed from the booking list. The resulting booking list is then solved exactly using the solver `Gurobi`[1]. This 1DBPP is an adaptation of conventional Bin Packing models and the MILP formulation is given by Equation 3. The minimization objective function consists of three terms. The first term is the conventional term for Bin Packing Problems and is equal to the unused space. The second term minimizes the deviation from the loading advice and the weight of this objective is p_c. The third and last part is to minimize the spreading of bookings across multiple ULDs, where the weight for this objective is p_s. The first objective term of minimizing the unused space becomes less important if the weights p_c and p_s increase. It is therefore expected that if these weights are increased, the total number of ULDs will also increase. As the first term is about unused space, it is expressed in m^3. So essentially both p_c as p_s can be considered the amount of unused space the model is willing to sacrifice to avoid spreading of shipments across multiple ULDs and to avoid deviation from the loading advice. Minimizing unused space and number of ULDs is the main objective for airlines and the other two objective terms would be desirable but not the main objective in palletization. Therefore, a good setting to approach the reality would be a very low value of p_c and p_s. A sensitivity analysis for the p_c weight was performed and can be found in appendix A.

[1] Intel® Core™ i5-5200U CPU @2.20 Ghz with 4 GB of memory

Algorithm 1 T-ULD building heuristic

Input Set of advised T-ULD's T, containing bookings from set J, set of ULDs I, matrix of booking compatibilities C.
Initialize ULD: $i = 0$
Initialize empty list of packed items Y
for $t = 1$ to T **do**
  Initialize remaining space capacity $V_r = VCap_i$
  Initialize remaining weight capacity $W_r = WCap_i$
  for j = 1 to J bookings in t **do**
    Initialize number of unpacked items $N_r = N_j$
    if Present items in ULD i are incompatible with booking j **then**
      Continue with $j = j + 1$
    end if
    Calculate maximum to be packed items $y_{ij} = min(W_j/W_r, V_j/V_r, N_r)$
    Add y_{ij} to list of packed items Y and update V_r, W_r and N_r
    if $V_r < 1m^3$ **or** $W_r < 500kg$ **then**
      Close ULD and go to ULD $i = i + 1$ and reinitialize V_r, W_r
    end if
    if j is last booking in t **and** T-threshold has been met **then**
      Close ULD and go to ULD $i = i + 1$ and reinitialize V_r, W_r
    end if
    if j is not last booking in t **and** T-threshold has not been met **then**
      Go to next booking $j = j + 1$ in t
    end if
    if j is last booking in t **and** T-threshold has not been met **then**
      Empty ULD, go to next T-group $t = t + 1$ and reinitialize V_r, W_r
    end if
  end for
end for
Output List of pre-packed items Y

The model contains common constraints such as the volume capacity (Constraint 3d), weight capacity (Constraint 3e) and that every item should be packed once (Constraint 3c). Additionally, there are extra constraints for the item compatibility due to special products. First there is Constraint 3f, which ensures that z_{ij} always takes the value of 1 if at least one item of booking j is packed into ULD i. Then Constraint 3g, which enforces that if two items are incompatible, the sum of the two z_{ij}'s should be less or equal than 1. This constraint only applies to a subset SHR of items that is incompatible with each other. This subset is mathematically formulated as $SHR = \{(j_1, j_2) \mid C(j_1, j_2) = 0\}$, which in words includes all booking combinations (j_1, j_2) for which the Compatibility matrix $C(j_1, j_2)$ is equal to 0.

$$
\begin{aligned}
\textbf{Min.} \quad & \Big(\sum_{i\in I} x_i * VCap_i - \sum_{i\in I}\sum_{j\in J} y_{ij} * V_j\Big) + \Big(p_c * \sum_{i\in I}\sum_{j\in J} |SLR_i - SLR_j| * y_{ij}\Big) && && \text{(3a)}\\
& + \ \Big(p_s * \sum_{i\in I} z_{ij}\Big) && && \text{(3b)}\\
\text{s.t.:} \quad & \sum_{i\in I} y_{ij} = N_j && ,\forall j \in J && \text{(3c)}\\
& \sum_{j\in J} V_j * y_{ij} \le VCap_i && ,\forall i \in I && \text{(3d)}\\
& \sum_{j\in J} W_j * y_{ij} \le WCap_i && ,\forall i \in I && \text{(3e)}\\
& z_{ij} * N_j \ge y_{ij} && ,\forall i \in I, \forall j \in J && \text{(3f)}\\
& z_{ij_1} + z_{ij_2} \le 1 && ,\forall i \in I,\ \forall (j_1, j_2) \in SHR && \text{(3g)}\\
& x_i \in \{0, 1\} && ,\forall i \in I && \text{(3h)}\\
& y_{ij} \in \{0, N_j\} && ,\forall i \in I,\ \forall j \in J && \text{(3i)}\\
& z_{ij} \in \{0, 1\} && ,\forall i \in I,\ \forall j \in J && \text{(3j)}
\end{aligned}
$$

3.3 Breakdown Scheduling model

After arrival of a flight or truck all M-ULDs are disassembled by the Breakdown department. During breakdown, the shipments are removed from the ULD and then transported to the build up location of their outbound flight from the hub. The urgency of a ULD is determined by the shipment with the shortest connection time on each ULD. This is based on the constraint that shipments have to be ready for build up 4 hours before departure of a flight because otherwise the shipments might not make the connection on time. The Breakdown Scheduling model is largely based on the build up scheduling model from Emde et. al (2020) [Emde et al., 2020]. The model is formulated for whole teams consisting of five handlers. In this configuration it was assumed that it requires 20 minutes to break down a whole ULD with one team. It is not possible to schedule more teams or half a team, which makes that the model is insensitive to the number of workers k as this is by default 1. It is also important to note that time is discretized to a set T time windows of 20 minutes, so it is not a continuous problem. For the model, three sets are included as shown in Table 6. A shift s lasts 8 hours, and therefore it contains a subset from T of $8 * 3 = 24$ time windows t. One day contains three shifts, which are the morning shift (06:00h - 14:00h), the midday shift (14:00h - 22:00h) and the night shift (22:00 - 06:00).

Table 6: Sets for the Breakdown Scheduling model

Set	Description
I	Set of i jobs (Breakdown ULDs)
T	Set of t time windows of 20 minutes
S	Set of s shifts of 8 hours, is a set of subsets from T

The Breakdown Scheduling model contains a small set of parameters that are given in Table 7. The release time, or the time the ULD i becomes available is given as a_i. This time is also discretized to a corresponding time window t. The release time a_i is set as the arrival time at the hub + 1 hour. This is to compensate for the time it takes to arrange all documentation and to transport it to the breakdown location in the warehouse. Similarly to a_i, the deadline time d_i is also discretized to a corresponding time window t. The deadline time d_i is determined as 4 hours before departure from the hub of the earliest connecting flight on the ULD. The 4 hours is needed as the shipments also have to be arranged onto new ULDs again and have to be transported to the aircraft. The processing time of one ULD is set to $p_i = 20$ minutes, which is equal to one time window t as explained before. The variable K_t can be changed in the future, but for now it is set to a high value to ensure the model finds an optimal number of breakdown teams to break every ULD down in time.

Table 7: Parameters for the Breakdown Scheduling model

Parameter	Description
a_i	Release time window of job i
d_i	Deadline time window of job i
p_i	Processing time of job i, set to 1 time window (= 20 minutes)
K_t	Number of breakdown teams available at time window t

The model is formulated as a MILP model. The decision variables are summarized in Table 8. Here α_t corresponds to the number of breakdown teams required in time window t. Then for all time windows t in shift s, the maximum α_t is equal to to the required breakdown teams β_s in shift s, which is also an integer variable. The last decision variable x_{it} is binary and stands for whether job i is executed in time window t. As is common for binary variables x_{it} takes the value 1 if this is case, and otherwise it is equal to 0.

The MILP formulation is provided in Equation 4. The objective (Equation 4a) is formulated as minimizing the sum of breakdown teams required in every shift. This is because it is assumed the shift capacity is fixed, which is often the case in cargo terminals. Constraint 4b ensures that α_t will be equal to the number of teams busy in time window t. Constraint 4c restricts the maximum number of teams busy in time window t to the number of teams available at time window t. As each ULD should be broken down exactly once, Constraint 4d sets the sum of x_{it} over all time windows t to 1, this is done for every ULD i. Constraint 4e and Constraint 4f ensure that every breakdown task is not started before the release time and not finished after the deadline time, respectively. Lastly, Constraint 4g forces β_s to take the largest value of all α_t's that are part of shift s.

Table 8: Decision Variables for the Breakdown Scheduling model

Decision Variable	Description
α_t	Number of Breakdown teams busy at time window t
β_s	Breakdown Teams required for shift s
x_{it}	1 if job i is executed in time window t, 0 otherwise

$$\textbf{Min.} \quad \sum_{s \in S} \beta_s \tag{4a}$$

$$\text{s.t.:} \quad \sum_{i \in I} x_{it} \leq \alpha_t \qquad , \forall t \in T \tag{4b}$$

$$\alpha_t \leq K_t \qquad , \forall t \in T \tag{4c}$$

$$\sum_{t \in T} x_{it} = 1 \qquad , \forall i \in I \tag{4d}$$

$$\sum_{i \in I} t * x_{it} \geq a_i \qquad , \forall i \in I \tag{4e}$$

$$\sum_{i \in I} (t + p_i) * x_{it} \leq d_i \qquad , \forall i \in I \tag{4f}$$

$$\beta_s \geq \alpha_t \qquad , \forall t \in s, \forall s \in S \tag{4g}$$

$$x_{it} \in \{0, 1\} \qquad , \forall i \in I, \ \forall t \in T \tag{4h}$$

$$\alpha_t \in \{0, K_t\} \qquad , \forall t \in T \tag{4i}$$

$$\beta_s \in \{0, K_s\} \qquad , \forall s \in S \tag{4j}$$

4 Description of the Case Studies

This research considered the hub and spoke network of a large combination airline. The data is anonimized from this section onwards due to the commercially sensitive nature of the data. This airline has a set of rules of thumb in place to improve the workload like previously mentioned in section 1. The new loading advice model is compared to the set of rules this airline uses. This current situation will be referred to as the Baseline in the results. A cargo flow is selected of continental outstations to an intercontinental outstation through a hub.

As the origin outstations are on the same continent as the hub, the seven outstations that were selected for this study are connected by a trucking leg to the hub. Trucks have a flexible capacity, as additional trucks can easily be scheduled. Only Lower Deck Pallets are transported in trucks, which have a volume capacity of $11m^3$ and a weight capacity of $4300kg$. A truck can transport four of these pallets. Additionally, the Aircraft Configuration Problem and Weight and Balance Problem are not applicable to Trucks. This makes trucking legs a good candidate for a test case as the potential gains are much larger and the problem itself is less complex. Below, in Table 9 the properties of the seven selected origin outstations are given. A variety of stations is selected with different volumes of cargo and the distances to the hub. For trucking, the distance to the hub is the main factor that determines the trucking cost. An additional factor is the uncertainty. Travel times of trucks are larger than for aircraft and can even exceed the 24 hour mark for far away stations. From a practical point of view this means that the further away an outstation is, the earlier the loading advice should be issued and the more uncertain the incoming workload at the hub is. An expected side effect of the dynamic T-threshold of the Loading Advice model on the trucking case is that the overall number of ULDs will increase, which in turn leads to more trucks as trucks have a capacity of only four ULDs. A lower T-threshold will result in more unused space on T-ULDs. More unused space might result in more ULDs required to transport all cargo, because the to be transported cargo remains the same. Furthermore, T-ULDs on its own have a lower volume utilization than M-ULDs, so more T-ULDs might also result in more ULDs required to transport all cargo. Therefore, it is expected that there is an increase of the number of trucks for every increase of the parameter b.

Name	Volume $[m^3]$ of cargo	Distance to hub
A	Large	Medium
B	Medium	Large
C	Large	Small
D	Medium	Medium
E	Small	Small
F	Small	Large
G	Medium	Medium

Table 9: Properties of selected origin outstations

In the model, the whole network of the combination airline was considered in terms of destinations. However, three destinations are selected for this paper to show the effects of the model. These three destinations with their respective properties are given in Table 10. Destination 1 has medium LFs and a medium contribution and is therefore selected as a baseline and normalized to 1. When looking at destination 2 and 3, it can be seen that destination 2 has a low LF and low contribution compared to destination 1. For destination 3 this is exactly the opposite. The expected outcome of the model is therefore that destination 2 should be used to create additional T-ULDs and that destination 3 will have less T-ULDs.

Name	Average LF	Average Contribution
Destination 1	1.00	1.00
Destination 2	0.44	0.69
Destination 3	1.38	6.08

Table 10: Properties of selected destination outstations, normalized to destination 1

A simulation is performed of the full year of 2019 as this was the most recent year without COVID-19. This makes it the most recent year that is not affected by heavy disruptions, which makes it the most recent representative year in terms of seasonal trends. In Figure 6 the cargo tonnes for the selected hub is normalized to the maximum of that year (March). This figure includes the cargo of all airlines for our selected hub and therefore does not 100% correspond to the trend of our partner airline.

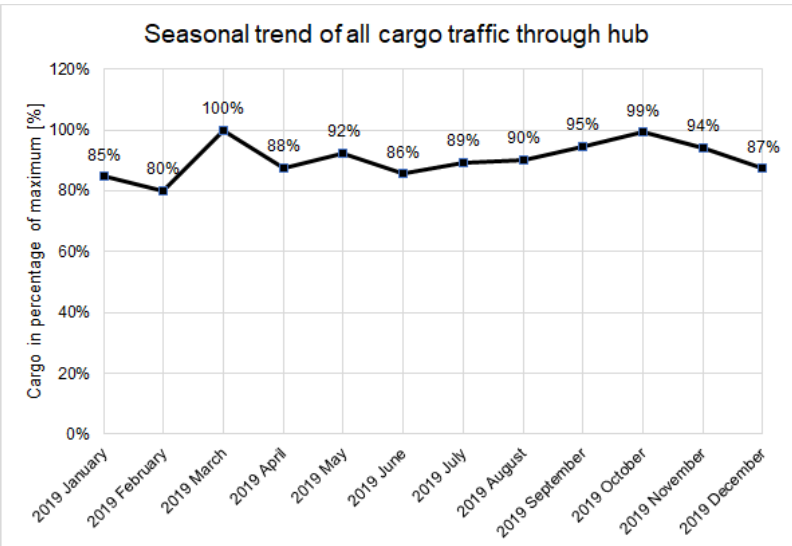

Figure 6: Seasonal trend of all airlines through the selected hub in 2019 [CBS, 2019]

There are a few scenario's for which this tool could be applicable. A set of three cases is chosen to test the impact of the dynamic T-threshold, which will be described below.

Case study 1: Weekend intervention

The weekends are the busiest days of the week for cargo airlines due to the nature of the cargo market. Most shippers and consignees are closed in the weekend and they therefore try to get rid of all the cargo just before the weekend on Fridays. This makes the weekend the most likely day of a shortage of handling capacity. This case is implemented by using a weight setting of $b = 6$ in the weekend and $b = 0$ during the week.

Case study 2: Shortage of Trucks
Until here the focus has been on reducing workload by increasing the number of T-ULDs arriving at the hub to reduce the M-ULDs. An increase in T-ULDs by setting lower T-thresholds probably comes at the cost of extra ULDs, as you leave more capacity unused. This will in turn result in a need for extra trucks. Recently, there has been a shortage of trucks in the European Union, which motivates this scenario. The question is if the developed tool can be used to reduce the number of trucks, and if so, what would be the maximum reduction of trucks. This is implemented by using a negative value for the weight setting b as described in section 3.1.

Case study 3: Intervene at closeby stations
The trucking costs are higher and the uncertainties a lot larger for far away stations, therefore it could be beneficial to only use the proposed dynamic T-threshold for closeby stations to minimize the additional trucking cost while still reducing workload. For this case all small and medium distance stations (Origin A, C, D, E and G) were used to intervene with weight setting $b = 6$ while the other far away stations (Origin B and F) used the standard rules of the airline.

5 Results

In this section the results of the model will be presented. In section 5.1 the effects of the dynamic T-threshold will be discussed, followed by the M-ULD clustering effects in section 5.2. Lastly the case studies are presented in section 5.3. In this chapter the values for the following parameters were fixed; $w_{LF} = 1$, $w_{Cb} = 1$, $p_c = 0.1$ and $p_s = 0.1$. Both w_{LF} and w_{Cb} are set to 1 to make both factors weigh equally in the Correction factor. A sensitivity analysis of these parameters has been performed and can be found in the appendix. The penalty values p_c and p_s were set on a value that it would not result in extra ULDs, but the values are high enough to encourage the model to follow the loading advice and discourage the spreading of shipments across multiple ULDs.

First the computational time ranges of all model parts can be found in Table 11. These results are produced on an Intel® Core™ i5-5200U CPU @2.20 Ghz with 4 GB of memory. All results are for a whole month and the seven origin outstations as presented in section 4. The runtime of the Loading Advice model is relatively low which is a good property considering the applicability in practice. This would mean it only takes a few seconds to provide an experienced planner with a loading advice solution. The low computational time for the T-threshold part can be attributed to the relative uncomplicated computation, which makes it faster. Regarding the clustering part, the used clustering method is highly efficient. Besides, the data sizes are still small which also helps in keeping the computational times low.

The Bin Packing model was expected to be computationally more demanding due to the exact approach, which is shown by the higher computational times. Regarding the Breakdown Scheduling model, the computational time is also relatively high which can be attributed to the high number of decision variables. One day contains 3 shifts s and 72 time windows t. In a month with 1000 M-ULDs and 30 days this will result in more than two million decision variables. This could have been optimized by only generating x_{it} decision variables when the ULDs are physically in the warehouse. However achieving low computational times were not the main goal of the model. The purpose of the Breakdown Scheduling model was to study the effects of the previous parts.

Model Part	**Runtime**
Loading Advice model	15 - 30 seconds
Bin Packing model	20 - 30 minutes
Breakdown Scheduling model	5 - 10 minutes

Table 11: Computational results of different model parts for a whole month

5.1 Dynamic T-threshold

The dynamic T-threshold is expected to decrease the number of M-ULDs, where a higher value of the weight b leads to a larger reduction of M-ULDs. This is achieved by creating more T-ULDs. Overall it is expected that this will result in more ULDs, which in turn leads to more Trucks. Even though the capacity of trucks is flexible, it does have a negative effect for cargo airlines in the form of additional cost.

In Figure 7 the effects of different values for the weight b on the number of M-ULDs are shown over the year. It shows a stepwise reduction for every increase of the weight b. Besides this, there are seasonal trends visible where in some months the effects are less large than in others. The graph can be dissected in two parts, the trend of the weight b and the seasonal trend.

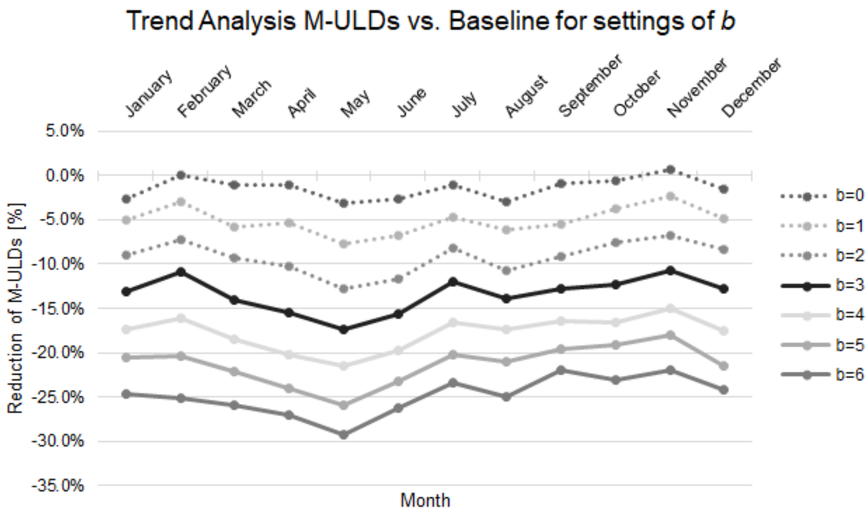

Figure 7: Seasonal trend for different values of b on number of M-ULDs

The sensitivity of the weight b is shown in Figure 8. The graph shows a linear trend for every setting of b. The trend could be described by a linear function with a slope of 3.97%, with a determination coefficient $R^2 = 0.999$. This means that every 1 point on the weight b can reduce the incoming work by almost four percent. However, a side note should be added that this is only determined for 7 data points and requires some more to confirm the trend. It does give an initial idea of the sensitivity of the weight b. Another observation is the reduction of M-ULDs for $b = 0$. The model is compared to a set of rules of thumb of a combination airline. With a value of $b = 0$ there is already an improvement of workload compare to the current situation. The setting $b = 0$ is solely based on the LF of the outgoing flight because $b = 0$ neutralizes the correction term in Equation 1. When basing the LF on the outgoing flight only it is ensured that there is space for all ULDs to load in the aircraft. For every setting of the weight b the risk of not fitting is increased.

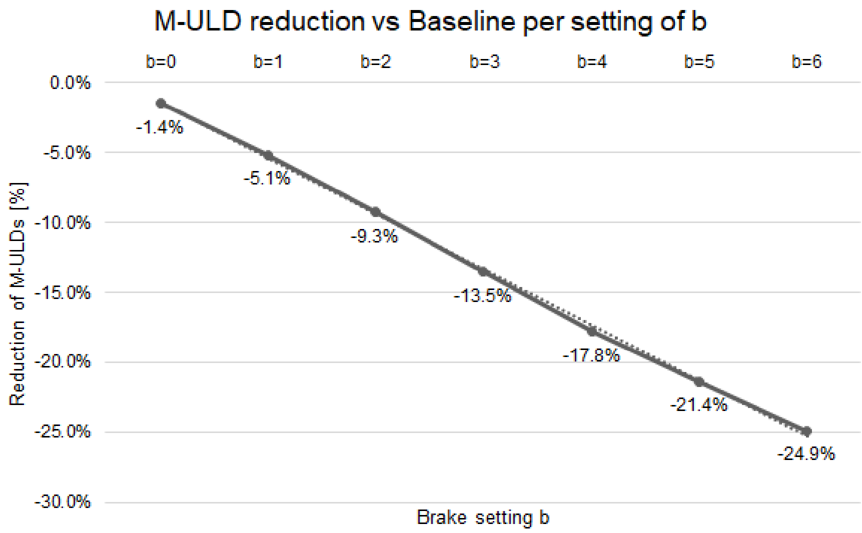

Figure 8: Effect of settings of b on number of M-ULDs

The underlying seasonal trend is depicted in Figure 9. This value is determined by calculating the average of all jumps between different values for b. It shows that the model is more sensitive in the first few months of the year (February till May) and from the summer onwards till January the model becomes less sensitive for the weight b. The largest gap between months is between September and May with a gap of 0.8%. This was expected for the last months of the year and January, as the demand for air cargo transport is naturally larger in these months, due to the Christmas period. This means higher LFs for airlines and therefore a higher T-threshold. The dip from February till May was expected to extend over the summer period, because in the holidays a lot of companies are not running on full capacity, resulting in less shipments and therefore in less demand, which leads to lower cargo LFs and also in lower expected T-threshold. A possible explanation however is that there are much more passengers in the summer months, which reduces the capacity in the cargo holds of a combination airline, mainly in weight but also in volume.

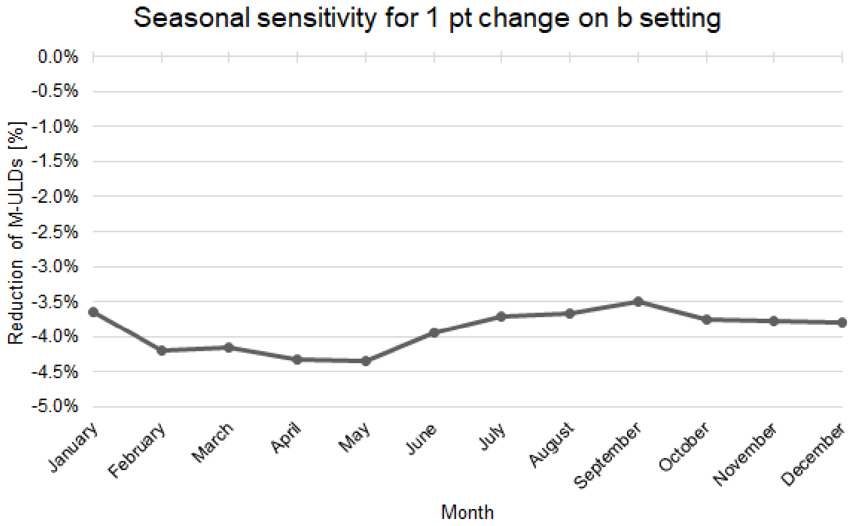

Figure 9: Underlying seasonal trend in Figure 7

In order to show which destination contributes to the extra T-ULDs and less M-ULDs an insight is given in Figure 10. The properties of the destinations can be found in Table 10. It can be concluded that the dynamic T-threshold has the desired effect. Destination 2 is considered as a low priority destination as it has low LFs and low Contributions. Therefore the model advises and builds much more T-ULDs for that specific destination. Destination 3 is a high priority destination, and the results show that even when using the largest setting of b, there are still less T-ULDs advised and consolidated for that destination. Destination 1 is considered a medium priority, and lies in between the two other destinations. The figure is also in line with what is to be expected, following Figure 8, that the larger b becomes, the more T-ULDs are built and the less M-ULDs.

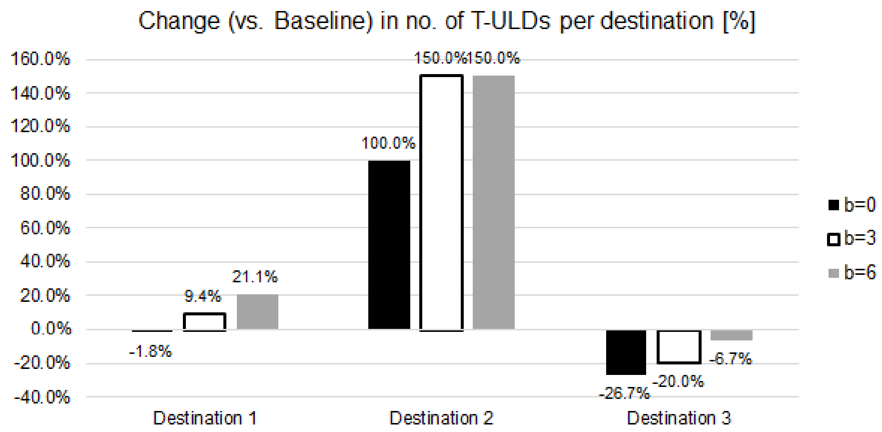

Figure 10: Resulting T-ULDs per destination outstation

Creating more T-ULDs to reduce the workload comes at a cost however. It was to be expected that the method would result in additional trucks and this is confirmed by Figure 11. Here the runs for the different values of b are shown and the resulting difference with the baseline in Trucks and M-ULDs. Once again there is a linear trend visible for the spectrum of settings used. For every 3.3% reduction in M-ULDs the number of Trucks required increases by 1%. This linear relation has a coefficient of determination $R^2 = 0.9963$.

Figure 12 shows a to be expected effect of larger costs for far away stations (Origin B and F). This figure in itself is a motivation for case study 3 of only intervening at origin outstations that are close to the hub. The consideration of deploying the model is airline specific, but it can be determined by calculating the labour cost of handling one extra ULD and compare it to the cost of trucks. This can provide the cargo airline with a motivation of which stations to use to decrease the total workload.

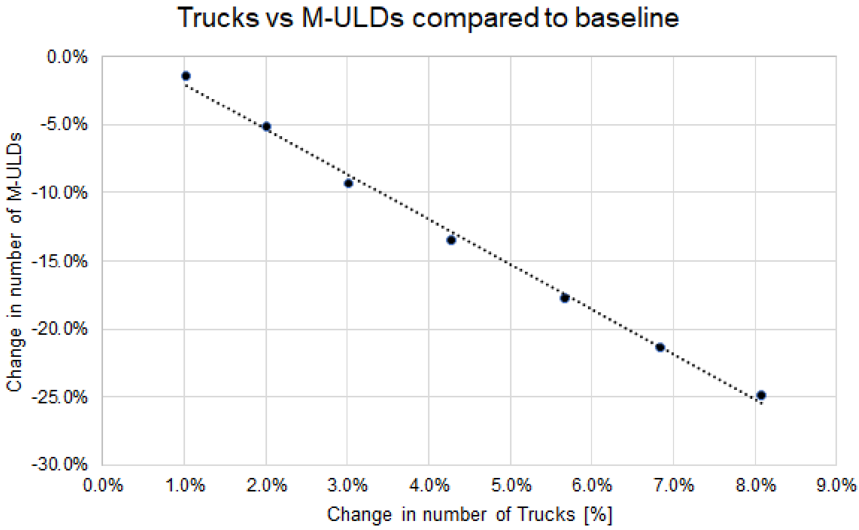

Figure 11: The ratio between M-ULDs and Trucks

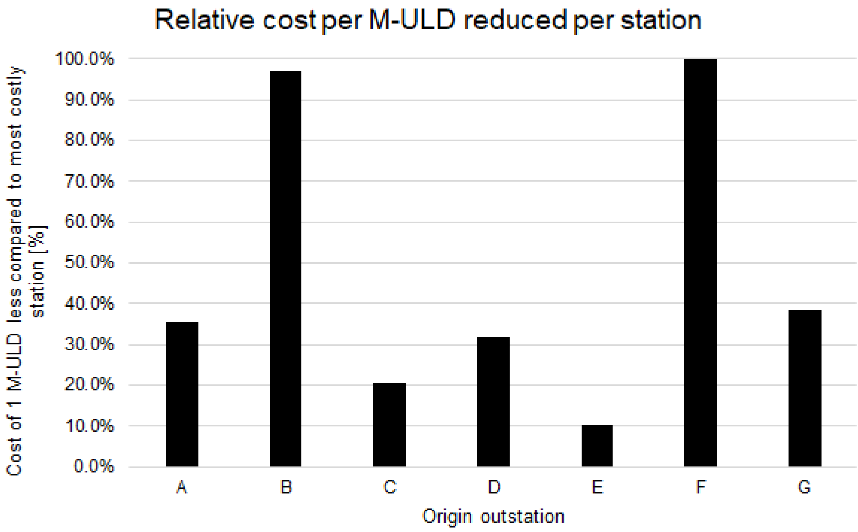

Figure 12: The associated trucking cost of reducing a M-ULD per origin, normalized to origin F

5.2 M-ULD Clustering

By using a clustering algorithm instead of a set of fixed rules it is expected to have better distributed workload in the hub. The rules are based on fixed thresholds in terms of connection times in the hub. This situation is compared to the clustering algorithm which clustered cargo on connection times as described in section 3.1. Both the baseline as the clustering algorithm were not run with a dynamic T-threshold. The effect on the number of Breakdown teams required is shown in Figure 13. It shows that for all months except June and August the clustering method does not result in fewer Breakdown teams, but in more required Breakdown teams. This is not the outcome that was expected, but it does confirm the assumption that changes in the composition of an ULD can result in significant differences in the required amount of Breakdown teams.

The unexpected outcome can be explained by Figure 14. Here the distribution of time windows is shown in a boxplot. It shows that the median time window value is smaller with the use of the clustering algorithm than in the baseline situation. This indicates that the clustering approach generally creates more short deadline ULDs than the baseline, meaning a higher pressure is put on the hub which leads to a higher peak workload instead of the expected lower peak workload. This could explain the unexpected outcome in Figure 13. The third quartile value is larger with the clustering approach, which means that after the first peak in workload the deadlines are better distributed. This could indicate that despite the logarithmic function the differences in small connection times are still not enlarged enough to create good clusters for the hub.

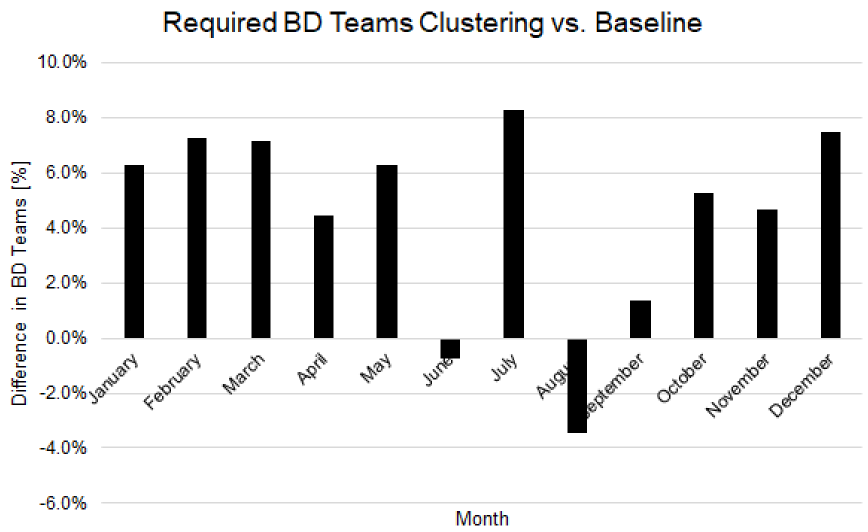

Figure 13: The monthly result of using clustering for M-ULDs compared to baseline

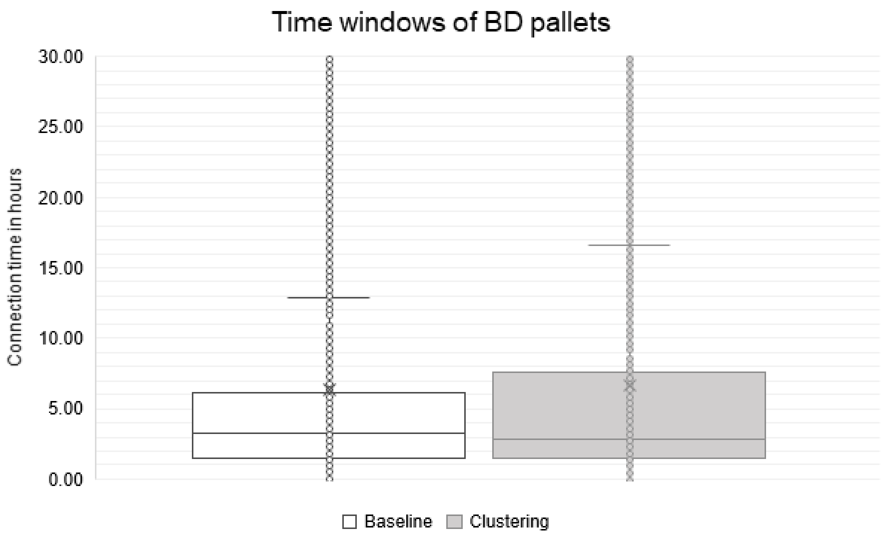

Figure 14: Boxplot of Breakdown time windows of M-ULDs at the hub

5.3 Case studies

In Figure 15 the result of the weekend case on the required number of breakdown teams is shown. In the figure it can be seen that it resulted in a reduction of the amount of required Breakdown teams to break all ULDs down in time. It is important to realize with this graph that only the cargo from the seven origin outstations were considered, meaning that only a fraction of the workload in the hub was taken into account in determining the required amount of Breakdown teams. It was tried to synthesize the data from the seven origin outstations but the difference with the actual ULDs and the required breakdown teams was too large to proceed. Even though the case is not completely accurate of what would happen in practice, it does show a reduction in workload for the weekends which also to be expected from the 24.9% reduction of M-ULDs with $b = 6$ as given in Figure 8.

The results of the truck shortage case are shown in Figure 15. The negative value of b, which results in a higher T-threshold, has the negative expected effect on the number of M-ULDs, which increases ranging from 8.4% to 13.7%. On the other hand is also results in the desired outcome of a reduction of Trucks, depending on the month this fluctuates between a reduction of 0.6% and 1.9%. As this is the truck optimal solution, it suggests that this specific airlines' current approach is mostly aimed towards truck optimization because the gap with the baseline is small.

In Table 12 the application of the dynamic T-threshold model for close stations is compared to similar results in M-ULD reduction. Here it is clear that using the model for only close stations could be very beneficial. With this setting a M-ULD reduction of 20% can be realized which is 27.0% and 27.7% less expensive per M-ULD

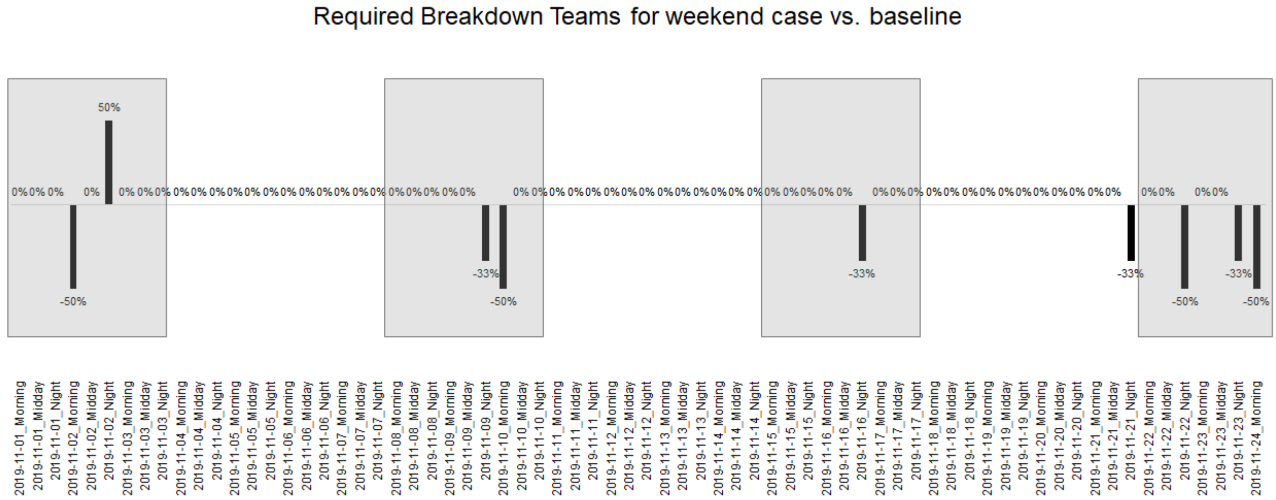

Figure 15: Example of the effect of Weekend case (November '19)

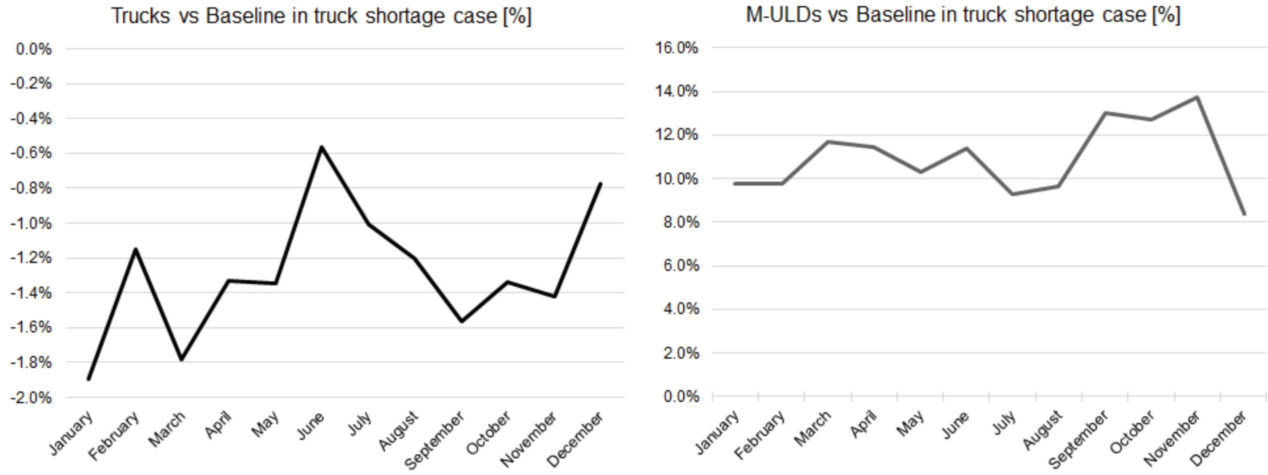

Figure 16: Effect of truck shortage case on number of Trucks (left) and number of M-ULDs (right)

than $b = 4$ and $b = 5$ respectively, while achieving similar results.

Setting	Δ M-ULDs	Δ Truck Cost	Cost per M-ULD
$b = 4$	-17.8 %	5.90 %	99.1 %
$b = 5$	-21.4 %	7.15 %	100 %
Close Trucking stations case	-20.0 %	4.83 %	72.3 %

Table 12: Comparison of Truck closeby case vs. similar settings of b

6 Discussion

The advantages of the given approach is that it is implementation ready. Given the current lack of data and low data quality in air cargo operations this is a good first step in realizing a more integrated approach between two subproblems. It gives airlines the opportunity to match the workload to the available workforce, which has only been possible the other way around. Additionally, due to the high computational times of for example three dimensional models, a large scale impact assessment as executed in this research would not have been possible in the given timeframe for this research.

The choice of creating an implementation-ready model has some downsides. The first is that a specific Bin Packing solution for an outstation would always be more optimal than a loading advice construction, as this is an additional optimisation step. However, as bookings do not necessarily translate into the same physical shipment, an exact item to ULD configuration, which is provided by a Bin Packing model, would not be feasible in practice. This is because shipment weights, volumes and dimensions are not always given or accurate. Besides, there is the possibility of a no-show, which means that the booking is not even delivered to the outstation at all. A second limitation of the approach is that both the Bin Packing model as the Breakdown Scheduling model can be more detailed. The addition of extra constraints and dimensions to both these models will result in quickly increasing computational times, but in the future when more efficient models or better computers are available both these models can be extended to increase the solution's accuracy.

6.1 Dynamic T-threshold

Following the results in section 5, it can be concluded that the dynamic T-threshold is an appropriate method to artificially reduce the workload. It shows a predictable linear pattern in the reduction of M-ULDs but also in the increase of trucks. Furthermore, it also creates an expected value distinction between certain destinations, where more profitable and high load factor flights are protected at the cost of less profitable and low load factor flights. It also shows that it is seasonal dependent and the effect of the model is less large in more busy months, which can be attributed to the higher load factors in these months. When considering trucking stations it is clear that this tool is far less costly for small trucking distances than for large trucking distances.

A possible side-effect of this model is lower achieved Load Factors at outbound flights. It can be assumed that the realized volume utilization of a T-ULD decreases with the increase of the setting b. At a certain point the realized volume utilization can dip under the Load Factor of the specific flight. This introduces a risk of not fitting all shipments in the cargo hold as a result of the dynamic T-threshold. This is not investigated in this research and this could be included in the future as well as the potential lost value associated to the lower achieved load factors. It is not included for now because from a practical point of view, a T-ULD can always be sent back to the build up department in case some extra handling capacity opens up. This means that for every T-ULD airlines still have the opportunity to handle them and add some extra items. For M-ULDs this is not the case as they always require handling which involves complete breakdown and build up. So from a planners perspective, it could be advised to always choose a setting that is slightly higher than the required workload reduction.

This leads to the question of when and where to deploy this model with which setting. Because of the introduction of a risk factor it is advisable to always choose the lowest setting that matches the workload to the workforce. Additionally, the model should be deployed more heavily at closeby stations than far stations because of the higher trucking cost the further away the outstation is from the hub. Another reason is that the closer to the hub, the smaller the transit time. This means that the airline has a more clear picture of the incoming workload when issuing the instruction for closeby stations.

6.2 M-ULD clustering

The K-means clustering method for M-ULDs was expected to create a more distributed pattern of workload than the current rules in place at the partner airline. The results do not confirm this expectation as the clustering method results in a larger amount of required breakdown teams. It can even be concluded that the current rules are more effective than the clustering method. It also shows that the first half of the M-ULDs created by the clustering algorithm have a shorter deadline, but the second half has a better spread of deadlines. It could therefore be argued that the clustering algorithm does work better in case some missed connections have to be recovered due to the better spread in longer connections. On the other hand the clustering algorithm itself would also cause more missed connections resulting in a higher amount of to be recovered shipments.

The implementation of volume constraints for the clustering algorithm as mentioned in section 3.1 could result in an improvement of the model. It was not implemented in this research as the current clustering model did not facilitate the addition of constraints. Another possible improvement can be found in an alternative for the ^{10}log transformation to put more emphasis on short connections than long connections. An third improvement could be the second input variable. An airline would be able to use this input to make a value distinction in M-ULDs. The airline would therefore be better able to prioritise certain ULDs. Examples of high priorities are ULDs with more profitable cargo, cargo on high LF flights or cargo with a high commercial priority. These three suggested improvements could still result in a better fit for M-ULD clusters, but this does require additional research.

7 Conclusions

This research is a first try of combining the considerations of two subproblems of the Air Cargo Load Planning Problem defined by Brandt and Nickel (2019) [Brandt and Nickel, 2019]. The objective was to create a model that uses the palletization of shipments onto ULDs at outstations to influence the workload at the breakdown and buildup department at the hub. In literature this is referred to as the Air Cargo Palletization Problem and the Build-up Scheduling Problem. A better match between workload and workforce results in more cargo making its connection in the hub. This in turn leads to more cargo delivered on time which has a positive impact on an airlines' customer satisfaction and achieved LFs on flights. In an air cargo terminal's perspective there are two types of ULDs, which are the Mixed-ULD (M-ULD) and Through-ULD (T-ULD). Here, the T-ULD does not require handling at the hub and is directly transferred from the inbound flight/truck leg to the outbound flight/truck leg. The reduction of workload at the hub was split into two parts; the overall workload and the peak workload. Both parts have their own modelling approach and were compared to the baseline rules an airline currently uses to address both these parts. In order to make the approaches applicable in practice it was implemented using a model that is able to issue similar loading advices to outstations as currently used in practice. A 1D Bin Packing model with a heuristic and MILP combination converted the loading advices to ULD configurations. Lastly the M-ULDs from the Bin Packing model were used as an input for a Breakdown Scheduling model to study the effects on the breakdown department in the hub. The three models were run for a simulation of the full year of 2019 with seven continental outstations to study the effects of the new loading advice model.

A dynamic T-threshold was introduced to influence the overall workload at the hub. This threshold can be used to artificially increase the number of T-ULDs and decrease the number of M-ULDs. The goal of the dynamic T-threshold is to create a tool that can be used to decrease the workload such that the hub is able to handle all cargo. In the dynamic T-threshold the value and LF of outbound flights was taken into account to determine which flights to protect and which flights to compromise. In the results it is shown that the proposed tool works as expected and it also shows a predictable behaviour in the form of a linear trend. This makes the tool already applicable in practice.

The use of a K-means clustering method was researched as a method to reduce the peak workload. In the current situation rules were in place to group cargo based on its connection time. These rules had fixed thresholds and it was therefore expected that a clustering algorithm could find better fitted groups, as a clustering method determines groups based on patterns in data instead of grouping cargo based on fixed threshold values. However this expectation did not turn out to be true, as the clustering method had the opposite effect, causing higher peaks in the workload resulting in a larger amount of required teams to handle all cargo in time. The proposed clustering technique is therefore not an improvement on the baseline situation and would require more research before implementing it in practice.

8 Recommendations

The recommendations for future research can be divided into three categories. The first category is the analysis of case studies. In this research an analysis of a full year of seven continental outstations was performed. This gave some good insights into the functioning of the proposed model, but in order to understand the full scale effects on the hub the case study can be increased to at least a full network of continental outstations. The next step would be the addition of intercontinental outstations. This step does require the considerations of the Aircraft Configuration Problem and Weight-and Balance Problem. Another recommendation for the analysis is to add the risk assessment of not fitting a ULD as explained before in this section. This analysis could give extra insight in the appropriate use of the weight b.

The second category for recommendations is the improvement of the loading advice model in the current modelling approach. The M-ULD clustering obviously needs improvement as it does not result in a better distribution of work deadlines. It can be tried to use different partitional clustering techniques, as well as the introduction of constraints on the minimum volume in a cluster. Another option is to look at alternatives of the log_{10} function to artificially enlarge the differences in small connection times. The dynamic T-threshold uses some weights, with the weight b being the most important, to reduce the workload. However, in the future this could be replaced by a self steering alternative. In this case you could provide the number of available trucks per outstation and the available workforce and then the model would calculate the most optimal solution. This can even be extended by multiple other considerations, such as the available workforce at outstations or the amount of cargo for a specific destination that missed its connection in the hub and needs recovery. Another step into future research would be the use of real-time data in providing a loading advice. In this way the incoming composition of cargo from trucking stations far away from the hub could be taken into account while generating a loading advice for closer stations.

The third and last category of recommendations is for the approach itself. In the future, if air cargo terminals are automated and data is widely available in good quality, it might be better to change the complete approach. This is also the case for developing a model to find a theoretical optimal instead of a practical optimal. In this new approach the purpose of the Loading Advice model could be directly implemented in a highly detailed 3D

Bin Packing model. A large variety of constraints can be included and a multi objective term can be added to minimize the spread of connection times on a single ULD. In theory, this should result in the best possible workload spread. The output would be a 3 dimensional solution which can be followed in the air cargo terminals. The dynamic T-threshold can still be a basis for determining a T-ULD, but in automated cargo terminals the difference in handling time between T-ULDs and M-ULDs might not be as large as it is today.

References

[Bholowalia and Kumar, 2014] Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9).

[Bortfeldt and Wäscher, 2013] Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.

[Brandt and Nickel, 2019] Brandt, F. and Nickel, S. (2019). The air cargo load planning problem - a consolidated problem definition and literature review on related problems. *European Journal of Operational Research*, 275(2):399–410.

[Cargo-IQ, 2019] Cargo-IQ (2019). Cargo iq.

[CBS, 2019] CBS (2019). Aviation; monthly figures of dutch airports.

[Emde et al., 2020] Emde, S., Abedinnia, H., Lange, A., and Glock, C. H. (2020). Scheduling personnel for the build-up of unit load devices at an air cargo terminal with limited space. *OR Spectrum*, 42(2):397–426.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[IATA, 2018] IATA (2018). Cargo strategy 2018. Accessed: 5-3-2021.

[Kaufman and Rousseeuw, 2009] Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

[Lurkin and Schyns, 2015] Lurkin, V. and Schyns, M. (2015). The airline container loading problem with pickup and delivery. *European Journal of Operational Research*, 244(3):955–965.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Ng and Han, 2002] Ng, R. T. and Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016.

[Paquay et al., 2018a] Paquay, C., Limbourg, S., and Schyns, M. (2018a). 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(1):52–64.

[Paquay et al., 2018b] Paquay, C., Limbourg, S., Schyns, M., and Oliveira, J. F. (2018b). Mip-based constructive heuristics for the three-dimensional bin packing problem with transportation constraints. *International Journal of Production Research*, 56(4):1581–1592.

[Paquay et al., 2016] Paquay, C., Schyns, M., and Limbourg, S. (2016). A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. *International Transactions in Operational Research*, 23(1-2):187–213.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[Saxena et al., 2017] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664–681.

[Van den Bergh et al., 2013] Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European journal of operational research*, 226(3):367–385.

[Yan et al., 2006a] Yan, S., Chen, C.-H., and Chen, C.-K. (2006a). Long-term manpower supply planning for air cargo terminals. *Journal of Air Transport Management*, 12(4):175–181.

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

[Yan et al., 2008b] Yan, S., Chen, C.-H., and Chen, M. (2008b). Stochastic models for air cargo terminal manpower supply planning in long-term operations. *Applied Stochastic Models in Business and Industry*, 24(3):261–275.

[Yan et al., 2006b] Yan, S., Chen, C.-K., and Chen, C.-H. (2006b). Cargo terminal shift setting and manpower supplying in short-term operations. *Journal of Marine Science and Technology*, 14(2):6.

Appendices

A Sensitivity of weights

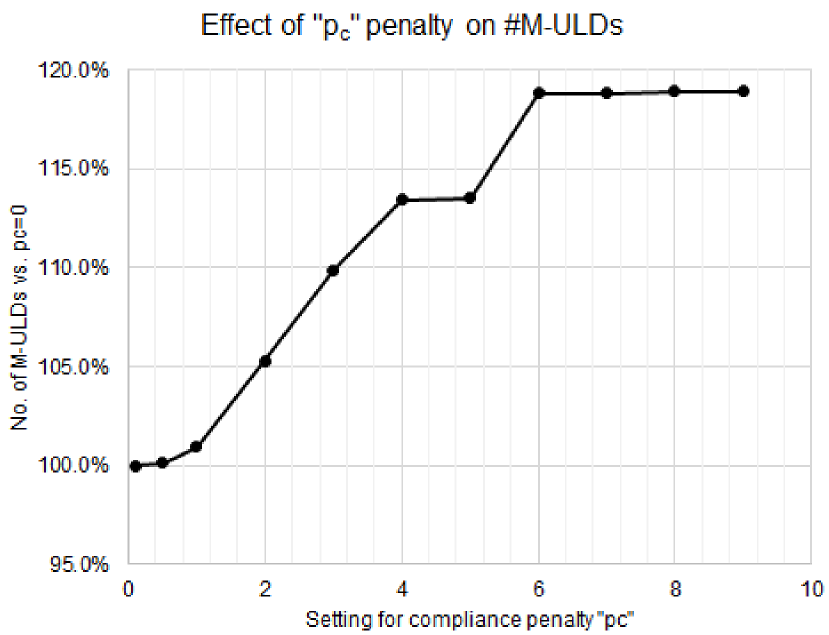

Figure 17: Sensitivity of the p_c penalty on the number of M-ULDs

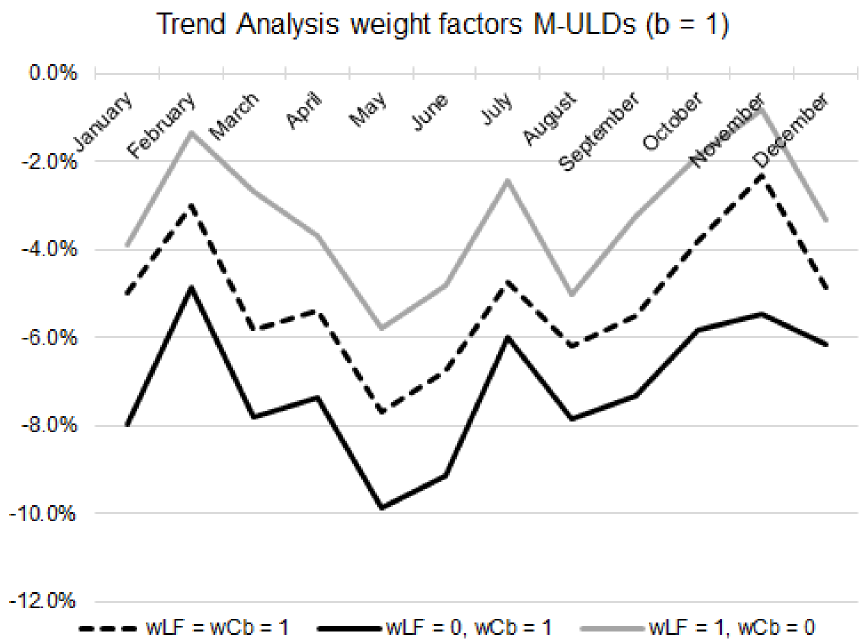

Figure 18: Sensitivity of the two weights w_{LF} and w_{Cb} on the number of M-ULDs

B Table of all simulations and settings

	Dynamic_M	Dynamic_T	b	w_cb	w_lf
1	FALSE	FALSE	0	0	0
2	FALSE	TRUE	0	1	1
3	FALSE	TRUE	1	1	1
4	FALSE	TRUE	2	1	1
5	FALSE	TRUE	3	1	1
6	FALSE	TRUE	4	1	1
7	FALSE	TRUE	5	1	1
8	FALSE	TRUE	6	1	1
9	FALSE	TRUE	1	1	0
10	FALSE	TRUE	1	0	1
11	TRUE	FALSE	0	0	0
12	TRUE	TRUE	1	1	1
13	FALSE	TRUE	Weekend	1	1
14	FALSE	TRUE	Save Trucks	1	1

Table 13: Simulation runs and their respective settings

C Pseudo code of Loading Advice model

Algorithm 2 Loading advice model

Input Booking list J, Flight contributions Cb_{fd}, Flight LFs LF_{fd}, calibration parameter b, LF weight w_{LF}, Contribution weight w_{Cb}, T-threshold function $T_{thresh}\{f,d\}$, Connection times of bookings T_j, Volumes of one piece in bookings V_j, Number of items in booking N_j, Contribution of bookings Cb_j, volume for one datapoint in clustering e, K-means clustering algorithm K

// Determine the T-ULDs

Initialize empty set of T-ULDs T

Make list F of all flight f and dates d combinations in J

for $f = 1$ to F **do**

  Compute $T_{thresh}\{f,d\}$

  Make a list t of all bookings that satisfy $O_j = f$

  sum total volume V_{fd} of bookings in t

  if $V_{fd} > T_{thresh}\{f,d\}$ **then**

    Add t to set T

  end if

  clear t

end for

// Determine the M-ULDs

for $j = 1$ to J **do**

  if $T_j > 30$ **then**

    set $T_j = 30$

  end if

  append to clustering vector $x_j = log_{10}(T_j)$

  Calculate extra datapoints to be added for booking j by $N_j * V_j/e$

  append the extra datapoints to clustering vector $x_j = log_{10}(T_j)$

end for

Sum volumes of all bookings in J not assigned to T-ULDs V_{mix}

Divide V_{mix} by capacity of a ULD $VCap$ to obtain upper bound of clusters u.

for $k = 2$ to u **do**

  Run K-means algorithm K with input x and number of clusters k

  Compute Silhouette value S_k

end for

Select largest value of S_k and the corresponding k

Run K-means algorithm again for this value k

Append mix clusters SLR_j to booking list

Output Mix cluster assignment SLR_j and set of T-ULDs T

D Example of a Loading Advice

Booking	Origin	Destination	Weight	Volume	Pieces	Next Flight	Next DepDate	Cb	Hub Conn [h]	T_thresh	T_ULD	M_ULD
0	B	AA1	3776	10	6	AE0001	3-8-2019	1851.39	3.00	8.09	0	1
8	B	AA7	1700	9.408	7	AE0013	5-8-2019	2190.27	5.00	4.43	1	1
11	B	AB1	301	1.18	3	AE0019	4-8-2019	169.43	6.42	6.02	2	2
46	B	AB1	3440	10	2	AE0019	4-8-2019	2333.48	9.42	6.02	2	3
15	B	AB5	815	6.6	84	AE0025	4-8-2019	568.36	6.58	8.38	3	2
17	B	AB5	877	4.57	4	AE0025	4-8-2019	422.83	6.58	8.38	3	2
18	B	AB6	1500	3.84	4	AE0027	4-8-2019	714.65	6.75	6.93	4	2
47	B	AB6	2300	10	1	AE0027	4-8-2019	2042.64	9.75	6.93	4	3
23	B	AB9	885	2.88	6	AE0033	4-8-2019	433.12	7.00	7.40	5	2
24	B	AB9	1054	5.15	10	AE0033	4-8-2019	401.88	7.00	7.40	5	2
25	B	AB9	245	1.3	10	AE0033	4-8-2019	101.66	7.00	7.40	5	2
27	B	AB9	293	1.41	1	AE0033	4-8-2019	220.66	7.00	7.40	5	2
26	B	AC1	1300	16.22	7	AE0035	4-8-2019	1387.31	7.00	7.24	6	2
32	B	AC4	4000	30	3	AE0041	4-8-2019	6006.73	8.50	7.89	7	3
33	B	AC4	290	2.3	2	AE0041	4-8-2019	472.51	8.50	7.89	7	3
34	B	AC4	1280	7.45	5	AE0041	4-8-2019	1562.83	8.50	7.89	7	3
35	B	AC4	319	0.78	1	AE0041	4-8-2019	405.34	8.50	7.89	7	3
36	B	AC4	1799	3.253	5	AE0041	4-8-2019	2249.91	8.50	7.89	7	3
37	B	AB2	910	6.24	52	AE0043	4-8-2019	496.29	8.67	4.00	8	3
43	B	AC7	136	1.1	1	AE0051	4-8-2019	88.74	9.42	4.13	9	3
44	B	AC7	100	0.87	18	AE0051	4-8-2019	65.22	9.42	4.13	9	3
45	B	AC7	300	3	30	AE0051	4-8-2019	323.76	9.42	4.13	9	3
63	B	AC7	1530	4.31	2	AE0051	5-8-2019	931.2	33.42	4.00	10	5
65	B	AD8	2976	34.56	192	AE0077	5-8-2019	2663.99	35.25	7.24	11	5
66	B	AC4	3800	40	4	AE0041	5-8-2019	9784.02	35.50	7.10	12	5
1	B	AA2	44	0.432	6	AE0003	5-8-2019	362.02	5.00	8.14	N	1
2	B	AA2	125	0.74	8	AE0003	4-8-2019	360.69	5.00	5.88	N	1
3	B	AA3	121	0.693	6	AE0005	4-8-2019	223.82	5.00	4.75	N	1
4	B	AA4	109	0.737	1	AE0007	6-8-2019	163.01	5.00	7.64	N	1
5	B	AA5	100	0.48	1	AE0009	4-8-2019	326.16	5.00	7.41	N	1
6	B	AA5	8	0.101	3	AE0009	4-8-2019	89.55	5.00	7.41	N	1
7	B	AA6	840	6.72	10	AE0011	4-8-2019	1355.2	5.00	7.12	N	1
9	B	AA8	63	0.653	1	AE0015	4-8-2019	170.85	5.00	8.21	N	1
10	B	AA9	124	1.306	2	AE0017	5-8-2019	578.16	5.00	7.02	N	1
12	B	AB2	78	0.87	1	AE0021	4-8-2019	71.05	6.42	5.56	N	2
13	B	AB3	1819	2.837	4	AE0021	4-8-2019	959.45	6.42	5.56	N	2
14	B	AB4	160	0.37	1	AE0023	4-8-2019	61.36	6.58	4.53	N	2
16	B	AA5	100	0.11	1	AE0009	4-8-2019	106.72	6.58	7.41	N	2
19	B	AB7	750	1.65	1	AE0029	4-8-2019	466.03	6.83	7.89	N	2
20	B	AB7	796	2.39	3	AE0029	4-8-2019	501.62	6.83	7.89	N	2
21	B	AA8	440	0.62	2	AE0015	4-8-2019	242.61	6.92	8.21	N	2
22	B	AB8	106	0.89	6	AE0031	4-8-2019	85.85	6.92	5.92	N	2
28	B	AC2	172	0.52	1	AE0037	4-8-2019	137.73	7.25	7.48	N	2
29	B	AC3	160	0.29	1	AE0039	4-8-2019	77.29	7.33	8.26	N	2
30	B	AA3	289	0.76	2	AE0005	4-8-2019	181.59	7.58	4.75	N	2
31	B	AA1	82	0.35	1	AE0001	4-8-2019	128.23	7.92	6.66	N	3
38	B	AC5	315	0.97	2	AE0045	4-8-2019	317.19	8.75	4.00	N	3
39	B	AC5	118	0.58	1	AE0045	4-8-2019	142.16	8.75	4.00	N	3
40	B	AB6	350	1.32	2	AE0047	4-8-2019	226.59	9.00	8.50	N	3
41	B	AC6	264	1.39	3	AE0049	4-8-2019	244.18	9.33	8.13	N	3
42	B	AC6	618	1.5	2	AE0049	4-8-2019	521.56	9.33	8.13	N	3
48	B	AA7	295	1.19	1	AE0053	4-8-2019	173.55	11.00	8.50	N	4
49	B	AC2	230	0.96	1	AE0053	4-8-2019	143	11.00	8.50	N	4
50	B	AC8	166	1.63	2	AE0055	4-8-2019	185.33	11.00	7.40	N	4
51	B	AA7	46	0.62	1	AE0053	4-8-2019	91.65	11.00	8.50	N	4
52	B	AA3	805	2.16	4	AE0057	4-8-2019	355.99	11.00	4.00	N	4
53	B	AC9	130	0.72	1	AE0059	4-8-2019	76.33	11.42	4.80	N	4
54	B	AC9	195	0.94	4	AE0059	4-8-2019	108.19	11.42	4.80	N	4
55	B	AD1	464	1.75	2	AE0061	4-8-2019	185.74	11.83	8.50	N	4
56	B	AD1	228	0.23	1	AE0061	4-8-2019	154.42	11.83	8.50	N	4
57	B	AD2	36	0.164	1	AE0063	4-8-2019	270.93	11.92	6.39	N	4
58	B	AD3	255	0.21	1	AE0063	4-8-2019	50.28	11.92	6.39	N	4
59	B	AA8	1210	2.565	2	AE0065	4-8-2019	1226.98	14.58	7.50	N	4
60	B	AD4	102	0.3	3	AE0067	4-8-2019	94.53	15.00	7.50	N	4
61	B	AD5	560	0.768	1	AE0069	4-8-2019	307.3	15.00	8.50	N	4
62	B	AD6	1021	2.7	2	AE0071	4-8-2019	578.69	15.42	7.68	N	4
64	B	AD7	4000	25	2	AE0075	5-8-2019	2368.53	34.00	7.50	N	5
67	B	AD9	830	1.94	1	AE0071	5-8-2019	267.75	39.42	8.50	N	5
68	B	AE1	1635	12.1	54	AE0073	6-8-2019	2228.19	51.50	8.03	N	5
69	B	AE2	225	0.36	1	AE0079	6-8-2019	194.44	54.42	7.38	N	5
70	B	AE3	2800	8.4	1	AE0081	6-8-2019	4886.34	64.25	7.50	N	5

Figure 19: Anonimized example of a loading advice, upper part are T-ULDs

II

Literature Study
previously graded under AE4020

1
Introduction

2020 marked a turbulent year for airlines, as the COVID-19 pandemic resulted in the lowest demand for passenger flights in decades. As cargo is also transported using passenger networks, the total amount of Cargo Tonne Kilometers (CTK's) dropped by 10.6% compared to the year before [34]. The demand for air cargo did not change dramatically however, opposed to the passenger market. This resulted in an increase of 7.7% of the average Load Factor (LF) compared to 2019 [34].

The continuation in air cargo demand could be attributed to the unique advantages of air cargo in comparison to other modalities. According to IATA [35], air cargo accounts for 35% of the global trade in value transported in 2018. While in terms of weight, air cargo only accounts for less than 1% of intercontinental transport in the Netherlands [24]. The main advantage of air transport over other means of transport is that it is a fast and safe way of transport. This makes it particularly interesting for urgent, perishable, valuable and dangerous cargo [12]. Therefore, it is the primary goal of airlines to build a network that is able to transport the cargo as fast and safe as possible across the globe. Delivering the cargo as promised is therefore an important quality indicator to choose for an airline, this will be described in subsection 1.1.1.

Besides airlines, the shipping of cargo by air often involves forwarders. Forwarders usually transport the cargo from a shipper to an airline. The airline ships the cargo across the globe to another station, where a forwarder picks up the package and delivers it to the consignee. There are three types of airlines transporting cargo, the first is a full-freighter airline, which uses aircraft that are only meant for transporting cargo and these airlines do not transport passengers. A second type is a so called integrator, which fulfills both the role of the forwarder and the airline. Examples of these integrators are FedEx and DHL. The third type is a passenger airline that uses free space in passenger aircraft to transport cargo, this is called a combination airline. In addition to a passenger fleet, these airlines sometimes own their own full freighter aircraft as well. An example of such an airline is KLM Cargo, where this case study is for.

1.1. Background

KLM Cargo is a part of Air France KLM Martinair Cargo (AFKLMP), which in turn is the cargo division of the Air France KLM group. The group is the result of a merge in 2004 between the Dutch airline KLM and the French airline Air France. After the merge, both airlines continued to operate using their original colours. The Air France - KLM group operates across 157 countries to a total of 457 destinations from their two main hubs; Amsterdam Airport Schiphol (AMS) and Paris Charles de Gaulle Airport (CDG). KLM mostly operates from AMS and Air France operates from CDG. As this is a case study for KLM Cargo, this case study is centered around the hub in Amsterdam.

The cargo network of KLM uses three means of transport. The first type is a full freighter aircraft, the capacity of this aircraft is fully used for cargo. The second type is the passenger aircraft. As KLM operates a large passenger network, the unused capacity in the lower deck can be used to transport cargo. Aircraft connections for cargo are mostly for intercontinental destinations. For European or continental transport, KLM uses trucks as they can transport more cargo than a narrow body aircraft and are more flexible to use. The three means of transport result in several possible journeys a piece of cargo can take. Airlines want to ensure the quality of the network connections they offer, as quality of service is a motivation for a shipper to choose a certain carrier. There are standards for measuring an airlines operational performance, which will be described in subsection 1.1.1.

1.1.1. Operational performance

On-time performance is one of the major key performance indicators (KPI's) to measure an airline's performance. In order to standardize the operational performance indicators, the Cargo IQ initiative was founded [2], which is an IATA interest group for parties involved in air cargo transportations. A large group of major cargo carriers and forwarders joined this initiative with the goal of creating and implementing quality standards for the worldwide air cargo industry.

Cargo IQ designed a route-map for the complete journey of a piece of cargo, from shipper to consignee. In this route-map, a number of milestones were distinguished and airlines and forwarders are measured on their performance at each of these milestones. In Figure 1.1 a few of the milestones relevant to airlines are shown. The figure starts with a few forwarder process steps, where a specific origin and destination combination is booked (BKD), after this a Freight Way Bill (FWB) is made before delivering it to the airline. A forwarder has until the Latest Acceptance Time (LAT) to submit a shipment at the airline's warehouse.

From this point, the milestones are for carriers. For an airline, the collection of freight delivered in the warehouse is called Freight on Hand (FOH). At a certain point shipments receive the "Received Cargo from Shipper" label, at this point the baseline commitment of an airline is made. In this baseline commitment, all expected times and locations of the future milestones in the route map are defined. such as the departures (DEP) and Arrivals (ARR) at future stations. In the end, when a notification of delivery (NFD) has been given, this actual NFD milestone is compared to the baseline RCS commitment of the NFD milestone. If this NFD milestone happened at the promised time or earlier, the shipment was *delivered as promised (DAP)*. This DAP is the most important operational quality standard.

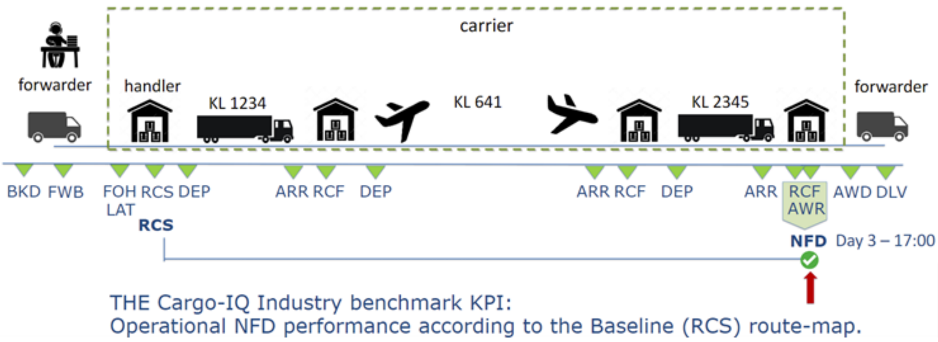

Figure 1.1: The route map as defined by CargoIQ [1]

In addition to the DAP, there are KPI's that are focused on the recovery of cargo that missed a milestone somewhere in the process. An example is the *Flown As Planned (FAP)*, which corresponds to the percentage of cargo that is flown on the flight it was planned. The FAP can be used to determine where bottlenecks are within the whole route map. An extension of this KPI is the DEP-OK, which corresponds to the shipments that were flown as originally planned from the hub to an outstation. A last KPI associated to recovery is the number of *recovery days*, which is the sum of delayed days of all shipments.

Besides *operational performance*, the *contribution margin* is a KPI that could be useful for this research. It is an expression for how much contribution a flight and itinerary have in terms of profit. The contribution margin is calculated by subtracting all variable costs of an itinerary from the generated revenue. This is then divided by the most constraining factor, which is either the weight or the volume. Therefore, the resulting unit is either €$/kg$ or €$/m^3$. An example of variable costs in this contribution margin are the handling fees of an outstation and the hiring cost of a truck driver.

KLM operates their flights using a hub and spoke network, with Amsterdam Airport Schiphol as their hub

(AMS). This means that (almost) every single piece of cargo is shipped through the hub. In AMS, shipments arrive from their origin locations to be connected to destinations all over the world. This shows that the hub plays a key role in the whole process of delivering cargo as promised on their final destination. If the hub fails to connect incoming cargo onto their scheduled flight, the shipments will not be delivered as promised, resulting in a lower operational performance. So the hub has to be able to handle all incoming cargo in order to fulfill the promise to the customer. During peak hours it could happen that there is not enough time to handle all cargo. There are two major measures that could ensure that the hub can handle more efficiently:

1. Increase handling capabilities of the hub, for example schedule more personnel or add extra resources. These kind of measures come with a cost.

2. Organize the incoming cargo in such a way that the workload is minimized and the peak load is reduced.

Selective loading rules were introduced to address this second measure, and will be described in the next section, subsection 1.1.2. It is important to mention that the selective loading rules are designed to be executed at the outstations; the "spokes" of the network.

1.1.2. Selective loading rules

Air cargo is built onto Unit Load Devices (ULDs). There are several types of ULDs, which will be described in chapter 2. When building these ULDs, handlers have to deal with a lot of hard constraints, these are called co-loading rules and include special handling requirements of certain product categories. In addition to these hard constraints, the selective loading rules are introduced at outstations to improve the incoming work at the hub. These are not mandatory to follow, but preferable as it increases the handleability of incoming cargo at the hub. So selective loading rules can be considered as soft constraints. The current selective loading rules are depicted in Figure 1.2.

This image is omitted due to confidentiality reasons

Figure 1.2: The current selective loading rules [15]

In the current selective loading rules, two main types of measures can be observed that improves the composition of incoming ULDs at the hub. The first is to combine cargo with the same destination, whether this is AMS or another destination. For this case, there has to be enough cargo to fill a complete ULD. These types of ULDs are called *Through-ULDs (T-ULDs)* as they simply move directly through the hub to their next flight or truck leg. These T-ULDs are obviously preferred as they do not require any handling at the hub.

If there is not enough cargo for a destination to build a T-ULD, it is preferred that shipments are combined onto a ULD together with packages that have similar connection times at AMS. This is the second type of measure to improve the composition of incoming ULDs. Currently this is done by defining three categories of cargo (as shown in Figure 1.2), which are SHORT, MEDIUM and LONG connections. If the ULDs are combined based on the connection times, the ULDs can be disassembled step-by-step in the hub. Otherwise the hub might have to disassemble them all at once, resulting in a high peak in terms of work load. This could mean that some connecting flights are missed because there is not enough time to handle all incoming cargo. The name of ULDs containing shipments for multiple destinations is a *Mixed-ULD (M-ULD)*.

1.2. Problem Statement

In practice, there are some limitations of the current situation with the fixed selective loading rules. One of these limitations is the large uncertainty in the planning of air cargo. Forwarders do not receive a penalty when a booking does not show up, this is called a no-show. It also happens that more or less cargo is delivered than what was booked, this is called high-show or low-show respectively. Another aspect is that cargo can be submitted up until 6 hours before departure, which is the Latest Acceptance Time (LAT) for KLM. This concerns cargo that is not consolidated onto a ULD yet. Given this, an airline knows just a few hours before departure which cargo will show up in which volumes. Because outstations have limiting capabilities in terms of handling, some cargo will be built up onto ULDs well before this Latest Acceptance Time.

A second complication is that the first task of a local handler is to fill the aircraft as much as possible. Selective loading is preferable for the airline, but for a handler it is still less important than ensuring that all cargo that could be loaded onto the aircraft, is loaded. This sometimes means a small package to destination B is added onto a T-ULD for destination A, which instantly makes it a M-ULD that has to be handled and broken down in the hub.

A third limitation is the use of rules itself to optimize incoming ULDs. The rules have to be interpreted by a handler on the spot when assembling an ULD. This can make it very difficult to follow the rules, especially when realizing an outstation handles multiple airlines, all with their own selective loading rules.

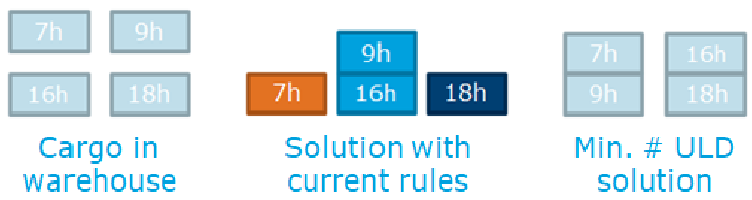

Figure 1.3: Illustration of the limitation of the hard threshold conditions. (Orange=short, Light-blue=medium, Dark-Blue=long)

Furthermore, the current selective loading rules are the same for every singular situation and not really made specific. An example are the harsh threshold conditions for "short", "medium" and "long" connections. This is shown in Figure 1.3. Here a simplified situation consisting of four shipments with the volume of half a ULD is considered. The current rules would ask for three separate ULDs, whereas the logical combination would be to combine shipments onto two ULDs.

A last limitation is that the use of the current fixed rules only optimizes the first leg. It does not consider the shipment characteristics, status of the hub and also the characteristics of the future segment. In a world where a lot of data is available it is possible to make more informed decisions based on actual information of the whole network. Using this data to generate loading advices by a *decision support tool* could result in much more tailored solutions than the use of fixed rules.

These limitations illustrate that there are opportunities to improve the methods of optimizing ULDs at outstations for hub handling. The current limitations can be summarized in the following problem statements:

The current fixed selective loading rules are generic and do not provide sufficient guidance to ground handlers in the building up of ULDs. Therefore, the current selective loading rules are not always fully executed.

The current fixed selective loading are fixed for every scenario and do not consider shipment and network characteristics in determining a loading advice.

Following the problem statements above, it can be suggested that there is an opportunity to replace the fixed loading rules with a decision support tool that generates loading advices based on the received cargo and expected bookings. In these loading advices, the airline can suggest groups of cargo that are preferred to be built together to an outstation. These loading advices would give more guidance to ground handlers in grouping shipments together such that an airline is able to 1) Handle more cargo in the hub and 2) prioritise more profitable cargo.

In practice, there is an issue of the timing of the to be developed model. There are two time points which are key moments in the process of building ULDs. The first moment in time is when the outstation starts building ULDs, which depends on the outstation, but this could be around 12 hours before departure of the truck or flight. However, at this moment in time, cargo can still be delivered to the outstation up until 6 hours before departure. This 6 hours before departure is the latest acceptance time (LAT), which is the second key moment. A two stage loading advice could address this timing issue. In a first stage the number of groups could be decided, along with an initial item to group assignment and an advice to start building a specific group or not. In the second stage the advice could be adjusted based on the updated information.

1.3. Research Objective

The goal of this research is to develop a model that can replace the current selective loading rules with a tool that provides real-time ULD loading advices to outstations. This ULD loading advice is aimed at increasing the operational performance and the average contribution margin of all transport legs. The operational performance could increase by tailoring the incoming work at the hub to the capacity of the hub. The contribution margin could improve by rearranging the work in such a way that the legs with high yield and high load factors are prioritised over legs with low yield and low load factors. This is because the need for ULD optimization is much less for low load factor flight legs than for high load factor flight legs.

The limitations in the problem statement can be divided into three sub-functionalities of the to be designed model. The first is a model that can determine a loading advice based on different parameters across the network. This model should contain logic on how to divide a listed of bookings into groups, which contains both T-ULDs as M-ULDs. A second element is an item-to-ULD assignment model. This model is a simulation tool for the effect of loading advices and should consider outputs like number of ULDs, volume utilization and the ratio between T-ULDs and M-ULDs. A third part which could be added is a framework to cope with the uncertainty in air cargo. Such a framework is called *a stochastic framework*. This could add extra detail to the loading advices by considering forecasts in determining the groups in a loading advice.

So to summarize in one sentence, the research objective is:

Maximize operational performance and contribution margin by providing dynamic ULD loading advices to outstations given shipment characteristics, capacity constraints and flight characteristics.

Where in the required model, three functionalities can be distinguished.

1. A model that can provide loading advices to outstations, based on the properties of the current and future segments.
2. A stochastic framework to incorporate uncertainty in determining a loading advice.
3. A model that performs the assignment of items to a ULD (packing model) to simulate the effect of the loading advices.

There will be difficulties in the evaluation whether the dynamic loading advices will have an impact on the operational performance (DAP). This is because the DAP is dependent on a lot of different factors. Therefore, there are two measurements that are known to have a link with the operational performance:

- *T/M-ratio*: The T/M-ratio corresponds to the ratio between T-ULDs and M-ULDs. In a previous thesis from Veldhuizen (2012) it is shown that more T-ULDs result in a higher hub performance [61]. This is a logical consequence as T-ULDs do not require as much work as M-ULDs. The items on the latter have to be broken down and consolidated onto a new ULD again, whereas T-ULDs can move through the hub into a new aircraft or truck.
- *Peak Workload*: M-ULDs have to be broken down at a certain deadline, this deadline is determined by the shipment with the shortest required handling time. So if a ULD has ten shipments with a ten hour connection time, and one shipment with a six hour connection time, the deadline for handling this specific ULD is six hours. It could be assumed that if the deadlines are more spread out, the workload is more evenly distributed and the peaks are damped. In practice, this could mean that more cargo will be handled on time, resulting in a higher hub performance.

1.4. Research Questions

Using the problem statement and the research objective, a research question can be formulated as follows:

What impact has the use of a real-time dynamic ULD loading advice tool on a cargo airlines operational performance and contribution margin?

A set of subquestions can be formulated to make a distinction between several (sub)problems that have to be addressed in the process.

1. What methods can be used to generate dynamic ULD loading advices for outstations that adjust the (peak) workload at the hub, while considering the future segments of the package journey?
   - What is the effect of dynamic ULD loading advices on the ratio between T-ULDs and M-ULDs?
   - What is the effect of dynamic ULD loading advices on the expected peak workload at the hub?
2. Can a stochastic framework facilitate a two-stage issued ULD loading advice to cope with the uncertainty in the showing up of air cargo?
3. What methods can be used to evaluate the effects of a real-time dynamic loading advice tool?
   - What is the effect of dynamic ULD loading advices on the average space utilization?
   - What is the effect of dynamic ULD loading advices on the average connection spread on a M-ULD?

1.5. Report Structure

In this report, literature that is relevant to the sub-functionalities of the model will be discussed. First in chapter 2, present literature regarding the load planning in air cargo will be discussed, followed by a categorization of problems related to the loading of ULDs and an overview of some packing models is given (Research question 1). Then in chapter 3, an introduction of determining strategies under uncertainty followed by a description of two-stage stochastic programming is given (Research question 2). In chapter 4 three groups of machine learning techniques that could be relevant to determine the ULD loading advice are discussed (Research question 3). The closing remarks and conclusions are provided in chapter 5.

2

Air Cargo Load Planning

In air cargo transport Unit Load Devices (ULD) are used, this is done to ease the handling process as it is more easy to handle groups of cargo than bulk cargo. There are several types of ULDs, but for KLM Cargo three types of ULDs can be considered. These types are the Lower Deck Container (LDC), the Lower Deck Pallet (LDP) and the Main Deck Pallet. The LDC and LDP are shown in Figure 2.1 and Figure 2.2, respectively. The Main Deck Pallet has the same width and length dimensions as the LDP, but has a larger height of 300 cm. This is, as the name suggests, because of the fact that the Main Deck Pallet is for the main deck, instead of the lower deck. Because of the phasing out of the Combi Boeing 747-400, KLM only has three aircraft left that transport cargo on the main deck. Therefore if a pallet is built, it is almost always built as a LDP.

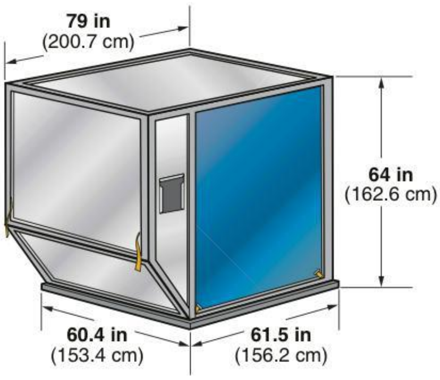

Figure 2.1: The Lower Deck Container (LDC), type LD-3

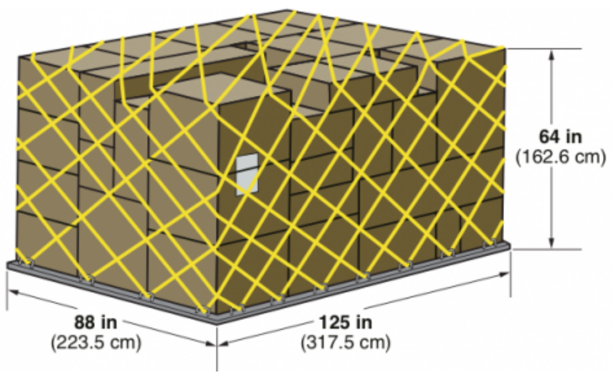

Figure 2.2: The Lower Deck Pallet (LDC), type LD-7

Figure 2.3: Images from Searates [55]

For every flight, the composition of ULD types can be different due to several reasons. Firstly, the available space is aircraft dependent. As KLM operates six different wide-body aircraft for long haul destinations, the belly capacity can differ from day to day. Furthermore, because of passenger operations, a part of the belly is used to transport passenger luggage, resulting in less space for cargo. As part of a previous thesis at TU Delft, a forecasting tool was made for the available space [58]. A third reason for a different composition is that a choice can be made in the configuration of ULDs itself. An example can be found in Figure 2.4, where different combinations for a Boeing 787-10 are shown. This aircraft type is one of the aircraft used by KLM for it's intercontinental destinations.

In Europe, trucks are used to transport cargo from the outstation to Amsterdam. There are four pallet positions (Lower-deck or Main-deck pallet) in one truck, or eight LDC positions. So for trucks it can be stated that one pallet position equals two container positions. Opposed to flight legs, the truck leg capacity can be scaled up due to expected demands, which is currently done by the regional planning department.

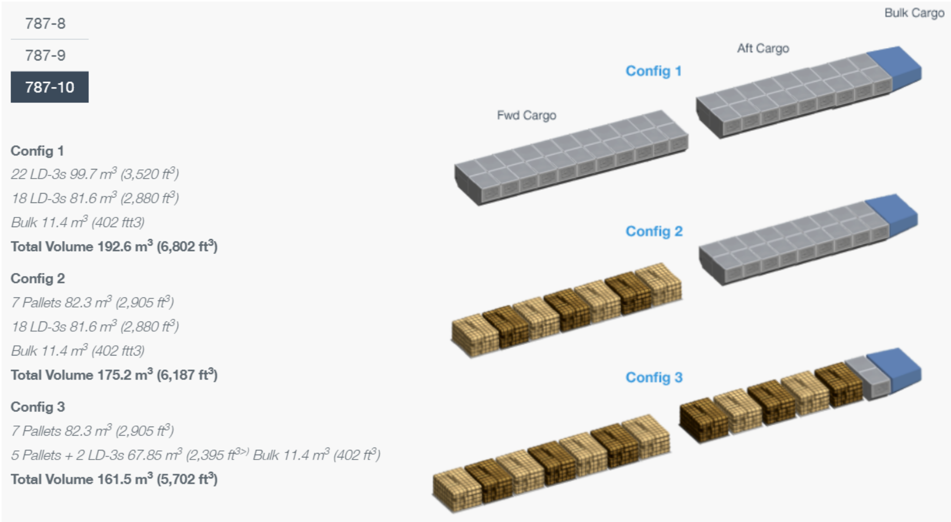

Figure 2.4: Different ULD combinations for a Boeing 787-10 [10]

2.1. Air Cargo Load Planning Problem

There are several choices to be made while planning the shipping of air cargo. Brandt and Nickel (2019) published a comprehensive overview of the problems to be addressed in the air cargo operation from booking to flight [12]. Brandt and Nickel also formulated a group name of all problems to be solved, this is the Air Cargo Load Planning Problem (ACLPP). A set of four subproblems that form the ACLPP were identified in [12]. These respective subproblems and the decision they tackle are:

1. **Aircraft Configuration Problem (ACP)**: The types and number of ULDs to be selected for a flight.
2. **Build-up Scheduling Problem (BSP)**: At what time at what workstation a ULD should be built.
3. **Air Cargo Palletization Problem (APP)**: Assign specific items to a ULD and decide their placement onto a specific ULD.
4. **Weight-and Balance Problem (WBP)**: Assign the ULDs to a loading position in the aircraft.

As the selective loading rules involve the assignment of items to a specific ULD, the Air Cargo Palletization Problem (APP) is the subproblem that will be directly affected by the rules. The APP has been researched extensively before. The general term in literature for these kind of problems is container loading problems. Besides the air cargo industry, there are many more industries that have to deal with these container loading problems. [11] provided an extensive review of 160 papers addressing Container Loading Problems in 2013. According to [11], these kind of problems have either one of two objectives; minimize the used number of bins (containers) or maximize the utilization of a given set of bins. [12] distinguished elements of both this first and second objective in the ACLPP, because for handling requirements the number of used ULDs should be minimized, but in case of an exceedance of aircraft capacity, the available space in the given set of ULDs should be maximized.

2.2. Air Cargo Palletization Problem (APP)

There are a lot of constraints to be included that affects the objective of the APP of minimizing the unused space, minimizing the required number of bins or maximizing the utilization of a given set of bins. Brandt and Nickel provided the following overview in [12].

ULD assignment: Every item should be assigned to a ULD or marked as "not loaded".

Item rotation: Most items are rectangular and therefore only orthogonal rotations should be considered. Sometimes rotation is not possible due to heavy or fragile items.

Item position inside the ULD / the aircraft fuselage: A loaded item must be positioned fully inside the admissible contour of its assigned ULD. In practice some items are larger than a ULD. These are either loaded on one ULD with overhang and some space must be left out on the adjacent ULD or the entire adjacent loading position is left free.

Item availability: Each item can only be assigned to a ULD whose build-up period is at least the period at which the item becomes available for build-up.

ULD weight limit: Each ULD has a maximum allowed weight after loading.

Net weight limits: Some items may contain substances (dry ice, radiation, chemicals, explosives) whose total sum needs to be limited within a subset of ULDs.

ULD weight distribution: Items should be loaded in a way such that the center of gravity of each ULD is centric and low.

Item stacking: When loading multiple items into the same ULD some items might have to be stacked on top of others. As most items carry significant weight and the load-bearing strength of each item is limited, we need to restrict the stacking accordingly.

Complete shipment: Shipments may consist of multiple items that can in theory be shipped individually. But, in some countries customs does not permit the arrival of partial shipments.

Item grouping: Certain sets of items, like items of the same shipment, should be split across as few ULDs as possible. This decreases the risk of disruptions and reduces handling effort. Furthermore, item grouping can also be seen from a ULD perspective where as little as possible different sets of items, say items with the same final destination, should be mixed on a ULD.

Item compatibility: Besides the soft item grouping constraints, there are also hard constraints on items that must not be placed on the same ULD or in close proximity.

Item positioning: For some items the allowed position inside the aircraft or inside the assigned ULD needs to be restricted. Similarly, the relative position of an item might need to be constrained. For example, to place it only on the periphery of the ULD for easy accessibility.

Stability: We distinguish between vertical and horizontal stability. Vertical stability prevents the items from falling down or tipping. Horizontal stability prevents the shifting of items when the ULD is moved or tilted.

The palletization problem can be modelled with the use of so called packing models. There are two main branches of packing models, which will be described in the following sections. The types that are described are both bin packing problems and Knapsack problems in section 2.3 and section 2.4, respectively. Referring to the two objectives of these models distinguished in [12], the Bin packing problem has the objective to minimize the number of required bins (ULDs). The knapsack problem has the objective to maximize the utilization of a given set of bins.

2.3. Bin Packing Problems

Bin Packing problems have the objective to minimize the number of required bins to pack all items or to minimize the unused space. In literature, there are already models that solve instances for one-dimensional (1DBPP), two-dimensional (2DBPP) and three-dimensional (3DBPP) problems. A basic 1D formulation is displayed in Equation 2.1 [42], which involves a column generation algorithm of Gilmore and Gomory for determining the set of feasible bins [28]. In this equation, the set of feasible bins is represented by the matrix A.

$$
\begin{aligned}
\min \sum_{j=1}^{M} x_j & \\
\text{subject to} & \\
\sum_{j=1}^{M} a_{ij} x_j = 1 & \quad (i = 1, \ldots, n) \\
x_j \in \{0, 1\} & \quad (j = 1, \ldots, M)
\end{aligned}
\tag{2.1}
$$

In Equation 2.1, a_{ij} is part of a column vector A_j, where a_{ij} takes a value of 1 if item i belongs to bin j, and 0 if item i does not belong to bin j. The set of all feasible bins is represented by the matrix A, which consists of all possible columns A_j. The value of x_j equals 1 if the bin j is included in the final solution and is 0 otherwise. In the end it is the goal to minimize the number of bins.

The 1DBPP considers a maximum weight to be loaded in a bin. If there is no possibility of adding another item into the bin, a new bin is 'opened'. A two-dimensional bin packing problem (2DBPP) considers the width w_j and height h_j of a set of n rectangular items to be loaded into a bin with a width W and height H [42]. When extending the 2DBPP to a three-dimensional bin packing problem (3DBPP), the depth d_j is added to the set of n rectangular items and for the bin, the depth D is added [45].

The bin packing problems can be categorised further, based on the item and bin characteristics. Wäscher et al. (2007) defined a typology of Cutting and Packing problems (C&P) [62], which was an improvement of the initial typology proposed by Dyckhoff (1990) [22]. One of the elements of this typology is the assortment of items, which is split into three classes in [62]; identical, weakly heterogeneous and strongly heterogeneous set of items. Weakly heterogeneous set of items indicate that the items can be grouped in a few classes, whereas strongly heterogeneous means that the items consists of many classes and identical set of items means that all items are the same. Brandt and Nickel (2019) identified that air cargo items are strongly heterogeneous, because of the very diverse dimensions and shapes, but that the bins, ULDs in this case, only consists of a few shapes and therefore can be considered weakly heterogeneous [12]. This makes the bin packing problem for air cargo a Multiple Bin Size Bin Packing Problem (MBSBPP).

2.4. Knapsack Problem

Knapsack problems are focused upon maximizing the value of items put into a given set of bins (knapsacks). These knapsack problems exist in two main forms. The first is the 0-1 knapsack problem, where an item is either fully loaded or fully unloaded into the bin. A second type is the fractional knapsack problem, where it is possible to partially load an item. From the view of an air cargo palletization application, it would not be possible to cut shipments to fulfill maximum space utilization and therefore a 0-1 knapsack problem would be most applicable.

In 2004, Fréville provided an overview of the Multi-dimensional 0-1 Knapsack Problem (MKP) [26]. This term was first introduced by Weingartner et al. [64]. A basic formulation of the MKP is shown in Equation 2.2.

$$
\begin{aligned}
\max \sum_{j=1}^{n} c_j x_j & \\
\text{subject to} & \\
\sum_{j=1}^{n} a_{ij} x_j \leq b_i, & \quad i \in M = \{1,2,\dots,m\} \\
x_j \in \{0,1\}, & \quad j \in N = \{1,2,\dots,n\}
\end{aligned}
\tag{2.2}
$$

In Equation 2.2, n is the number of items and m is the number of knapsack constraints with capacities b_i. Each item requires a_{ij} units of resources in the ith knapsack. The item yields c_j units of profit when it is included in the knapsack. The knapsack problem has already been solved as one-dimensional, two-dimensional and three-dimensional problem. According to the typology of Wäscher et al. (2007), the knapsack problem for air cargo falls in the category of Multiple Heterogeneous Knapsack Problems (MHKP) [62].

2.5. Packing problems for air cargo operations

The most recent developments in literature solving the Air Cargo Palletization Problem have been aimed at solving the three-dimensional MBSBPP (3D-MBSBPP). Brandt and Nickel (2019) provided an overview of papers addressing the 3D-MBSBPP for air cargo palletization. Generally speaking there are two solution approaches to solve the problem. These approaches are Mixed Integer Linear Programming (MILP) and a heuristic approach. Sometimes a MILP is also combined with a heuristics method.

2.5.1. Mixed-Integer Linear Programming

Three methods known to solve the 3D-MBSBPP for air cargo with the use of only a MILP model are the methods from Paquay (2016) [48], Wu (2008) [65] and Hong Ha (2016) [32]. The method from Paquay (2016) is solved from the airline's perspective, whereas Wu (2008) solves the 3D-MBSBPP from a forwarder's perspective. Because Wu (2008) solves from a forwarder's perspective, the objective is to minimise the cost associated to renting a ULD from an airline. In Paquay (2016) the objective is to minimize the unused space within a ULD. Paquay (2016) considered a large range of handling constraints, including item orientation, physical stability, weight limits, weight distribution and stacking. This resulted in a total of 65 constraints. For Wu (2008), which considered less constraints there were 18 constraints.

The research of Hong Ha (2016) also took the perspective of a forwarder that has to consider the availability of items and the priority of certain items on a daily basis [32]. The maximum difference in delivery time between items on a single ULD was set to two days, meaning that for example an item delivered to the forwarder on friday should not be combined with an item delivered on the monday before. Additionally, priority levels were assigned to items to distinguish items based on priority, which is similar to the objective of this research. However, the objective of this research is to optimize for hub handling, where priority levels can be used as a basis to make decisions about how to arrange items onto ULDs.

All MILP models were only able to solve small instances within a reasonable amount of time. For Paquay this means instances with one ULD and 12 boxes within one hour [48], for Hong Ha (2016) problems with two ULDs and 15 boxes were solved in 20 minutes. This is not realistic in a practical situation where multiple ULDs for a few hundred boxes are used. The 3D-MBSBPP is also known to be a NP-hard problem, which makes that the number of items to be packed strongly influences the complexity of the model [12].

2.5.2. Heuristic approaches

Heuristics are known as suitable approaches for complex problems. Even though heuristics do not always provide an exact and/or optimal solution, they are able to solve large instances within a reasonable amount of time. Sometimes heuristics are also used in combination with a MILP model, these combinations will also be discussed in this section.

Two of the most recent heuristic methods are from the same authors of the MILP-model from the previous section. Paquay et al. (2018) formulated a MILP-based constructive heuristics method that is able to solve instances with 80 items between 5 and 35 minutes [50]. Here they solve the same problem as in the paper from Paquay (2016) [48]. An improvement on this was a fast two phase constructive heuristic from Paquay et al. (2018) [49], which solved instances with 100 items for multiple bins within 12 seconds. However, the obtained average volume utilization was around 50%, which is far below what airlines are able to achieve in 2019, according to Brandt and Nickel [12].

The objective of this research is to arrange shipments onto ULDs in such a way that the (peak) workload is minimized. The research which comes closest to this goal is the research of Chan (2006) [17]. The proposed model of Chan (2006) is the only model which considered grouping constraints, according to Brandt and Nickel (2019) [12]. In this model, a two phase method is used, where the first phase is modelled with the use of a Linear Programming model and the second phase is modelled with a heuristic algorithm. In the first phase the ULDs to be selected for loading are determined, and only liquid volumes of shipments are considered. The second phase aims to divide the shipments across ULDs to maximize the load of all ULDs. The model was able to place 20 to 100 boxes onto a ULD within one minute while achieving a volume utilization around 90% [17]. In this model a constraint is added to avoid that items belonging to same shipments are assigned to multiple ULDs. However, opposed to the goal of this research, the grouping of multiple shipments to ease the hub handling process are not considered in the research of Chan (2006).

2.5.3. Applicability for this research

Even though a lot of research has been done on the field of packing models for the air cargo palletization problem, to our knowledge there is not a model yet developed that could address the problem presented in chapter 1. The objective of packing models is to maximize the space utilization or to minimize the required number of bins. In previous research, a lot of constraints have been added to packing models. However, there is not a model to our knowledge that also considers to pack items together based on the items characteristics. This means that packing models that build ULDs with the same destination (T-ULD) or with similar connection times to their next segment (M-ULD) have not been researched or described in literature before.

However, as the output of a packing model would be a detailed item-to-ULD list, the result would be too specific for an outstation. Due to the irregular arrival times of cargo and also the unpredictability of size, shape and volume, less specific solutions are more preferable for an outstation. The desired outcome of the model should be a rough list in which the to be shipped cargo is divided into several groups of T-ULDs and

M-ULDs. In order to find a model that can produce this outcome, Machine Learning techniques are also researched for this research and described in chapter 4.

This does not mean however, that a packing model is not useful for this research. A packing model could be used as a tool to simulate the effects of a loading advice. The output of the packing model can be converted to relevant outputs to the operational performance, such as the percentage of T-ULDs or the distribution of M-ULD connection times. Due to the unavailability and inaccuracy of shipment dimension data, a two-dimensional or three-dimensional packing model would not add extra detail to the simulation in comparison to a one dimensional BPP. The to be designed BPP should have extra functionalities compared to the literature, such as the ability to create T-ULDs and M-ULDs besides the usual objective to minimize the required number of bins.

3

Strategies under Uncertainty

This chapter discusses present literature that describes the evaluation of strategic discussions under uncertainty. In this research, large uncertainties come due to the phenomenon of no-show, low-show, high-show and late-show. A booking list is not representative as it is still not sure whether the booked cargo will show up on time, in the booked volumes or if it even shows up at all. This does not only illustrate the potential for forecasting in the air cargo industry, but it also raises the question on how to act on those forecasts. Courtney, Kirkland and Viguerie wrote a business review on several scenarios that occur in industries [18]. In this review, four levels of uncertainty are identified:

- **Level 1: A clear enough future.** In this scenario, a single forecast is precise enough in order to determine a strategy. Although it is subject to a certain business' uncertainties, the forecast is good enough to provide a clear strategy.
- **Level 2: Alternate Futures.** For Level 2, the futures are discretized in a few scenarios or alternate values. In this case, it is still uncertain which outcome will occur, but probabilities can be incorporated to determine the optimal strategy under the uncertainty.
- **Level 3: A range of futures.** In Level 3, a range of futures can be identified. This range is bounded, but the future lies anywhere between these two bounds in a continuous space.
- **Level 4: True ambiguity.** At Level 4, multiple dimensions of uncertainty interact to create an environment that is impossible to predict.

For Level 1 uncertainty problems with one single forecast, the problem is deterministic and can be solved using the single forecast as input. In the case of level 2 problems a stochastic framework can be implemented to determine an optimal strategy under multiple scenarios. Continuous problems (Level 3) can be discretized to multiple scenarios and then solved with a stochastic framework aswell. In the review of Courtney (1997) it is stated that at least half of these problems belong to level 2 or level 3, while most of the other problems are level 1 [18], which makes level 4 problems a rare occasion.

There are stochastic frameworks available that can determine strategies in situations where two to multiple successive decisions have to be made. The general term for this is stochastic programming [56]. In this research, a two stage problem is considered which makes two stage stochastic programming a suitable option to deal with the uncertainties while determining a loading advice.

3.1. Two-stage Stochastic programming

A two stage stochastic framework could be a suitable method to determine an optimal strategy under the uncertainties in air cargo. In such a framework, separate decisions have to be made at each stage but they are affected by each other. In this research it would involve two moments in time, which are the moment an outstation starts building ULDs and at the latest time cargo can be delivered (LAT). At the first stage, the set of cargo clusters, both T-ULDs and M-ULDs should be determined. This should be done while taking the forecasts for high show, low-show and no-show into account. At the second stage, all shipments are known and at this point the remaining shipments have to be assigned to one of the pre defined clusters. Here, some alterations could also be made with respect to the initial advice for all cargo that is not consolidated on a ULD yet.

In Shapiro et al. (2014), a basic definition of a (linear) two stage stochastic programming model is given as in Equation 3.1 and Equation 3.2 [56]. This basic two stage stochastic programming model originates from Dantzig (1955) [19] and Beale (1955) [5]. The first stage problem is

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} c^\top x + \mathbb{E}[Q(x,\xi)] \\ &\text{s.t.} Ax = b, x \geq 0 \end{aligned} \tag{3.1}$$

where $Q(x,\xi)$ is the optimal value of the second stage problem (Equation 3.2).

$$\begin{aligned} &\min_{y\in\mathbb{R}^m} q^\top y \\ &\text{s.t. } Tx + Wy = h, y \geq 0. \end{aligned} \tag{3.2}$$

The element ξ in Equation 3.1 consists of the data of the second stage problem, so $\xi := (q,h,T,W)$. Some or all elements of vector ξ are random and the expectation operator of the first stage problem is taken with respect to the probability distribution of ξ. The problem concerning the vector ξ is called the *recourse problem*. In [56], there are also generalized forms for more than only linear problems as well as multi-stage stochastic problems that consist of more than two stochastic stages. Once the model is formulated, there are multiple methods to solve the problem. In this literature study, a few methods are given, which can be split into exact solution methods and approximation methods. A complete overview can be found in Birge (2011)[9].

3.1.1. Exact solution methods

The first known method to solve a two-stage stochastic problem is the L-shaped method, which is an exact method and proposed by Van Slyke and Wets (1969) [60]. The approach behind the L-shaped method is to only consider a finite number K of possible realizations to approximate for the random vector ξ. This is more relevant for continuous scenarios, as for discrete scenarios a finite number K of possible realizations is already given. This is done to avoid a large number of evaluations of the nonlinear term in the objective, which is called the recourse function. The assumption made of only considering a finite number of K possible realizations makes it possible to write the problem as a deterministic linear problem. This problem can be defined as the extensive form:

$$\min_{x\in\mathbb{R}^n} c^\top x + \sum_{k=1}^{K} p_k q_k^\top y_k \tag{3.3a}$$
$$\text{s.t.} Ax = b, \tag{3.3b}$$
$$T_k x + W y_k = h_k, \;\; k = 1,\ldots,K; \tag{3.3c}$$
$$x \geq 0, y_k \geq 0, \qquad k = 1,\ldots,K. \tag{3.3d}$$

Here p_k corresponds to a probability that realization k happens. The decision variable y_k is the second-stage decision for realization k, with the corresponding values of q_k, T_k and h_k. This equation is then solved in three steps (excluding initialization). In the initialization three parameters are set to 0, which are the number of feasibility cuts r, the number of optimality cuts s and the number of iterations v. Below, the three steps are explained.

1. Set $v = v+1$. Solve the following LP problem:

$$\min_{x\in\mathbb{R}^n} c^\top x + \theta \tag{3.4a}$$
$$\text{s.t.} Ax = b, \tag{3.4b}$$
$$D_\ell x \geq d_\ell, \quad \ell = 1,\ldots,r; \tag{3.4c}$$
$$E_\ell x + \theta \geq e_\ell, \ell = 1,\ldots,s. \tag{3.4d}$$
$$x \geq 0 \tag{3.4e}$$

After solving this, the optimal solution obtained has the form (x^{ν}, θ^{ν}). The problem in Equation 3.4 is called the *master problem*. In this problem, constraints introduced by Equation 3.4c are called *feasibility cuts*, which correspond to the boundaries of feasibility. Constraints given by Equation 3.4d are called the *optimality cuts*, which as the name suggests, correspond to the linear approximations of optimality.

2. Check if x is possible in the set of possible realizations K in the second stage. If not, another feasibility cut is needed so return to step 1, otherwise, go to step 3.

3. For all possible realizations $k = 1, \ldots, K$ the following Linear Program should be solved:

$$\min q_k^\top y \tag{3.5a}$$
$$\text{s.t.} Wy = h_k - T_k^{\nu}, \tag{3.5b}$$
$$y \geq 0. \tag{3.5c}$$

The resulting simplex multipliers associated to the optimal solution of realization k for Equation 3.5 are denoted as π_k^{ν}. Then

$$E_{s+1} = \sum_{k+1}^{K} p_k * (\pi_k^{\nu})^\top T_k \tag{3.6}$$

and

$$e_{s+1} = \sum_{k+1}^{K} p_k * (\pi_k^{\nu})^\top h_k. \tag{3.7}$$

Let $w^{\nu} = e_{s+1} - E_{s+1} x^{\nu}$. If $\theta^{\nu} \geq w^{\nu}$, stop; x^{ν} is an optimal solution. Otherwise set $s = s+1$, add to the optimality cut and return to step 1.

A downside of the L-shaped method is that the efficiency is heavily dependent on the starting point of the algorithm [9]. Therefore additional exact methods have been developed that use the L-shaped method as basis. Examples are Regularized Decomposition, a Piecewise Quadratic Form and Bunching. For a complete overview we refer to the book of Birge (2011) [9].

3.1.2. Approximation methods

In the exact formulation the probability of satisfying a set of constraints can quickly make the problem very complicated. In these cases an exact approach with a Linear Programming formulation will result in large computation times, similarly to the MILP bin packing problems from chapter 2. These computational problems can be solved using approximation methods. However, this usually results in finding a good solution, but not an optimal solution. In the book of Birge (2011), a distinction is made between two branches of approximation methods. The first are methods that approximate the expectation term in Equation 3.1 [9] to simplify the computation of future realizations. The main idea behind this process is to take a subset of possible realizations that is very representative for the complete set of possible realizations.

The second branch are Monte-Carlo methods. Here, Monte-Carlo estimates are used to replace the recourse function where random samples are taken from the random vector ξ. Monte-Carlo estimates are relatively accurate and able to handle high-dimensional data, which makes them suitable as an approximation method [9]. The solution converges to an optimum as more samples are taken. There are several Monte-Carlo methods. One of them is the *Sample average approximation*, which is based on the L-shaped method. This method achieves relatively accurate results, but requires a large number of samples. Another well-known Monte-Carlo method is the *Stochastic decomposition method*. This method is proposed by Higle and Sen (1991) [31]. In their approach a small set of samples is taken initially, instead of a complete set. This set is slowly increased until it converges. Generally speaking, the solution becomes more accurate as more samples are taken. This is the case for all Monte-Carlo methods.

3.2. Applied case studies

In applied literature, Wu (2011) proposed a two stage stochastic recourse model that deals with uncertainties for forwarders in air cargo [66]. Forwarders deal with similar uncertainties in the shipping and booking of cargo as airlines. This makes an interesting case study for this research. The forwarders consolidate shipments from multiple shippers, which gives them discounted rates from airlines. Typically, forwarders book their cargo at the airline approximately 1 week before departure. At this point however, it is not certain yet how much cargo will be booked by shippers at the forwarder. Wu (2011) identified the first stage as the moment when a certain amount of cargo should be booked in advance, with uncertain information at the moment in time. The second stage happens on the day of the booking, when all bookings are known. There are additional costs associated to cancelling a booked ULD or to book an additional ULD. The objective of the model is to book the optimal number of ULDs at the first stage, given the possible outcomes of the second stage. This stochastic recourse model was formulated as a Mixed-Integer Programming (MIP) model. For the formulation, we refer to [66]. A case study was performed for a forwarder based in Hong Kong with a historic booking from Hong Kong to New York, involving 16 bookings from shippers. The model was able to realize a 18% improvement compared to the managers' decision in 0.27s.

Ang et al. (2009) developed a two-stage stochastic model to determine the optimal cargo mix in a multi-period planning horizon for sea cargo [4]. The objective here is to maximize the total expected profit derived from all freight bookings received in the planning horizon.

Delgado et al. (2019) formulated a multi-stage stochastic programming model to allocate cargo over a passenger belly network in order to maximize profit [20]. In this case study multiple flight legs with demand and capacity uncertainties were considered and therefore a two-stage stochastic programming model would not suffice. As was mentioned by Delgado et al. (2019), multi-stage problems are difficult to solve, whereas for two-stage problems efficient algorithms exist that are already widely used.

However, to our knowledge there is no previous research where a machine learning technique is used within a stochastic framework. Even though this does not necessarily mean it is not possible, it would certainly be a challenge. As this research is primarily about the effect a decision support tool could have on the operation of a cargo airline, the focus should be on the decision support tool first. This tool can later be extended with frameworks that can take uncertainties and forecasts into account.

4

Machine Learning Techniques

All the present packing models do not optimize the ULDs for hub handling, to our knowledge. Machine learning techniques could be suited to generate loading advices. In this chapter, three machine learning techiques are discussed that could be relevant. The first technique is clustering (section 4.1). This method is selected because it is able to distinguish clusters in data without training. This could be applied to the characteristics of shipments for cargo airlines, to identify good clusters of shipments to group items together in especially M-ULDs. Mixed ULDs are currently clustered based on connection times. This clustering is performed using simple rules and threshold conditions, but with the use of clustering algorithms these thresholds can be made more fluid and tailored to the actual set of bookings. Besides this, more factors can be considered in clustering items together, such as shipment priority or value.

The second method in section 4.2 is Random Forest (RF), which is a intuitive technique to solve classification and regression problems. After a good training it is able to classify different data entries based on it's characteristics and it could therefore be a promising technique for dividing multiple shipments in different classes. RF is a relatively intuitive method and is known to perform reasonably well after a good training. It could therefore be used as a good classifier for cargo, where it could decide whether or not to build a T-ULD for a specific flight and to assign cargo to M-ULD groups.

The third and last technique is Reinforcement Learning (RL), which is described in section 4.3. RL is a method where an agent is trained using a reward and/or penalty system based on the quality of it's actions. Historic data is used to develop a policy of how to act upon certain situations. Therefore, the training of a RL method for the generation of loading advices could be executed by using the effect of the advices as a feedback mechanism, which could make it a suitable application for this problem. In this way the model would develop a policy itself based on a range of input parameters. Once again, this policy would involve decisions like whether or not to build a T-ULD and also on how to cluster the M-ULD groups.

4.1. Clustering Algorithms

The classification of objects, cargo in this case, has been researched extensively before. According to Saxena et al. (2017) [54], the purpose of the study of classification of objects is to develop a tool or algorithm that can predict the class of an unknown object. Because the classes itself are initially unknown, a clustering algorithm can be categorised as unsupervised learning. In Maimon and Rokach (2005) [44], the clustering structure is formally represented as a set of subsets $C = C_1, \ldots.., C_k$ of S, such that $S = \bigcup_{i=1}^{k} C_i$ and $C_i \cap C_j = \emptyset$ for $i \neq j$. This means that any element of S belongs to exactly one subset. Clustering algorithms can be divided into two main branches: Hierarchical clustering and Partitional clustering. In Figure 4.1 the taxonomy of clustering algorithms is given.

4.1.1. Hierarchical Clustering

Hierarchical Clustering is based on the formation of dendrograms [54]. A dendrogram is a diagram that represents the hierarchical relationships between clusters. Within hierarchical clustering, there are two forms of clustering: agglomerative and divisive clustering [46]. Divisive follows a top-down approach and agglomer-

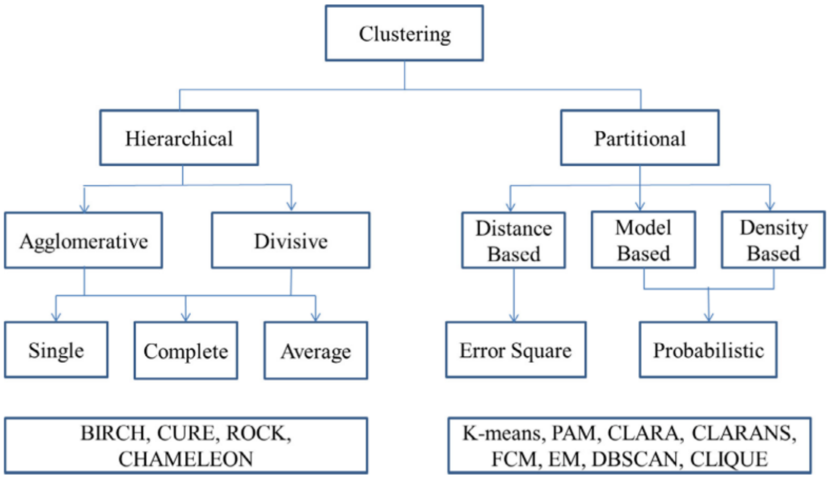

Figure 4.1: Taxonomy of clustering approaches [25]

ative follows a bottom-up approach. For agglomerative clustering, every data point has it's own cluster. The two clusters that are closest together are then merged up until there is only one cluster left. In divisive clustering the complete dataset is taken as one cluster, and then the cluster is split based on the two data points that have the largest distance between them. The distance between two clusters can be measured by *single-linkage clustering*, *complete-linkage clustering* and *average-linkage clustering*. For the difference between the two we refer to Saxena et al. (2017) [54].

An example of a dendrogram is given in Figure 4.2. Here, European countries are clustered based on their quality of life by muk (2015) [70]. According to this graph, for the Netherlands' point of view, the United Kingdom is the most similar based on their quality of life when looking from a botton-up perspective. When looking top-down the Netherlands belongs to one of two main clusters together with for example Switzerland, whereas Hungary belongs to a different cluster. At a certain point the number of clusters to be taken can be determined and to do that a line can be drawn for the maximum linkage distance. The number of clustering lines it crosses indicates the number of clusters. In Figure 4.2 this line is drawn vertically at 50, where it crosses 3 horizontal lines, indicating a total of 3 clusters.

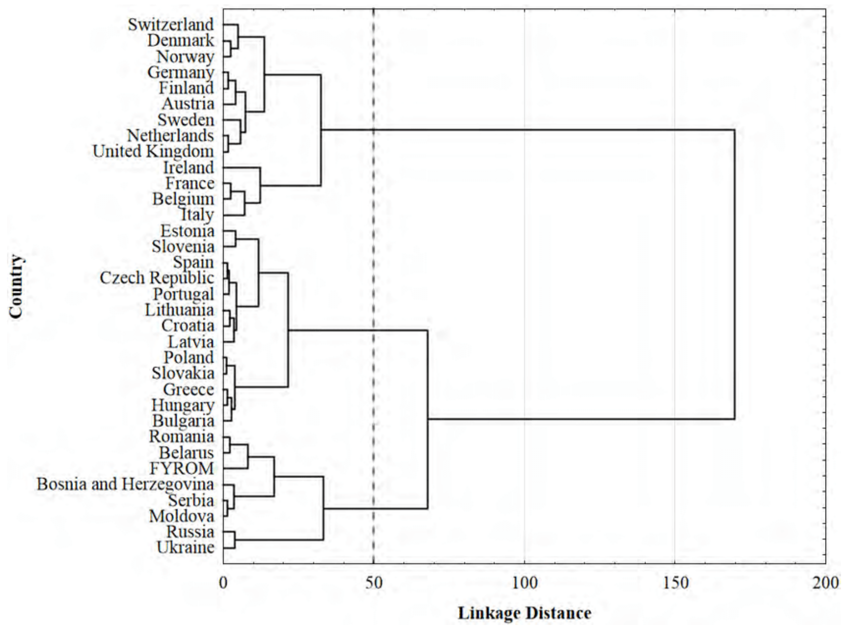

Figure 4.2: Example of a dendrogram with countries clustered based on quality of life by muk (2015) [70]

The classic criticism on hierarchical clustering methods is that they are sensitive to noise and outliers and that they are therefore not robust [54]. If a data entry is assigned to a certain cluster a hierarchical clustering method is not able to reassign it if it turns out another cluster would be a better fit. Because of the relatively high computational complexity, it is also not able to cope with large datasets [54]. However, there are some modern versions of hierarchical clustering methods that deal with some of the limitations of classic hierarchical clustering. Examples of these new hierarchical clustering algorithms are BIRCH, CURE, ROCK and Chameleon. A description is given below.

- *Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)* [69]: This algorithm uses so called cluster features to store the information. Therefore the data stored in the memory can be relatively lower which makes this algorithm less computationally complex. Besides, BIRCH is able to cope with large datasets.
- *Clustering Using Representatives (CURE)* [29]: CURE is more robust to outliers aswell as BIRCH, but CURE is even better compared to BIRCH. Besides, it is able to handle large scale data. However, it is more computational complex than BIRCH.
- *ROCK* [30]: This agglomerative algorithm is suitable for categorical data, opposed to BIRCH and CURE, which are suitable for numerical data. It is based on the number of links between two data records, which represents the number of records that are similar to each other.
- *Chameleon* [37]: Chameleon clusters based on the connectivity between two clusters compared to the internal connectivity of a single cluster. So if the distance between two clusters is smaller than the internal distance within the cluster, a merge between the two clusters occurs. Chameleon is not able to handle high dimensional data [54].

4.1.2. Partitional Clustering

In Partitional Clustering, data is clustered in k clusters without any hierarchical structure by optimizing some criterion function [40]. Some of these criteria are distance and density, where Euclidean distance is the most used criterion [54]. Examples of distance-based partitional clustering methods are k-means, PAM, CLARA, CLARANS and FCM. An example of a density-based partitional clustering method is DBSCAN.

The most basic and well known partitional clustering method is *k-means clustering*, which was initially developed by Macqueen in 1967 [43]. In k-means clustering a number of k required clusters has to be defined in advance. The goal is to find k centroids that define the center of a cluster, so for each cluster there is one centroid. The centroid can be seen as the average position of all data points belonging to that cluster. The objective function J is given as in Equation 4.1.

$$\min J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^j - c_j\|^2 \tag{4.1}$$

Here, $\|x_i^j - c_j\|^2$ represents the distance between a data point x_i and one of the current centroids c_j. Saxena et al. (2017) provided a basic and intuitive flowchart of how the k-means clustering method works [54], which is shown in Figure 4.3.

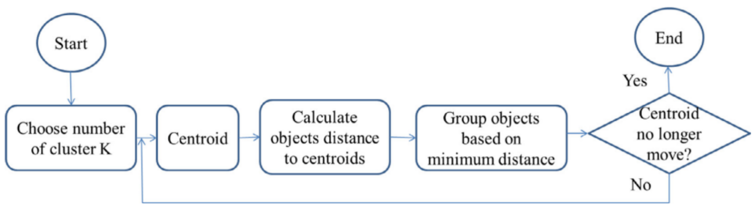

Figure 4.3: Flowchart of the k-means algorithm [54].

In practice, it occurs that the number of required clusters k is not defined yet in advance or that there is a need to find an optimal number of clusters k. This can be done with the use of the elbow method, which is described by Bholowalia and Kumar (2014) [8]. In the elbow method, the percentage of variance is plotted against the number of clusters k. Initially, when starting from k=1, the slope of the graph will be relatively steep until it reaches a point where the slope will flatten. This is the point where an extra cluster will not add significant improvements in accuracy. The resulting plot will look like an elbow, hence the name. At the tip of the elbow, an optimal number of k clusters is reached.

K-means clustering is known to have some limitations, even though it is one of the most popular partitional clustering methods. One of them is that there is not a universal method to identify the initial partition of clusters yet. Another limitation is that k-means is sensitive to noise and outliers. A point can be very far away from any cluster, but it still has to be assigned to one. This can also result in a different centroid which distorts the cluster shape [68]. There are other partitional methods, that can deal with some of the limitations. A few examples are given below.

- *Partitioning Around Medoids (PAM)* [38]: This method uses the same approach for clustering as k-means clustering. However, instead of a centroid a so called medoid is taken. The medoid is an actual data point that represents the cluster's centre the most. Formally, it is the data point with the smallest dissimilarity to the other objects in the cluster. Because an actual data point is taken, PAM is more robust to noise and outliers than k-means clustering. However, the time complexity of PAM is much higher, which makes PAM less suited for large datasets. PAM has to evaluate it's assigned medoids at every iteration, which becomes increasingly more complex with large datasets.
- *Clustering LARge Applications (CLARA)* [38]: The CLARA algorithm combines elements of PAM and k-means to create a robust clustering method that is able to handle large datasets. This is done by taking subsets of the complete dataset. The PAM method is run on this subset to determine the medoids. After completion of PAM on the subset, the medoids are fixed and taken as an input for the full dataset. The mean of the dissimilarities between a medoid and assigned data points is monitored. The process is repeated until a threshold value of the mean of the dissimilarities is reached. CLARANS, presented by Ng and Han (2002), is an improvement of the CLARA algorithm [47]. The main difference is that CLARA is restrained to the initial subset it takes for determining medoids. After the medoids are determined, the full dataset is run before new medoids are selected. In CLARANS a random subset is drawn during the processing of the full datasets, which adjusts the medoids on the way. This resulted in a better clustering performance than CLARA when considering the same runtime [47].
- *Fuzzy C-means clustering (FCM)* [21][7]: In FCM, a cluster can be assigned to more than one cluster. This is done based on the minimization of the following objective function [54]:

  $$\min J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2 ; 1 < m < \infty \tag{4.2}$$

  Here, m is a fuzzy partition matrix exponent that allows for a certain degree of fuzzy overlap, which expresses how fuzzy the overlap between two clusters can be. The degree of membership a point x_i has in cluster j is expressed by u_{ij}. FCM is better suited to understand ability of patterns, noisy and incomplete data and it can provide approximate solutions faster compared to k-means clustering [54]. However due to the higher complexity of FCM, k-means works faster.
- *DBSCAN* [23]: Ester et al (1996) proposed the density-based partitional clustering DBSCAN. Opposed to the previously discussed clustering methods, the algorithm does not use distance as a measure for clustering, but density. The algorithm uses the input parameter ϵ, which represents the range of a neighbourhood. Using this parameter, three types of points can be defined, which are *core points*, *reachable points* and *outliers*:
  - A core point p is a core point if there are more then `MinPts` within a distance ϵ. Here, `MinPts` is a pre-defined input parameter.
  - A reachable point q is directly reachable from point p if the minimum distance is smaller than ϵ.
  - A point is an outlier if there is no other (core) point p within a minimum distance ϵ.

  Another difference compared to other clustering methods described above is that DBSCAN does not require a method like the elbow method to determine the number of clusters, as it's able to distinguish clusters itself. Furthermore, it is also robust to outliers as it can recognise outliers itself. A limitation is that is not completely deterministic as one point can belong to multiple clusters, as two core points can be within reach of a single point without being the same cluster. Furthermore, DBSCAN requires a careful selection of the `MinPts` and ϵ value.

Both the hierarchical as the partitional clustering methods described above are a selection of the clustering algorithms. Xu and Tian (2015) discussed all clustering algorithms to their knowledge in a paper [67].

4.1.3. Clustering algorithms applicability to research

A clustering technique could be a good solution for determining the groups to build M-ULDs with. Besides the connection times, a score for shipment importance can be used to cluster groups on. In terms of techniques, especially distance-based and density-based clustering algorithms could be a good fit to cluster the mix groups. Density based algorithms, like DBSCAN, are able to recognise shapes better in data and they are also able to identify outliers in comparison to distance-based methods. Furthermore, there is not an additional method required to determine the number of clusters k. However, as this research does not require the recognition of shapes and outliers also have to belong to a cluster, the advantages of a density based method are not really applicable to this case. Even more so, the calibration of the range parameter ϵ could be difficult as the data could be very continuous. When the data is continuous the outcome could be too few clusters or too much. Therefore distance-based methods, like k-means or PAM, would be a better candidate for clustering the mix groups.

4.2. Random Forest Algorithms

In 2001, Breiman [13] introduced the Random Forest (RF) Algorithm, which is a powerful non-parametric statistical method which is able to deal with regression, two-class and multi-class problems [27]. Random Forest is based on the idea of Classification And Regression Trees (CART), also introduced by Breiman [14]. Another name used for CART is a decision tree. The process can be modelled as a tree where the process starts with the input at the root. The input is partitioned into different classes until all input data is partitioned. At this point is leaf is reached, which gives an output class.

A single decision tree is not very reliable, as an anomaly in the data or small deviations can lead to incorrect classifications. Random Forest counters this by randomly generating multiple decision trees, hence the name. The trees are generated using a randomly sampled dataset. The process of generating this randomly sampled dataset is called *Bootstrap aggregating*, which is sometimes abbreviated as *bagging*. In bootstrapping a predefined number of samples is randomly taken from the full dataset to generate the randomly sampled dataset. In this process it is possible to select the same data entry multiple times.

The bootstrapped dataset is then used to randomly generate multiple decision trees. Here, nodes are created from a subset of the data features that split into multiple branches up until the final node, after which a leaf (or classifier) is reached. After the growing of the multiple decision trees, the non-bootstrapped entries of the dataset can be used to test the random forest. Every single decision tree has it's own outcome and the average is taken to determine the final estimate. For regression problems this is done by averaging the values, and for classification problems a voting system is used, where every tree "votes" for a certain class, and the class with the most "votes" is the estimated outcome. In the RF algorithm there are a few important parameters. The first is the number of sampled items in the bootstrapped dataset. Another parameter is the maximum number of directions for splitting at each node and `nodesize` is the maximum number of times the tree is split into new classes. There are multiple variations in the growing of a Random Forest. Genuer et al. (2017) contains an overview of alternatives, which are also compared on their performance [27].

An advantage of Random Forests is that it can handle high dimensional data and it is able to handle missing values. Besides this, it is relatively easy to implement as there are a wide variety of statistical packages that have Random Forest applications available and it is known as a relatively accurate supervised learning method [16]. A downside however, is that a training set has to be available with the appropriate outcomes. As this research involves a complete new decision making process, there is not such a training set available. Another downside is that the growing of many decision trees could make the model a so called 'black box', which is a model of which the inner workings are unknown. At a certain point it could no longer possible to determine how the model was able to make its prediction.

4.3. Reinforcement learning

Reinforcement learning is one of the three main categories in machine learning, along with supervised and unsupervised learning. The problem faced in reinforcement learning is that an agent must learn behaviour in a dynamic environment [36]. The learning mechanism works by using a balance between exploring uncharted territory and exploiting known territories. Kaelbling et al. (1996) described a basic Reinforcement learning model in their survey [36], which is visualized in Figure 4.4.

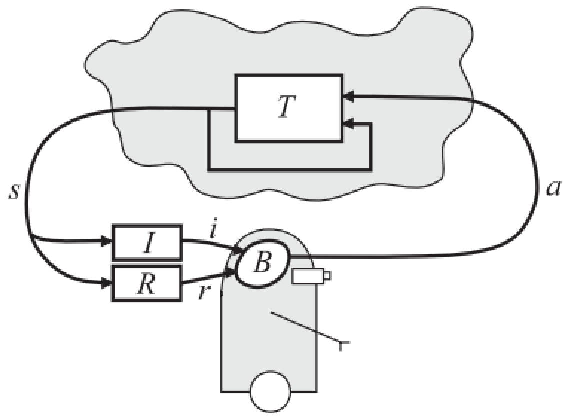

Figure 4.4: Example reinforcement learning model [36]

In Figure 4.4, an agent performs an action a, based on it's behaviour B. This action has an effect on the environment T which produces one or multiple states s, this state in converted to an input i and a reward r for the agent. This reward is based on the effect on it's action a. The behaviour B of the agent is then adjusted according to the quality of it's previous actions. In the long term, the agent's behaviour B which should maximize the sum of rewards r in the long term.

More formally, the model consists of:

- a discrete set of environment states S.
- a discrete set of agent actions A.
- a set of scalar rewards.

The actions of an agent do not only determine a reward, but also the next state. The next state in it's turn also has a set of rewards associated with the possible actions and so on. Therefore, it might occur that it is more beneficial for an agent to take a lower reward in a certain state in order to maximize the following states, resulting in a higher overall sum of rewards. This scenario is called Delayed Reinforcement [36].

Markov Decision Process (MDP)

The base of a Reinforcement Learning problems is a Markov Decision Process (MDP). The MDP is a mathematical way of modelling a decision making process where the outcomes are partly random and partly controllable. In order to formulate a MDP, the process should satisfy the Markov property. This means that the probability of the next state is only dependent on it's current state and not on it's past or future states. Another condition for a MDP is that it should be considering a set of states S, actions A, rewards R and a transition function T. The Markov Decision Process (MDP) is defined by van Otterlo et al. (2012) as [59]:

A MDP is a tuple $< S, A, T, R >$, in which S is a finite set of states, A a finite set of actions, T a transition function defined as $T : S \times A \times S \to [0,1]$ and R a reward function defined as $R : S \times A \times S \to \mathbb{R}$.

In a MDP, the transition function T combined with the reward function R form the model of the MDP. Using this MDP, a policy π can be developed by the agent, that can provide an action $a \in A$ for each state $s \in S$. For this, a deterministic policy can be formally expressed as $\pi : S \to A$. It is also possible to consider a stochastic policy, which is expressed as $\pi : S \times A \to [0,1]$ such that for each state $s \in S$, it holds that $\pi(s,a) \geq 0$ and

$\sum_{a \in A} \pi(s,a) = 1$ [59].

Optimality Criteria
It is the goal of the agent to maximize the rewards, which concerns more than just the immediate reward. Maximizing the rewards in a MDP can be defined in multiple ways in the form of optimality criteria. In van Otterlo et al. (2012) three models of optimality are distinguished [59]:

- *The finite horizon model*, which takes a finite horizon h. The agent has the goal to maximize the rewards over this horizon h.
$$E\left[\sum_{t=0}^{h} r_t\right] \tag{4.3}$$
- *The infinite horizon model*, which takes the long-term reward, but adds a discount factor γ into account. This discount factor γ has a value between 0 and 1, which decreases the further on the horizon a point is. Therefore, when using the infinite horizon model, the rewards of long-term decisions weigh less than short term decisions.
$$E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{4.4}$$
- *Average reward model*, which maximizes the average of all long-term rewards. This model is also sometimes described as the gain optimal policy.
$$\lim_{h\to\infty} E\left[\frac{1}{h}\sum_{t=0}^{h} r_t\right] \tag{4.5}$$

The decision about which model can depend on the circumstances of the reinforcement learning problem. For example, if the total length of the episode is known, the finite horizon model is best. But if it is not known, the infinite horizon model can be the best option. Koenig and Liu (2002) published an extensive overview of MDP models and their relationship with optimality [39].

Value functions
The optimality criteria can be directly linked to policies using value functions. The value functions are a method to estimate how good a certain state is for an agent or how good certain actions are for the next state. The value function is formally described by the value V of a state s under policy π: $V^\pi(s)$. For the infinite horizon model, the value function is as in Equation 4.6.

$$V^\pi(s) = E_\pi\Big\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s\Big\} \tag{4.6}$$

There is also a state action value function $Q^\pi(s,a)$. This function computes a value Q, given a state s and action a while following a policy π (Equation 4.7).

$$Q^\pi(s,a) = E_\pi\Big\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a\Big\} \tag{4.7}$$

Because the value function satisfy certain recursive properties, they can be recursively defined in the form of a Bellman equation (Bellman, 1966) [6]. For value functions, this is given in Equation 4.8. This equation shows that the expected value of a state is not only based on the immediate reward, but also on the possible future states which are weighted on their transition probabilities and the discount factor.

$$V^\pi(s) = \sum_{k'} T(s,\pi(s),s)\Big(R(s,a,s') + \gamma V^\pi(s')\Big) \tag{4.8}$$

The equation can be rewritten such that a maximum value can be determined, which in turn gives the optimal policy. The optimum value for the equation is denoted as V^* and the optimal policy corresponding to that optimum value as π^* (Equation 4.9).

$$V^*(s) = \max_{a\in A} \sum_{s\in S} T(s,\pi(s),s)\Big(R(s,a,s') + \gamma V^*(s')\Big) \tag{4.9}$$

The same process can be followed for the Q-value (Equation 4.10). Here, a relation between V^* and Q^* is used to find the optimum Q-value.

$$
\begin{aligned}
Q^*(s,a) &= \sum_{s \in S} T(s,\pi(s),s)\Big(R(s,a,s') + \gamma V^*(s')\Big) \\
V^*(s) &= \max_a Q^*(s,a)
\end{aligned}
\tag{4.10}
$$

The next step is to *learn* the optimal policy. The methods to learn the optimal policy can be split into two categories. These two methods are model-based methods and model-free methods [36]. For model-based methods, the model is known beforehand whereas for model-free methods the inner workings of the model are unknown [59]. Some of the most common model-based and model-free methods are described in subsection 4.3.1 and subsection 4.3.2 respectively.

4.3.1. Model-based methods

The general term for model-based methods is Dynamic Programming. There are two main types of dynamic programming, which are policy iteration by Howard (1960) [33] en value iteration by Bellman (1966) [6].

Policy iteration by Howard (1960) consists of two steps. The first step is called policy evaluation and during this step the current policy is run and the value function is computed. In the second step, called policy improvement an improved policy is developed by maximizing the value function. This process is repeated until an optimal policy is reached that can not be improved any further [33]. For finite MDPes, this process converges after a finite number of iterations. However, especially the policy evaluation part is computationally expensive as it has to compute value functions for every intermediate step [59].

Value iteration by Bellman (1966) only completes the evaluation as a first step [6]. Instead, it immediately blends the improvement phase into the iterations. In the process it estimates the value function directly. In the end, the process converges to V^* as given in Equation 4.9 and in the meantime it also computes Q^* along the way. Using this, a deterministic policy π can be determined. Even though one iteration of value iteration is faster in comparison to policy iteration, the number of required iterations can grow much faster because of the discount factor (for infinite horizons). Because of this, the overall computational time can be larger depending on the situation [59].

In general, policy iteration and value iteration can be seen as the two extremes in a spectrum, as mentioned by van Otterlo (2012) [59]. In between there are other methods like modified policy iteration by Puterman and Shin (1978) [51]. Another option could be to develop heuristics and search algorithms, as only a fraction of the state-space is relevant to look at for a certain problem. Usually these models provide reasonable good results within a short period of time.

4.3.2. Model-free methods

The model-based methods described above are all based on models where the inner workings of the system are known and a (almost) perfect model is available. In model-free methods this is not the case and an optimal policy is determined without having a perfect model. Within model-free methods, there are again two main approach branches. The first is to learn the transition and reward model first. If this model is completed or sufficiently accurate, all dynamic programming methods described in the previous section can be used to find an optimal policy. This approach is called *Indirect Reinforcement learning (RL)*. The second option is called *Direct Reinforcement learning (RL)*. In Direct RL the value of actions is directly estimated without first modelling the MDP. Besides these two methods hybrid forms also exists. However, most model-free methods follow either a direct or an indirect RL approach [59].

A problem to tackle is how to value the utility of a certain action, especially when the effects come much later. It is possible to value all the actions at the end of an episode, but this would require a lot of memory. So instead, the value of a sequence of actions is mostly estimated with taken discount rates into account. This method is called *Temporal difference learning*, which is the basis for most model-free reinforcement learning methods. The states and actions itself can be explored using several strategies, but the most commonly used is the ϵ-greedy strategy. In this policy there is a $1-\epsilon$ probability of taking the known most rewarding action, the probability ϵ corresponds to probability of taking another, randomly selected action. Other known exploration policies are the Boltzmann exploration strategy or optimistic Q-value optimization.

Temporal Difference Methods
As is mentioned before, the basis for most model-free reinforcement learning methods are Temporal difference methods. These methods learn by estimating values based on other estimates. It has an advantage over dynamic programming because it does not have to learn the MDP. Furthermore, when using an exploration strategy like the ϵ-greedy strategy there is no need to map the complete state space, which makes them easier to use in variable circumstances [59]. Three widely-used examples of temporal difference methods are:

- *Sutton's (1988) Temporal Difference (TD) methods* [57]: $TD(\lambda)$ is a family of temporal difference methods. In this equation, λ is a weighing factor where $0 \geq \lambda \geq 1$. The weighing factor is exponential where predictions are weighed up until k steps into the past by λ^k. For $TD(0)$, so $\lambda = 0$, the update rule is formulated as follows [59]:
$$V_{k+1}(s) = V_k(s) + \alpha\Big(r + \gamma V_k(s') - V_k(s)\Big) \tag{4.11}$$
Here, α is a learning rate value that has a value between 0 and 1. The value determines how many values are updated after the transition from state s to state s' and the received reward r after action a. A main difference between DP and Temporal Difference is that only a sample is updated and not the complete model as only one successive state is updated instead of a range of possible future states. A downside is that an additional model is needed to computed the expected value[59].

- *Q-learning* [63]: Watkins and Dayan (1992) formulated this model-free method to estimate Q-values. The basic idea behind Q-learning is that the Q-value is incrementally estimated based on the rewards for certain actions. A maximization operator is used to estimate the maximum Q-value of the next values. The update rule is formally described as follows [59]:
$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha\Big(r_t + \gamma \max_a Q_k(s_{t+1}, a) - Q_k(s_t, a_t)\Big) \tag{4.12}$$
Here the agent steps from state s_t to state s_{t+1} while performing action a_t and receiving reward r_t. For Q-learning there are no exploration policies as the method has the property that it converges to an optimal itself, without the help of an exploration policy based on the assumption that the algorithm will visit all states within a finite amount of time.

- *SARSA* [52] [53]: Where Q-learning is a so called "off-policy" method, which means it follows some exploration policy to find the optimal policy, SARSA is a on-policy method. With on-policy methods the current policy is iteratively used to compute the Q-value instead of the maximization operator. The value update rule can be written as:
$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha\Big(r_t + \gamma Q_t(s_{t+1}, a) - Q_t(s_t, a_t)\Big) \tag{4.13}$$
In this case, the action a_{t+1} is the action which would be taken by the current policy in the current state. SARSA will still converge to an optimal policy as long as all states will be visited in a finite amount of time. Additionally, SARSA is particularly useful in non-stationary environments where it is impossible to reach the optimal policy [59].

Monte Carlo Methods
Besides Temporal difference methods, there is also an option to use Monte Carlo methods to find an optional policy. In Monte-Carlo methods the frequency of state-action pairs is tracked and the sum of future rewards. The values of certain state-action pairs is based on this information. For each state $s \in S$ all the rewards obtained from this state are kept and the average is taken as the value of the state s. So the long-term reward is treated as a random variable and the value is estimated by taking the sampled mean. The difference with Temporal difference methods is that the values are based on an average return.

4.3.3. Applications of Reinforcement Learning
Reinforcement Learning has already been applied to some problems within the airline industry. An example of such an application is the pricing problem at revenue management departments. Lawhead and Gosavi

(2019) developed a model for a passenger pricing problem. This typical problem has the objective to maximize the revenue generated from a specific flight. Usually this is done by adjusting the price of a specific seat to the demand, in such a way that just before departure the aircraft is sold out. The RL model of Lawhead and Gosavi (2019) outperformed the best performing industry heuristic at that time (EMSR-b) [41]. Amaruchkul (2020) also addressed a pricing problem with reinforcement learning, but then for a cargo airline [3]. The model was formulated as a discrete time, finite horizon model and it also had the objective to maximize revenue. This model did not include the competition element, which also has an influence on the number of bookings. The model significantly outperformed the myopic policy, which was set as a baseline [3].

Both of these models show that Reinforcement Learning could be a good approach in situations with multiple inputs and a partly random process. However, when comparing to this research there are some differences that make RL not necessarily a good approach yet. An advantage of RL is that it can learn policies by providing rewards to the agent when it performs a good series of actions. But in this research, the formulation of a good reward system would be extremely difficult. If for example cost reduction would be chosen as the main reward function, it would be difficult to estimate what an increase in operational performance means in terms of revenue or cost. The change of operational performance on it's own would already be difficult to determine, as a lot of (external) factors have an impact on the operational performance. Therefore, in order to show the potential of dynamic selective loading, an other approach might be a better option for now.

5 Conclusions

In this literature study, an overview of relevant literature to the problem introduced in chapter 1 is given. On the operational side of cargo airlines, there is a large focus on ensuring that cargo is delivered as promised. This is a major quality standard for airlines that can make a shipper decide to choose for a certain airline. In order to improve this quality standard, selective loading rules were introduced to outstations. These rules are aimed to improve the composition of incoming ULDs at the hub in such a way that the work at the hub is minimized and the work is more spread out over the day. However, these rules are currently very generic, and the question arises whether a more data-driven dynamic loading advice could give more guidance and more informed advices to outstations instead of the set of fixed selective loading rules.

The present literature about air cargo load planning, described in chapter 2, reveals that a lot of research has been done on the assignment of items onto a ULD already. This particular problem can be addressed with packing problems, which is a well researched topic as well. Packing problems are not only useful in air cargo operations, but in many more industries. The research on models that address packing problems started with one-dimensional and two-dimensional problems, but in the last few years methods have been developed to address three-dimensional problems. However, these problems are aimed at utilising as much space with the least possible amount of ULDs. To our knowledge, there is not a model yet that groups items onto ULDs in such a way that the handleability at a hub is improved.

An important characteristic of air cargo operations are the large uncertainties. The dimensions of cargo are often unknown and even volumes and weights are not always given or correct. As cargo can be submitted up until 6 hours before departure and an outstation starts building approximately 12 hours before departure, a two stage advice could be considered. A framework to deal with uncertainties in two stages, called two-stage stochastic programming, is described in chapter 3.

For the generation of loading advices some Machine Learning techniques were researched. An overview is given in chapter 4. In this overview clustering algorithms, random forest and reinforcement learning were considered. Whereas clustering algorithms are unsupervised learning methods and able to learn without training. Random forest and reinforcement learning are both algorithms that require training. For random forest this is done by building multiple decision trees based on historic data or a test data set. In the case of reinforcement learning a reward system has to be implemented to train the model to perform certain actions in certain situations.

This literature study forms the basis of a thesis project, which will be a case study for KLM Cargo where historic data can be used to evaluate the effects and performance of the model. The goal of the research is to develop a model that has the objective to maximize the operational performance and contribution margin of a cargo airline by dynamically determining a ULD loading advice to outstations given shipment characteristics, capacity constraints and flight characteristics.

Bibliography

[1] Cargo iq - master operating plan, 2019.

[2] Cargo iq, 2019. URL https://www.cargoiq.org/.

[3] Kannapha Amaruchkul. Customized dynamic pricing for air cargo network via reinforcement learning. In *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, pages 213–224. Springer, 2020.

[4] James S.K. Ang, Chengxuan Cao, Mahmut Parlar, and Heng Qing Ye. Two-stage stochastic integer programming model for multiperiod sea cargo mix problem in container shipping industry. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 39(2):460–465, 2009. ISSN 10834427. doi: 10.1109/TSMCA.2008.2010754.

[5] Evelyn ML Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(2):173–184, 1955.

[6] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[7] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[8] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[9] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[10] Boeing. Boeing 787 by design - more revenue cargo. URL https://www.boeing.com/commercial/787/by-design/#/more-revenue-cargo.

[11] Andreas Bortfeldt and Gerhard Wäscher. Constraints in container loading-A state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20, 2013. ISSN 03772217. doi: 10.1016/j.ejor.2012.12.006. URL http://dx.doi.org/10.1016/j.ejor.2012.12.006.

[12] Felix Brandt and Stefan Nickel. The air cargo load planning problem - a consolidated problem definition and literature review on related problems. *European Journal of Operational Research*, 275(2):399–410, 2019. ISSN 03772217. doi: 10.1016/j.ejor.2018.07.013.

[13] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, oct 2001. ISSN 08856125. doi: 10.1023/A:1010933404324. URL https://link.springer.com/article/10.1023/A:1010933404324.

[14] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[15] Air France KLM Martinair Cargo. One pager selective loading rules, 2021.

[16] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[17] Felix TS Chan, Rajat Bhagwat, Niraj Kumar, MK Tiwari, and Philip Lam. Development of a decision support system for air-cargo pallets loading problem: A case study. *Expert Systems with Applications*, 31(3):472–485, 2006.

[18] Hugh Courtney, Jane Kirkland, and Patrick Viguerie. Strategy under uncertainty. *Harvard business review*, 75(6):67–79, 1997.

[19] George B Dantzig. Linear programming under uncertainty. *Management science*, 1(3-4):197–206, 1955.

[20] Felipe Delgado, Ricardo Trincado, and Bernardo K. Pagnoncelli. A multistage stochastic programming model for the network air cargo allocation under capacity uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 131:292–307, nov 2019. ISSN 13665545. doi: 10.1016/j.tre.2019.09.011.

[21] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[22] Harald Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(90)90350-K. URL https://www.sciencedirect.com/science/article/pii/037722179090350K. Cutting and Packing.

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[24] KiM | Netherlands Institute for Transport Policy Analysis. Key transport figures 2018. URL https://english.kimnet.nl/binaries/kimnet-english/documents/documents-research-publications/2019/01/11/key-transport-figures-2018/Key+transport+figures+2018.pdf. Accessed: 5-3-2021.

[25] C. Fraley. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, aug 1998. ISSN 0010-4620. doi: 10.1093/comjnl/41.8.578. URL https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/41.8.578.

[26] Arnaud Fréville. The multidimensional 0-1 knapsack problem: An overview, may 2004. ISSN 03772217.

[27] Robin Genuer, Jean Michel Poggi, Christine Tuleau-Malot, and Nathalie Villa-Vialaneix. Random Forests for Big Data. *Big Data Research*, 9:28–46, sep 2017. ISSN 22145796. doi: 10.1016/j.bdr.2017.07.003.

[28] PC Gilmore and RE Gomor. Multistage cutting problems of two and mord dimensions. *New York.: Thomas J. Watson Research Center, Yorktown Heigh*, 1964.

[29] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: An efficient clustering algorithm for large databases. *ACM Sigmod record*, 27(2):73–84, 1998.

[30] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

[31] Julia L Higle and Suvrajeet Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research*, 16(3):650–669, 1991.

[32] Hai Thi Hong Ha and Narameth Nananukul. Air Cargo Loading Management System for Logistics Forwarders. In *International Conference on Urban Planning, Transport and Construction Engineering*, pages 51–58, 2016. ISBN 9789384422516. doi: 10.17758/UR.U0116322. URL http://dx.doi.org/10.17758/UR.U0116322.

[33] Ronald A Howard. Dynamic programming and markov processes. 1960.

[34] IATA. Air cargo market analysis december 2020, . URL https://www.iata.org/en/iata-repository/publications/economic-reports/air-freight-monthly-analysis---december-2020/. Accessed: 5-3-2021.

[35] IATA. Cargo strategy 2018, . URL https://pdf4pro.com/amp/view/iata-cargo-strategy-2018-50960f.html. Accessed: 5-3-2021.

[36] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. ISSN 10769757. doi: 10.1613/jair.301.

[37] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[38] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[39] Sven Koenig and Yaxin Liu. The interaction of representations and planning objectives for decision-theoretic planning tasks. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(4):303–326, 2002.

[40] Dao Lam and Donald C Wunsch. Clustering. *Academic Press Library in Signal Processing*, 1:1115–1149, 2014.

[41] Ryan J Lawhead and Abhijit Gosavi. A bounded actor–critic reinforcement learning algorithm applied to airline revenue management. *Engineering Applications of Artificial Intelligence*, 82:252–262, 2019.

[42] Andrea Lodi, Silvano Martello, and Michele Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, sep 2002. ISSN 03772217. doi: 10.1016/S0377-2217(02)00123-6.

[43] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[44] Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*. Springer, 2005.

[45] Silvano Martello, David Pisinger, and Daniele Vigo. The Three-Dimensional Bin Packing Problem. Technical Report 2, 2000. URL https://about.jstor.org/terms.

[46] F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4):354–359, nov 1983. ISSN 0010-4620. doi: 10.1093/comjnl/26.4.354. URL https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/26.4.354.

[47] Raymond T. Ng and Jiawei Han. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016, 2002.

[48] 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(1-2):187–213, 2016. ISSN 14753995. doi: 10.1111/itor.12111.

[49] Célia Paquay, Sabine Limbourg, and Michaël 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(1):52–64, 2018. ISSN 03772217. doi: 10.1016/j.ejor.2017.11.010.

[50] Célia Paquay, Sabine Limbourg, Michaël Schyns, and José Fernando Oliveira. MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints. *International Journal of Production Research*, 56(4):1581–1592, feb 2018. ISSN 1366588X. doi: 10.1080/00207543.2017.1355577. URL https://www.tandfonline.com/doi/abs/10.1080/00207543.2017.1355577.

[51] Martin L Puterman and Moon Chirl Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

[52] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[53] Gavin Adrian Rummery. *Problem solving with reinforcement learning*. PhD thesis, Citeseer, 1995.

[54] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, dec 2017. ISSN 18728286. doi: 10.1016/j.neucom.2017.06.053.

[55] SeaRates. Air cargo uld containers: internal and external dimensions. URL https://www.searates.com/reference/uld/. Accessed: 5-3-2021.

[56] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[57] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1): 9–44, 1988.

[58] Iordanis Tseremoglou. A combined forecasting and packing logistics: an application to air cargo under uncertainty in transportation model to optimize decision-making. *TU Delft Repository*, 2020.

[59] Martijn van Otterlo and Marco Wiering. *Reinforcement Learning and Markov Decision Processes*, pages 3–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_1. URL https://doi.org/10.1007/978-3-642-27645-3_1.

[60] Richard M Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[61] Linda Veldhuizen, R Nunez-Queija, and H.J. van der Sluis. A Simulation model to evaluate the workload at KLM Cargo. *University of Amsterdam - Faculty of Operations Research MSc. Thesis*, 2012.

[62] Gerhard Wäscher, Heike Haußner, and Holger Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, dec 2007. ISSN 03772217. doi: 10.1016/j.ejor.2005.12.047.

[63] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[64] H Martin Weingartner and David N Ness. Methods for the Solution of the Multidimensional 0/1 Knapsack Problem. *Operations Research*, 15(1):83–103, 1967. doi: 10.1287/opre.15.1.83. URL http://pubsonline.informs.org.https//doi.org/10.1287/opre.15.1.83http://www.informs.org.

[65] Y Wu. Modelling containerisation of air cargo forwarding problems. *Production Planning and Control*, 19(1):2–11, 2008.

[66] Y. Wu. Modelling of containerized air cargo forwarding problems under uncertainty. *Journal of the Operational Research Society*, 62(7):1211–1226, jul 2011. ISSN 14769360. doi: 10.1057/jors.2010.84. URL www.palgrave-journals.com/jors/.

[67] Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, jun 2015. ISSN 2198-5804. doi: 10.1007/s40745-015-0040-1. URL https://link.springer.com/article/10.1007/s40745-015-0040-1.

[68] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16 (3):645–678, 2005.

[69] C Yabing and C Gao. Birch: An efficient clustering method for very large databases, 1999.

[70] Berislav muk. Quality of life indicators in selected european countries: Statistical hierarchical cluster analysis approach. *Croatian Review of Economic, Business and Social Statistics*, 1:42–54, 12 2015. doi: 10.1515/crebss-2016-0004.