Optimizing Cargo Operations for a Combination Airline

Master of Science (MSc.) Thesis

Sidharatha Shanker Thakur

TUDelft

Optimizing Cargo Operations for a Combination Airline

Master of Science (MSc.) Thesis

by

Sidharatha Shanker Thakur

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 19-June-2023 at 14:00 hrs

Student number:	4541340
Project duration:	October 2021 – June 2023
Thesis committee:	**Chair:** Dr. BF (Bruno) Lopes Dos Santos
	Daily Supervisor: Dr. A. (Alessandro) Bombelli
	External Examiner: Dr. S. (Stefano) Fazi
	External Examiner: Dr. F. (Felipe) Delgado

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover image taken from https://www.nt.nl/luchtvracht/2017/03/16/lufthansa-cargo-sluit-2016-af-met-rode-cijfers

TU Delft

Acknowledgements

This thesis marks the final part of my masters degree at the faculty of Aerospace Engineering of the Delft University of Technology (TU Delft).

I would like to thank my daily supervisor Alessandro Bombelli for all his support during the process. His insights and recommendations were extremely useful in performing the research and writing this master thesis. I always promised Alessandro that our meetings would be short, however they almost always ended up taking a life of their own and being longer than expected. I can only hope that this was due to our mutual interest in the thesis topic and a shared mindset of brainstorming new ways to approach the problem! Furthermore, I would also like to thank Bruno Santos and Paul Roling for providing valuable suggestions during milestone meetings. Also, I would like to thank all members of my thesis committee: Alessandro Bombelli, Bruno Santos, Stefano Fazi and Felipe Delgado for their participation.

Last but not least, I would like to thank my family and friends for their support during, not only this thesis project, but also throughout my time at the TU Delft. They provided the much-needed healthy escape from studying, which was essential for me to recharge and relax.

As my master's thesis and my time at TU Delft comes to an end, I am reminded of a quote supposedly attributed to Socrates: "Education is the kindling of a flame, not the filling of a vessel". In my interpretation of this quote, studying at TU Delft was not solely about obtaining a degree (i.e. filling of a vessel), but also about being exposed to and nurturing a curiosity for new ideas that will contribute to shaping the future. I am confident that the skills I have learned at TU Delft, both analytical and otherwise, have prepared me to maintain a perpetual curiosity for new ideas that will shape our future.

SIDHARATHA SHANKER THAKUR,
12 June 2023,
Delft, the Netherlands

Contents

List of Abbreviations

ACSRP	Air Cargo Schedule Recovery Problem
CAGR	Compund Annual Growth Rate
CD-ABM	Request Clustered Detour Ratio Reduced Model
CD-CGR-ABM	Request Clustered Detour Ratio Reduced Column Generation Reduced Arc Based Model
D-CGR-ABM	Detour Ratio Reduced Column Generation Reduced Arc Based Model
F-ABM	Full Arc-Based Model
IATA	International Air Transport Association
MILP	Mixed Integer Linear Programming
MTOW	Maximum Takeoff Weight
OD	Origin-Destination
SITC	Standard International Trade Classification
ULD	Unit Load Devices

Introduction

Air cargo is a crucial means of transporting cargo quickly around the world. The global demand for air cargo is expected to increase in the coming years. Combination airlines, also known as combination carriers, transport cargo using dedicated full-freighters as well as the cargo capacity available in the belly space of passenger flights. While combination carriers represent an important segment within the air cargo industry, there is limited research on their operations. The objective of this research is to develop a decision tool that assists combination carriers in routing and scheduling their full-freighters while also considering the cargo-carrying capacity provided by passenger flights. The decision tool must also have a short runtime so that it can be effectively used in real-life settings. Therefore, another key aspect of this research is to explore ideas that can help reduce the runtime of the decision tool.

The decision tool can be used by combination carriers to cost efficiently route and schedule their full-freighters, enabling them to optimize their operations. Additionally, it can support them in conducting long-term simulation studies to help in making informed decisions regarding future full-freighter fleet planning. From a societal perspective, air cargo plays a vital role in global supply chains. Although the field is vast, optimizing the operations of a significant industry player such as combination carriers can contribute to cost reduction and/or increased supply. This will benefit numerous stakeholders.

This research consisted of three key phases. In the literature review phase, relevant research pertaining to the thesis topic was analyzed. The mid-term phase involved creating a small working model of the decision tool. Lastly, in the final phase, the decision tool was completed and extensively tested on numerous problem instances.

This thesis report is organized as follows: In Part I, the scientific paper is presented. Part II contains the relevant Literature Study that supports the research.

I

Scientific Paper

Optimizing Cargo Operations for a Combination Airline

Sidharatha Shanker Thakur*

Delft University of Technology, Delft, The Netherlands

Abstract

Air cargo is vital for efficient global supply chains, enabling the rapid transport of valuable goods. Combination airlines, a significant player in the industry, use both dedicated full freighters and the belly space of scheduled passenger flights to transport cargo. This research develops a decision tool that combination airlines can use to optimize the routing and scheduling of full freighters. By incorporating anticipated cargo demand data as an input, the tool enables the most cost-effective routing and scheduling of full freighters, also taking into account the available cargo carrying capacity offered by passenger flights. The tool combines and leverages the intrinsic advantages of passenger flights (frequencies) and full-freighters (capacity). The decision tool relies on a Mixed Integer Linear Program (MILP) to minimize operational costs while adhering to operational constraints faced by a combination airline. Although this method performs well for small instances, it suffers from long computational times when applied to larger problem instances. To address this, modifications are introduced to shorten the runtime by reducing the problem size. Specifically, clustering is used to group cargo requests together, and column generation is used to efficiently select optimal routing decisions. These modifications lead to a shorter runtime, albeit with an acceptable increase in operating costs. The decision tool serves multiple purposes. In the short term, it can be used to route full freighters and identify non-profitable full-freighter origin-destination pairs, among other applications. Additionally, it can be used to analyze, simulate, and provide recommendations for future long term full freighter fleet planning decisions.

Keywords: Air cargo, combination airlines, full freighters, belly space, routing and scheduling, decision tool, optimization.

1 Introduction

Air cargo plays a key role in integrating global supply chains and facilitating the fast transportation of goods across the globe. Its importance was observed during the COVID-19 pandemic as lifesaving medical equipment and millions of vaccines were transported by air cargo carriers around the world [United Nations Department of Global Communications, 2023]. In general, while air cargo accounts for only around 1% of global shipments by weight, it represents approximately 35% by value [IATA, 2015]. Boeing predicts that "air cargo traffic is forecast to grow at a rate of 4.0% per year over the next 20 years" [Crabtree et al., 2020], driven by increased trade and the rising demand from e-commerce . Similarly, Airbus expects a compound annual growth rate (CAGR) of 3.2% until 2041 for world air cargo traffic [Shparberg and Lange, 2022].

There are different types of air cargo carriers. For instance, some carriers exclusively operate full-freighters, which are aircraft designed solely for cargo transportation. Others use the available belly-space of passenger flights to transport cargo. Combination airlines (or combination carriers), such as Lufthansa Cargo, Turkish Cargo, Qatar Airways Cargo, and others, employ both dedicated full-freighters and the belly-space of scheduled passenger aircraft to transport cargo. Combination carriers are a significant segment of air cargo carriers, accounting for an estimated 36% of the total revenues earned by all air cargo carriers [Crabtree et al., 2020].

Given the projected growth of the air cargo industry, numerous combination carriers are placing orders for full-freighters in addition to their regular orders for passenger aircraft. For example, Qatar Airways has announced plans to B777X full-freighters from Boeing [Bushey et al., 2022]. Etihad Airways has expressed its intention to acquire 7 A350F full-freighters from Airbus [Airbus, 2022]. Additionally, Singapore Airlines has also made arrangements to acquire 7 A350F full-freighters from Airbus [Singapore Airlines, 2022].

There are notable differences between cargo operations and passenger operations, particularly in terms of demand volatility. Cargo demand tends to be more unpredictable compared to passenger demand. This is due to the fact that cargo capacity is often booked well in advance, but there are usually limited penalties for not using the booked capacity [Feng et al., 2015]. As a result, instances of no-shows, rebooking, or the actual cargo payload being lower than the initially booked amount are quite common.

*Msc Student, Sustainable Air Transport, Faculty of Aerospace Engineering, Delft University of Technology

These problems naturally extend to the cargo operations of combination carriers, requiring them to employ both proactive and reactive strategies to address these challenges. A proactive approach involves improving cargo demand forecasting and developing a full-freighter routing and scheduling strategy that synergizes with the expected available capacity on scheduled passenger flights. On the other hand, a reactive approach involves making last-minute scheduling and routing adjustments to capture additional demand or reduce costs. However, it is important to note that combination carriers have limited options for implementing reactive measures. Making sudden and significant changes to the passenger flight schedule can create more problems than solutions, as it may inconvenience passengers and lead to missed flight connections. Similarly, for full-freighters, making last-minute drastic changes to the schedule and routing is often impractical due to operational constraints.

From an academic standpoint, the majority of research has primarily focused on either the transportation of cargo in the belly-space of passenger aircraft or exclusively on full-freighters. However, the case of combination carriers, which use both the belly-space and dedicated full-freighters for cargo transportation, has received limited attention. This paper aims to address this gap by developing a decision support tool for combination carriers that can help them optimize the routing and scheduling of their full-freighters. The tool is designed to operate within a tactical timeframe, typically a few months before the commencement of operations. During this period, the passenger flight schedule is already finalized, and the routing of the full-freighters must be planned in order to enhance and complement the cargo carrying capacity being provided by the passenger flights.

In essence, this decision tool is intended to be used in a proactive manner for routing and scheduling full-freighters. It uses anticipated cargo demand, among other factors, as an input to optimize the routing and scheduling of full-freighters. By analyzing historical data, including shipment volumes and patterns, and applying trend analysis, combination carriers can estimate the expected cargo demand for future periods.

This decision tool relies on anticipated cargo demand data at the Unit Load Device (ULD) granularity. A ULD refers to the individual pallets or containers that are loaded onto an aircraft. Each ULD is composed of smaller cargo request packages such as mail, baggage etc [IATA, 2023]. In the anticipated cargo demand data, each ULD needs to be transported from its origin to its destination within a specified timeframe. The tool's objective is to create a minimum cost routing and scheduling plan for full-freighters such that all ULDs are effectively transported, either on dedicated full-freighters or within the available belly-space of pre-defined scheduled passenger flights. Not being able to a cargo request will result in large penalty costs. The decision tool considers various constraints, including range limitations, time restrictions, and capacity constraints, among others. These constraints play a crucial role in establishing a feasible full-freighter routing and scheduling solution.

Aside from generating a feasible and cost-effective routing plan for full-freighters, the decision tool must also have a runtime that is sufficiently short for practical applications. The conceptualization of this decision tool is focused on a proactive perspective rather than a disruption management standpoint, where extremely short runtimes are required. Nevertheless, it is crucial for the decision tool to provide solutions within a reasonable timeframe in order to be useful in a real-world scenario. Routing and scheduling optimization problems are commonly formulated as Mixed Integer Linear Programs (MILPs) and are often based on a time space network. However, one drawback of these formulations is that they have long runtimes. To address this, several methods are proposed to help reduce the runtime of the tool. It should be noted that reducing the runtime will likely result in a solution that has increased operational costs. Therefore, a trade-off analysis between the increase in operational cost and the decrease in runtime is performed to determine the usefulness of these modifications.

This paper is organized as follows: Section 2 presents relevant literature related to the problem. Section 3 describes the methodology employed to develop the decision tool. It also elaborates on various modifications implemented to improve the tool's speed. In Section 4, a list of problem instances on which the decision tool is tested is provided. These instances serve as test cases to evaluate the performance of the tool. Section 5 presents an illustrative example of the decision tool in action. This section also provides the results of the tool's performance in terms of cost and time across all instances. Finally, in Section 6, the main findings are discussed, and further recommendations are provided for improving the decision tool.

2 Literature Review

This research focuses on the air cargo operations of combination carriers. The available literature looks primarily at the air cargo landscape and the operations of air cargo carriers, specifically combination carriers.

The transportation of cargo, airmail, and freight by aircraft is the primary function of the air cargo industry, with five key players involved in the process [Feng et al., 2015]. These participants are the shipper, consignee, road transporter, freight forwarder, and air cargo carrier. The shipper is responsible for sending the cargo, while the consignee is the intended recipient. The road transporter manages ground transportation between the airport and the shipper/recipient. Freight forwarders act as agents and coordinate the entire shipment process, including documentation, preparation, and capacity booking. Air cargo carriers are responsible for flying cargo between airports and usually interact primarily with freight forwarders. While each participant in the air cargo shipment process has their own set of problems and areas for improvement in their operations, this research focuses primarily on air cargo carriers.

[Crabtree et al., 2020] categorizes air cargo carriers into four groups: belly-only carriers, cargo specialists, combination carriers, and integrated express carriers. Belly-only carriers only offer cargo capacity in the lower deck of passenger aircraft, they are limited in the size and type of cargo they can transport. Cargo specialist carriers only operate full freighters and can carry large and oddly sized cargo, often with stopovers to maximize capacity. Combination carriers operate both passenger and dedicated full-freighters. They can use their passenger network to feed cargo to their cargo network. Integrated express carriers provide a door-to-door service using a fleet of trucks and aircraft, prioritizing short transport times and imposing size & handling requirements on the cargo [Brandt and Nickel, 2019].

Air cargo accounts for less than 1% of global trade by volume, but it represents approximately 35% of global trade by value [IATA, 2015]. According to [Brandt and Nickel, 2019], the average cost of shipping cargo via aircraft is between "10-15 times" more expensive as compared to surface transport. This raises the question of what justifies the high cost of air cargo. The main advantage of air cargo is its fast-shipping time, making it suitable for certain situations such as time-sensitive goods like technology and fashion trends, just-in-time logistics for complicated machine parts and critical spares, high-value goods with security demands, and perishable items like fruits, vegetables, flowers, and vaccines that degrade quickly and require fast transport.

There are a few papers that specifically focus on the operations of combination carriers. [Reis and Silva, 2016] provide a comprehensive overview of the business models adopted by combination carriers. The study is based on interviews conducted with staff members from various airlines. [Lange, 2019] examines the correlation between combination carriers and their departure on-time performance, specifically within the American context. The study reveals that combination carriers experience higher departure delays compared to pure passenger carriers. [Hong et al., 2023] examines the challenges encountered by two Korean combination airlines in their operations. The paper highlights the main challenge faced by combination carriers, which is to effectively balance supply and demand in the industry. Additionally, it highlights the fierce competition within the air cargo sector, as integrated express carriers are expanding and introducing new products, while new competitors are entering the market. Although the research specifically focuses on the two Korean airlines, the findings are relevant and applicable to other combination carriers as well.

The use of operations research techniques in the domain of air cargo operations is well-established. [Brandt and Nickel, 2019] offers an extensive literature review that covers a range of problems within the field of air cargo load planning. This includes challenges such as optimizing the packing of Unit Load Devices (ULDs) with cargo, determining the optimal timing for ULD packing, and placing ULDs in aircraft while considering weight and balance requirements.

[Feng et al., 2015] also provides a comprehensive literature review on air cargo operations. The paper examines various problems from different perspectives, such as those of air cargo carriers and freight forwarders. In the case of cargo carriers, it also explores topics including cargo carrier fleet planning and flight scheduling. The literature review presented in this paper also highlights the limited number of studies that have examined the case of routing and scheduling for combination carriers.

2.1 Routing and Flight Scheduling

A number of papers have focused on the routing and flight scheduling decisions for cargo carriers. The study by [Yan et al., 2006] proposes a mixed integer linear program that combines airport selection, fleet routing, and timetable setting for pure cargo carriers. The authors use a time space network formulation to solve the problem. This is essentially a graph consisting of nodes and arcs (also known as edges). A node denotes an airport at a particular point in time. The arcs denote movement from one node to another. There can be different kinds of arcs in a time space network, each representing different movement types. For instance, a flight arc can represent the movement of a full-freighter from one airport to another airport. The time space network is used to create a MILP formulation. One drawback of using a time space network is that the size of the MILP formulation explodes as the problems become bigger. The authors propose a family of heuristics that aim to simplify the problem by reducing the number of variables in the decision space. For example, the "non-stop heuristic" only considers non-stop flights, while the "one-stop heuristic" considers flights with a maximum of one layover. The "mixed-stop heuristic" determines the number of stops based on the origin-destination distance.

[Yan and Chen, 2008] is a similar paper that looks at optimal flight scheduling models for cargo carriers in alliance with other cargo carriers. [Tang et al., 2008] is another similar paper that focuses on developing a model to schedule passenger, cargo, and combi flights. The term "combi flights" refers to flights that have the capability to accommodate both passengers and cargo. It is important to note the subtle distinction between combi flights and combination carriers. In the case of combi flights, the aircraft is designed to transport cargo and passengers on the main deck, with a section of the main-deck dedicated for ULDs while another section has seats for passengers. On the other hand, combination carriers employ dedicated full-freighters and use the belly-space of passenger aircraft to transport cargo.

[Delgado et al., 2020] present a study on how to recover from demand disruptions on a cargo network caused by the uncertainty in cargo booking demand. They introduce the air cargo schedule recovery problem (ACSRP), which aims to cost effectively reschedule full-freighters and reroute cargo given cargo demand disruptions. This paper is one of the first to focus on recovery for freighters, as previous studies mainly looked at passenger flight recoveries. To formulate the problem, the authors also use a time space network with nodes denoting airports for every time period. The network has four types of arcs that have a cost, and a penalty function is defined to denote the cost of rescheduling, focusing on the additional costs incurred by reassigning the crew to the new schedule. The ACSRP is modelled as a mixed integer linear programming problem with the objective of minimizing the additional costs of rescheduling.

[Delgado and Mora, 2021] uses a metaheuristic approach to address the air cargo recovery problem in the presence of demand disruptions. Similar to [Delgado et al., 2020], the paper focuses on adjusting the schedule of full-freighter flights in the short term to respond to cargo demand disruptions. The objective was to optimize the routing of full-freighters and cargo requests while minimizing deviations from the original schedule. It also uses a time space network approach, with nodes and arcs. The model has a block structure allowing it to be decomposed into sub-problems for each aircraft in the set of aircraft. There is a one linking condition that connects the various sub-problems.

[Xiao et al., 2022] performs simultaneous full-freighter tail assignment and cargo routing. The cargo routing is based on itineraries, which can lead to a significant increase in the problem size due to the potentially large number of itineraries. To solve this challenge, the authors propose a column-generation inspired heuristic as a solution approach.

2.2 Literature Gap and Problem Statement

Most of the research in air transportation primarily focuses on passenger operations, while cargo operations have received less attention. Furthermore, within the field of cargo operations, the specific operations of combination carriers has largely been overlooked, despite the fact that combination carriers are among the major air cargo carriers in terms of revenue. [Tang et al., 2008] is perhaps one of the few papers that considers the routing and scheduling of both passenger and full-freighter aircraft. However, it does so from a longer term point of view where passenger flights can still be scheduled. However, in practice, the uncertain nature of cargo demand means that actual cargo requests are often unknown until closer to the departure time. As a result, a more tactical planning model is necessary, which can route and schedule full-freighters after the passenger scheduling has been completed.

The research objective is to develop a decision tool for combination carriers to determine the optimal routing of full-freighters while adhering to constraints such as flow conservation, cargo release & due times and aircraft capacity limitations. The process is approached from a tactical perspective, where the scheduling of passenger flights has already been determined, but there is flexibility in the routing and scheduling of full-freighters. The goal is to minimize operating costs while ensuring that all the constraints are satisfied. The decision tool will use anticipated cargo demand data at the ULD granularity level to perform the full-freighter routing and scheduling.

Several previous papers have employed the concept of a time space network to construct a Mixed-Integer Linear Programming (MILP) model. While this approach ensures optimal results, it often results in long runtimes. Hence, another goal of this paper is to explore methods that can help reduce the runtime of the model. While this may lead to a sub-optimal solution, the gain in shorter runtimes can make this a valuable tradeoff.

Given the uncertain nature of cargo demand data, it is possible for the actual cargo demand, known only a few days before operations, to differ from the cargo demand used to create the initial routeing and scheduling. To minimize the effects of this uncertainty, two approaches can be considered. First, using better tools to anticipate cargo booking demand, such as advanced statistical techniques or machine learning algorithms, Second, solve a rescheduling problem that can make permissible adjustments to the current routing and scheduling of full-freighters a few days before operations. This task is left for future research.

3 Methodology

This section provides an overview of the various models that can be used to develop the decision tool. Section 3.1 explains the concept of a time space network, which serves as a fundamental component for making routing and scheduling decisions. The first model, referred to as the full arc-based model (F-ABM), is discussed in detail in Section 3.2. In Section 3.3, a model that incorporates cargo request clustering and also uses a detour ratio-based approach to reduce the problem size is presented. In Section 3.4, a column generation-based method is introduced that can help reduce the problem size. Finally, Section 3.5 presents a model that combines all the key features of the previous models. It integrates cargo request clustering, detour ratio reduction, and column generation techniques into a single model.

3.1 Time Space Network

A time space network forms the core of the routing and scheduling process. It consists of nodes ($\mathcal{N}$) and various arcs ($\mathcal{A}$). Cargo flight arcs ($\mathcal{A}^{CF}$) represent the movement of full-freighters between two airports. When a full-freighter traverses this arc, it indicates that it is flying from one airport to another. On the other hand, when a cargo request traverses this arc, it signifies that the cargo is being transported on a full-freighter that is flying along this arc. Ground arcs ($\mathcal{A}^{G}$) connect consecutive nodes within the same airport. Both full-freighters and cargo requests can traverse these arcs, indicating that they are on the ground at the airport. Passenger flight arcs ($\mathcal{A}^{PF}$) correspond to scheduled passenger flights. Cargo requests can traverse these arcs, indicating that they are being transported in the belly-space of a passenger flight. Multiple passenger flight arcs can exist between two nodes if multiple passenger flights are scheduled. Lastly, the no service arcs ($\mathcal{A}^{NS}$) correspond to the direct movement of a cargo request from its origin node to its destination node, signifying the scenario in which the cargo request cannot be delivered. Consequently, a high cost is associated with using a no service arc. Appendix A presents a diagram of the time space network.

An important aspect of time space networks is the temporal discretization. The time window is continuous, but it needs to be discretized for modeling purposes. Smaller time discretization intervals result in larger models with longer runtimes, but they are likely to provide better results. Linking the continuous time values with the discretized time values must be done properly, especially for the release/due times of cargo requests and the timings of passenger flights. For cargo requests, the release node corresponds to the first available node after the cargo is released. For example, if the cargo is released at 11:00, but the nodes are discretized as 10:00, 12:00, etc., then the cargo is assumed to be released at 12:00. Similarly, for the due time, the last available node before the cargo is due is the due node. Using the same example, if the cargo is due at 11:00, then it is assumed to be due at 10:00. The opposite principle is applied to passenger flights. If a passenger flight is scheduled to depart at 11:00, then the departure node is the node at 10:00. If a passenger flight is scheduled to arrive at 11:00, then the arrival node is at 12:00.

The cargo flight arcs play a key role in the time space network as they determine the possible routing and scheduling options of full-freighters. They are generated for all origin-destination pairs where at least one full-freighter in the fleet is capable of flying, taking into account range constraints. At each node, a cargo flight arc is created from the airport to all other airports that can be reached. The endpoint of a cargo flight arc depends on the flying time between the origin and destination, taking into account the turnaround time as well. It is important to note that a single cargo flight arc is generated that can be traversed by any aircraft in the fleet.

3.2 Full Arc Based Model (F-ABM)

A combination carrier operates a fleet of full-freighters ($k \in \mathcal{K}^C$) to handle a range of cargo requests ($r \in \mathcal{R}$). Each request is associated with specific release and due time, as well as corresponding release and due airports. As a result, each cargo request $r \in \mathcal{R}$ is assigned a release node (N_r^+) and a due node (N_r^-) within the time space network.

Table 1 lists the sets used to construct the MILP formulation. Table 2 lists the decision variables of the MILP model. Tables 3 and 4 give other important parameters.

Table 1: Sets

Set	**Description**
$\mathcal{K}^C$	The set of all full-freighter aircraft in the fleet.
$\mathcal{T}^C$	The set of all fleet types that are present among the full-freighter aircraft.
$\mathcal{R}$	The set of cargo requests to be delivered.
$\mathcal{N}$	The set of all nodes in the time space network. Every node represents an airport at a particular point in time.
$\mathcal{N}_S$	The set of all nodes in the time space network at the start of the time window; $\mathcal{N}_S \subset \mathcal{N}$.
$\mathcal{N}_E$	The set of all nodes in the time space network at the end of the time window; $\mathcal{N}_E \subset \mathcal{N}$.
$\mathcal{A}^{CF}$	The set of cargo flight arcs.
$\mathcal{A}^{PF}$	The set of passenger flight arcs.
$\mathcal{A}^{G}$	The set of ground arcs.
$\mathcal{A}^{NS}$	The set of no service arcs.
$\mathcal{A}$	The set of all arcs in the time space network, $\mathcal{A} := \mathcal{A}^{CF} \cup \mathcal{A}^{PF} \cup \mathcal{A}^{G} \cup \mathcal{A}^{NS}$

Table 2: Decision Variables for MILP

Decision Variable	**Description**
$x_{f,k}$	This binary variable is 1 if arc $f \in \mathcal{A}^{CF} \cup \mathcal{A}^{G}$ is traversed by aircraft $k \in \mathcal{K}^{C}$.
$q_{f,r}$	This binary variable is 1 if arc $f \in \mathcal{A}$ is traversed by request $r \in \mathcal{R}$.
$u_{f,t}$	This variable represents the total weight of requests carried on a cargo flight arc $f \in \mathcal{A}^{CF}$ by an full-freighter aircraft of type $t \in \mathcal{T}^{C}$.
$y_{f,t}$	This binary variable is 1 if there is a full-freighter of type $t \in \mathcal{T}^{C}$ traversing cargo flight arc $f \in \mathcal{A}^{CF}$.

Table 3: Parameters

Parameter	**Description**
Dist_f	The total flying distance of cargo and passenger flight arc $f \in \mathcal{A}^{CF} \cup \mathcal{A}^{PF}$.
$\mathrm{RC}_{f,k}$	This parameter is 1 if the distance of arc $f \in \mathcal{A}^{CF}$ is less than the maximum range of aircraft $k \in \mathcal{K}^{C}$.
$I^{C}_{k,t}$	This parameter is 1 if full-freighter $k \in \mathcal{K}^{C}$ is of type $t \in \mathcal{T}^{C}$.
$I^{R}_{r,k}$	This parameter is 1 if request $r \in \mathcal{R}$ can be loaded on full-freighter $k \in \mathcal{K}^{C}$.
$I^{P}_{r,f}$	This parameter is 1 if request $r \in \mathcal{R}$ can be loaded on the passenger aircraft flying on passenger flight arc $f \in \mathcal{A}^{PF}$.
$\mathrm{TRDEP}_{f,r}$	This parameter is 1 if arc $f \in \mathcal{A}$ departs at or after the release time of request $r \in \mathcal{R}$.
$\mathrm{TRARR}_{f,r}$	This parameter is 1 if arc $f \in \mathcal{A}$ arrives at or before the due-time of request $r \in \mathcal{R}$.
W_r	This weight of cargo request $r \in \mathcal{R}$.
$\mathrm{Cap}_{f,k}$	This parameter denotes cargo carrying capacity of aircraft $k \in \mathcal{K}^{C}$ on arc $f \in \mathcal{A}^{CF}$.
CapPax_f	This parameter denotes cargo carrying capacity on passenger flight arc $f \in \mathcal{A}^{PF}$.
M	A large positive numbers.
δ^{+}_{n},δ^{-}_{n}	The set of all arcs originating (+) and terminating (−) from node $n \in \mathcal{N}$.
N^{+}_{r},N^{-}_{r}	The originating (+) and terminating (−) node in the time space network for request $r \in \mathcal{R}$.

Table 4: Cost Parameters

Parameter	**Description**
$\mathrm{LandingC}_{f,t}$	The landing fees associated with fleet type $t \in \mathcal{T}^{C}$ on cargo flight arc $f \in \mathcal{A}^{CF}$.
FixC_t	The fixed operating cost per distance for fleet type $t \in \mathcal{T}^{C}$.
VarC_t	The variable cost per distance per kg for fleet type $t \in \mathcal{T}^{C}$.
$\mathrm{PaxVarC}_f$	The variable cost per distance per kg for using passenger flight arc $f \in \mathcal{A}^{PF}$.
PC_r	The penalty cost for not delivering request $r \in \mathcal{R}$.

$$
\begin{aligned}
\textbf{min} \quad & \sum_{t\in\mathcal{T}^C}\sum_{f\in\mathcal{A}^{CF}}(\text{LandingC}_{f,t}+(\text{FixC}_t\times\text{Dist}_f))\times y_{f,t} \\
& +\sum_{t\in\mathcal{T}^C}\sum_{f\in\mathcal{A}^{CF}}\text{VarC}_t\times\text{Dist}_f\times u_{f,t}+\sum_{f\in\mathcal{A}^{PF}}\text{PaxVarC}_f\times\text{Dist}_f\times\sum_{r\in\mathcal{R}}W_r q_{f,r} \qquad (1)\\
& +\sum_{r\in\mathcal{R}}\sum_{f\in\mathcal{A}^{NS}}\text{PC}_r\times q_{f,r}
\end{aligned}
$$

s.t

$$x_{f,k}\le\text{RC}_{f,k}\qquad\forall f\in\mathcal{A}^{CF},k\in\mathcal{K}^C \tag{2}$$

$$q_{f,r}\le\text{TRDEP}_{f,r}\times\text{TRARR}_{f,r}\qquad\forall f\in\mathcal{A},r\in\mathcal{R} \tag{3}$$

$$\sum_{k\in\mathcal{K}^C}x_{f,k}\le 1\qquad\forall f\in\mathcal{A}^{CF} \tag{4}$$

$$q_{f,r}\le\text{NS}_{f,r}\qquad\forall f\in\mathcal{A}^{NS},r\in\mathcal{R} \tag{5}$$

$$q_{f,r}\le\sum_{k\in\mathcal{K}^C}I^R_{r,k}x_{f,k}\qquad\forall f\in\mathcal{A}^{CF},r\in\mathcal{R} \tag{6}$$

$$\sum_{r\in\mathcal{R}}W_r q_{f,r}\le\sum_{k\in\mathcal{K}^C}\text{Cap}_{f,k}x_{f,k}\qquad\forall f\in\mathcal{A}^{CF} \tag{7}$$

$$\sum_{r\in\mathcal{R}}I^P_{r,f}\times W_r\times q_{f,r}\le\text{CapPax}_f\qquad\forall f\in\mathcal{A}^{PF} \tag{8}$$

$$y_{f,t}=\sum_{k\in\mathcal{K}^C}I^C_{k,t}x_{f,k}\qquad\forall f\in\mathcal{A}^{CF},t\in\mathcal{T}^C \tag{9}$$

$$u_{f,t}\ge\sum_{r\in\mathcal{R}}W_r q_{f,r}-M\left(1-\sum_{k\in\mathcal{K}^C}I^C_{k,t}x_{f,k}\right)\qquad\forall f\in\mathcal{A}^{CF},t\in\mathcal{T}^C \tag{10}$$

$$\sum_{n\in\mathcal{N}_S}\sum_{f\in\delta^+_n\cap(\mathcal{A}^{CF}\cup\mathcal{A}^G)}x_{f,k}=1\qquad\forall k\in\mathcal{K}^C \tag{11}$$

$$\sum_{n\in\mathcal{N}_E}\sum_{f\in\delta^-_n\cap(\mathcal{A}^{CF}\cup\mathcal{A}^G)}x_{f,k}=1\qquad\forall k\in\mathcal{K}^C \tag{12}$$

$$\sum_{f\in\delta^+_n\cap(\mathcal{A}^{CF}\cup\mathcal{A}^G)}x_{f,k}-\sum_{f\in\delta^-_n\cap(\mathcal{A}^{CF}\cup\mathcal{A}^G)}x_{f,k}=0\qquad\forall n\in\mathcal{N}\setminus(\mathcal{N}_S\cup\mathcal{N}_E),k\in\mathcal{K}^C \tag{13}$$

$$\sum_{f\in\delta^+_n}q_{f,r}-\sum_{f\in\delta^-_n}q_{f,r}=\begin{cases}1 & n=N^+_r\\ -1 & n=N^-_r\\ 0\end{cases}\qquad\forall n\in\mathcal{N},r\in\mathcal{R} \tag{14}$$

$$x_{f,k}\in[0,1]\qquad\forall f\in\mathcal{A}^{CF},k\in\mathcal{K}^C \tag{15}$$

$$q_{f,r}\in[0,1]\qquad\forall f\in\mathcal{A},r\in\mathcal{R} \tag{16}$$

$$u_{f,t}\ge 0\qquad\forall f\in\mathcal{A}^{CF},t\in\mathcal{T}^C \tag{17}$$

$$y_{f,t} \in [0,1] \qquad \forall f \in \mathcal{A}^{CF}, t \in \mathcal{T}^C \tag{18}$$

The objective function shown in Equation (1) has four main components; fixed operating costs for flying full-freighters, variable operating costs for cargo transport on full-freighters, variable operating costs for cargo transport on passenger flights and penalty costs for undelivered cargo requests. In the case of full-freighters, the fixed operating cost refers to the cost of operating the aircraft without any payload. It encompasses expenses such as crew costs, among others. The variable cost corresponds to the additional expenses incurred for transporting cargo on the full-freighter. This is primarily driven by the extra fuel costs. The two main full-freighter cost parameters, FixC_t and VarC_t, are dependent only on the aircraft type $t \in \mathcal{T}^C$. It is possible to make them dependent on both the arc type $f \in \mathcal{A}^{CF}$ and fleet type $t \in \mathcal{T}^C$ in order to accommodate differences among airports and arcs. The landing fees, $\mathrm{LandingC}_{f,t}$, depend on both the cargo flight arc $f \in \mathcal{A}^{CF}$ and the aircraft type $t \in \mathcal{T}^C$. This is because, for example, the cost of using the runway at night is higher than during the day. Since the cost parameters are dependent on the aircraft type $t \in \mathcal{T}^C$, the variables $y_{f,t}$ and $u_{f,t}$ are used in the objective function. Equation (2) ensures that full-freighters can only fly on cargo flight arcs that meet the range constraints. Equation (3) restricts cargo requests to only traverse arcs that originate after they are released and terminate before they are due. Equation (4) ensures that only one full-freighter traverses a cargo flight arc. Equation (5) ensures that a request only uses a no-service arc that is assigned to it. Equation (6) ensures that a cargo request is only assigned to full-freighters that can accommodate it. Equation (7) ensures that the total cargo payload on a cargo flight arc does not exceed the capacity available ($\mathrm{Cap}_{f,k}$). The capacity depends on both the arc $f \in \mathcal{A}^{CF}$ and aircraft $k \in \mathcal{K}^C$. This is because of the payload range trade-off. Equation (8) allows cargo requests to traverse passenger flight arcs if the passenger flight arcs can accommodate the requests and there is capacity available to carry the request. The passenger flight capacity parameter is at the flight level $f \in \mathcal{A}^{PF}$. Another approach could be to define a passenger aircraft type (similar to $t \in \mathcal{T}^C$). However, cargo on a passenger flight will be transported in the belly-hold of the aircraft, given there is space available after loading the passenger luggage. Indeed, it is possible that passenger check-in luggage characteristics vary by route. Hence, attributing the capacity to the flight level is a better approach. Equation (9) assigns a value to $y_{f,t}$. Equation (10) assigns a value to $u_{f,t}$. If there is no aircraft $k \in \mathcal{K}$ of type $t \in \mathcal{T}^C$ traversing the cargo flight arc, then the term within the brackets simplifies to 1. Due to the large M, the right hand side for $u_{f,t}$ will be a large negative number, this will mean that $u_{f,t}$ gets a value of 0 (due to its lower bound). If there is an aircraft of type $t \in \mathcal{T}^C$ flying on the arc, then the term within the brackets will be equal to 0. Thus, $u_{f,t}$ will take on a value equal to the total weight of the cargo request traversing the arc. Note that constraint Equation (4) ensures that the term within the brackets of Equation (10) can only be 0 or 1. This is because there can at most be only one full-freighter flying on a cargo flight arc. As previously stated, Equation (7) guarantees that the payload weight does not exceed the capacity limit. Hence, the value of M is chosen to be the maximum payload capacity among all fleet types $t \in \mathcal{T}^C$. Equations (11) and (12) ensure that each full-freighter has a starting and ending node in the time space network. Each full-freighter must traverse a cargo flight arc or ground arc emanating from one of the starting nodes ($\mathcal{N}_S$). Similarly, each full-freighter must traverse a cargo flight arc or ground arc terminating at one of the ending nodes ($\mathcal{N}_E$). The model is given the freedom to decide the best choices for the starting and ending node for each aircraft. Equation (13) ensures there is full-freighter flow conservation for all the intermediate nodes. Equation (14) enforces the flow conservation of the cargo requests. Unlike in the case for full-freighters, the cargo requests do have predefined starting and ending nodes. This is not determined by the model. Finally, Equations (15) to (18) define the nature of the decision variables.

3.3 Request Clustered Detour Ratio Reduced Model (CD-ABM)

This section introduces two modifications that can be implemented to decrease the runtime of the F-ABM model. Section 3.3.1 outlines how cargo requests can be grouped together using clustering. Section 3.3.2 looks at employing a detour ratio approach to reduce the problem size. Section 3.3.3 presents a flow diagram for a model that incorporates the two modifications.

3.3.1 Cargo Request Clustering

Cargo requests that have similar release and due times can be grouped together. This reduces the problem size by decreasing the number of individual cargo requests present in the model. A hierarchical clustering model is used to perform such a clustering. The clustering algorithm is applied to cargo requests that share the same origin-destination pair, this ensures that requests with different routes are not clustered together. The clustering process relies on a distance metric, this quantifies the dissimilarity between two cargo requests.

Figure 1 shows the release and due times of two cargo requests. The first request, indicated by the blue color, has a release time of t_a and a due time of t_b. Similarly, the second request, represented by the red color, has a release time of t_c and a due time of t_d. To assess the dissimilarity between these requests, a distance metric is used. The distance metric is defined as $|(t_c - t_a)| + |(t_d - t_b)|$. This quantifies, in minutes, the total time wasted if these two cargo requests were to be clustered together.

There are cases where certain requests cannot be clustered together, primarily due to two reasons. Firstly, if two requests have non-overlapping time windows, they cannot be put into the same cluster. Secondly, even if the time windows overlap, there may not be sufficient time available to transport the request from its origin to the destination within that window. For instance, if the overlapping time window is only 2 hours, but the transportation time from the origin to the destination is 5 hours, these requests cannot be clustered together. A safety factor of 1.5 is used when determining this. This means that the overlapping time window must be at least 1.5 times longer than the minimum required transportation time from the origin to the destination for the requests to be allowed to be in the same cluster. Requests that cannot be in the same cluster are given a large distance metric of 1000000.

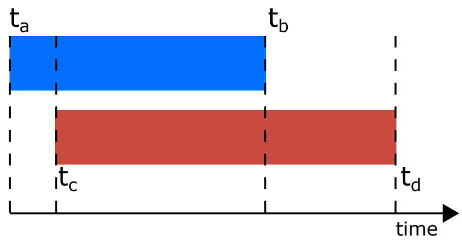


Figure 1: Distance Metric Calculation

Once the distance metric has been constructed, hierarchical clustering using complete linkage (i.e., farthest neighbor) is applied. The complete linkage method ensures that if two clusters contain requests that cannot be clustered together (with a distance metric of 1000000), then those clusters cannot be combined either. The outcome is a dendrogram, which visualizes the hierarchical relationship between the cargo requests. The next step involves pruning or cutting the dendrogram to create the final clusters. The optimal pruning distance is determined by evaluating the silhouette scores for each possible pruning distance. Silhouette scores measure the performance of a particular clustering outcome by comparing the intra-cluster distance (distance within clusters) with the inter-cluster distance (distance between clusters) [Rousseeuw, 1987]. The scores range from -1 to 1, with higher scores indicating better clustering. The pruning distance with the highest silhouette score is used to create the clusters.

When performing clustering, two considerations need to be taken into account. Firstly, the total weight of a cluster, which is the sum of the weights of its individual requests, may exceed the payload capacity of the full-freighter fleet. In such cases, the cluster needs to be divided into smaller subclusters. To accomplish this, the k-means clustering algorithm is applied. For instance, if the payload capacity is 100 tons but the cluster weight is 150 tons, the cluster is divided using $k = 2$. If the resulting subcluster still exceeds the weight limit, further breakdown is performed based on the individual sample silhouette scores of the requests within the cluster. In essence, the subcluster is broken down by selecting only the best requests while ensuring that the total weight remains below the payload capacity.

Another consideration is the compatibility of different cargo request types. Cargo requests can vary in their type, such as being either pallets or containers. Additionally, within these categories, there can be further variations in the types of pallets and containers. For this particular research, which focuses on a timeframe approximately a few months prior to operations, only two types of cargo requests are considered: containers and pallets. Containers are smaller in size, while pallets are larger. It is assumed that all containers can be accommodated on any aircraft, while pallets can only be loaded onto widebody aircraft. Consequently, during the clustering process, if a cluster contains a pallet cargo request, the entire cluster is treated as a pallet. In scenarios where the day of operations is much closer, there may be more information available on the exact types of cargo requests. In this case, it is possible to include the cargo request type as a factor when creating the clusters. Instead of breaking down the cargo requests solely based on the origin-destination pair and then performing the clustering, the breakdown can be done on both the origin-destination pair and the cargo request type.

Once the request clusters have been determined, it is necessary to establish the attributes associated with individual cargo requests to cargo request clusters. Specifically, the request cluster's release time is defined as the latest release time among all the requests that form the cluster. Similarly, the request cluster's due time is determined as the earliest due time among all the requests within the cluster. Additionally, the weight of the cargo request cluster is equivalent to the combined weight of all the requests that constitute it.

3.3.2 Detour Ratio Reduction

The concept of the detour ratio reduction is rather straightforward. When examining the route of a cargo request, such as from London to New York, it is unnecessary for the request to pass through unrelated airports like Beijing. Consequently, there is no need to use arcs that originate or terminate in Beijing for that specific request. More generally, the shortest distance between the origin and destination is calculated for each OD pair. Subsequently, a list of all feasible airport paths (i.e. list of airports) that start at the origin and end at the destination is computed. A path is deemed feasible if it can be flown by at least one aircraft, while adhering to range constraints. In cases where the network is extensive, the total number of paths can be very large. Hence, only paths with a maximum of two transshipment points (equivalent to four airports) are considered. The total distance of each path is computed. The detour ratio is obtained by dividing the distance of the path by the shortest possible distance for that OD pair. Paths with a detour ratio below a predetermined threshold (e.g. 2) are kept for further analysis. A list of all airports traversed by these paths is generated, representing the airports that a particular request on an OD pair can access. Consequently, a request is limited to traversing arcs that originate and terminate at one of the specified airports. This corresponds to creating $q_{f,r}$ variables only for arcs $f \in \mathcal{A}$ that are eligible to be traversed by the cargo request $r \in \mathcal{R}$.

3.3.3 Model Flow Diagram

The modifications of cargo request clustering and detour reduction are incorporated into the full arc-based model (F-ABM). This model is referred to as the Request **C**lustered **D**etour Ratio Reduced **A**rc **B**ased **M**odel (CD-ABM).Figure 2 shows the flow diagram of this model.

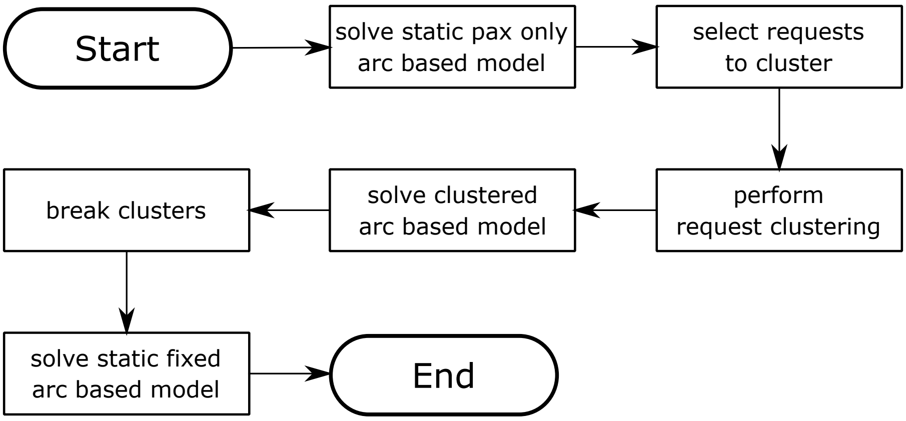


Figure 2: Request Clustered Detour Ratio Reduced Model

First, the detour ratio reduction modification is applied. This helps to decrease the number of $q_{f,r}$ variables. The next step involves solving a static model that exclusively considers passenger flight arcs ($\mathcal{A}^{PF}$). This aims to identify the cargo requests that can be optimally served using passenger flights only. Afterwards, the cargo request clustering is performed on the remaining set of cargo requests that could not be fully accommodated through passenger flight arcs. The requests that could be served by passenger flights are excluded from the clustering process. This results in a new cargo request set, denoted as $\mathcal{R}^C$, which consists of a reduced number of cargo requests compared to the original set, $\mathcal{R}$, due to the clustering. The full arc based model is then solved using $\mathcal{R}^C$, instead of $\mathcal{R}$. This results in a shorter runtime since the model is smaller (due to a smaller cargo request set $\mathcal{R}^C$). The output of this model is a full-freighter routing and scheduling solution. Lastly, a static full arc based model is solved using the original set of requests, $\mathcal{R}$. This model passenger flight arcs and those cargo flight arcs that were part of the full-freighter routing solution. The purpose of this step is to address cases where a cargo request cluster as a whole cannot be accommodated within the defined cargo flight routing, but individual requests from the cluster can be accommodated. This may occur if the cluster's total weight exceeds aircraft capacity or if the cluster's release and due times cannot be met collectively but individual requests can be accommodated.

3.4 Column Generation Arcs Reduction Model (D-CGR-ABM)

Out of the many cargo flight arcs, only a handful of them will be used in the final solution. In the case of large problem instances, the sheer number of cargo flight arcs generated can make the problem challenging to solve, resulting in long computational times. An alternative approach could involve selecting a subset of cargo flight arcs to include in the problem and solving the arc based model using only these pre-selected cargo flight arcs. Mathematically, this approach involves creating a set $\mathcal{A}_R^{CF}$ that consists of cargo flight arcs, which is a subset of all the cargo flight arcs ($\mathcal{A}_R^{CF} \subset \mathcal{A}^{CF}$). This process can be iterated multiple times. In each iteration, a subset of cargo flight arcs is selected and the arc based model is solved. Afterwards, based on the current solution, a better subset of cargo flight arcs is selected and the arc based model is solved again. This process can be repeated until a stop condition is reached. The selection of the subset of cargo flight arcs needs to be done in an wise manner. Randomly creating a subset and using it to solve the arc based model will likely not yield satisfactory results.

The proposed approach involves solving a sub-problem to determine the subset of cargo flight arcs. In this sub-problem, the cargo requests are assigned specific paths from their origin node to their destination node. The resulting solution of the sub-problem will be a collection of paths that the cargo requests will follow. The arcs comprising these selected paths will then be used to construct the set $\mathcal{A}_R^{CF}$. There are a potentially large number of paths a request can take from its origin to its destination. Solving the sub-problem with all possible paths is impractical. Instead, column generation can be used to generate good paths for the cargo requests. The problem of generating variables (e.g. paths in this case) is referred to as the pricing problem.

The formulation of the sub-problem is presented below. It uses several decision variables, parameters, and constraints from the original full arc-based model (F-ABM) described in Section 3.2. However, there are also notable differences. The key difference is that cargo requests are assigned paths while aircraft are assigned arcs. The sub-problem only includes cargo flight arcs and ground arcs, excluding passenger flight arcs and no-service arcs. Each request $r \in \mathcal{R}$ has the flexibility to follow a specific path p from the set of possible paths $\mathcal{P}_r$ associated with that request. Instead of the decision variable $q_{f,r}$ indicating whether request $r \in \mathcal{R}$ traverses arc $f \in \mathcal{A}$, a different decision variable is introduced. The binary variable $z_{p,r}$ takes the value of 1 if request $r \in \mathcal{R}$ follows path $p \in \mathcal{P}_r$. In cases where delivery is not possible, the request follows its designated no-service path z_r^{NS}, defined for each request $r \in \mathcal{R}$. Additionally, the set $\mathcal{P}_f$ represents all paths that use cargo flight arc $f \in \mathcal{A}^{CF}$. Cost parameters are assigned for each specific aircraft $k \in \mathcal{K}^C$ rather than aircraft types $t \in \mathcal{T}^C$. This minor adjustment eliminates the need for $y_{f,t}$, reducing the overall model size. Finally, there is only one variable cost associated with transporting the cargo requests on cargo aircraft. This modification helps in the process of implementing column generation. The smallest variable cost across all the aircraft type $t \in \mathcal{T}^C$ is used; it is referred to as "MinVar". The objective function of the sub-problem is shown in Equation (19). The constraints are given in Equations (20) to (29).

$$\begin{aligned}\textbf{min} \quad & \sum_{k \in \mathcal{K}^C} \sum_{f \in \mathcal{A}^{CF}} (\text{LandingC}_{f,k} + (\text{FixC}_k \times \text{Dist}_f)) \times x_{f,k} \\ & + \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}_r} W_r \times \text{MinVar} \times \text{dist}_p \times z_{p,r} + \sum_{r \in \mathcal{R}} \text{PC}_r \times z_r^{NS}\end{aligned} \tag{19}$$

s.t

$$x_{f,k} \leq \text{RC}_{f,k} \qquad \forall f \in \mathcal{A}^{CF}, k \in \mathcal{K}^C \tag{20}$$

$$\sum_{k \in \mathcal{K}^C} x_{f,k} \leq 1 \qquad \forall f \in \mathcal{A}^{CF} \tag{21}$$

$$\sum_{p \in \mathcal{P}_r} z_{p,r} + z_r^{NS} \geq 1 \qquad \forall r \in \mathcal{R} \tag{22}$$

$$\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}_r \cap \mathcal{P}_f} W_r z_{p,r} \leq \sum_{k \in \mathcal{K}^C} \text{Cap}_{f,k} x_{f,k} \qquad \forall f \in \mathcal{A}^{CF} \tag{23}$$

$$\sum_{n \in \mathcal{N}_S} \sum_{f \in \delta_n^+ \cap (\mathcal{A}^{CF} \cup \mathcal{A}^G)} x_{f,k} = 1 \qquad \forall k \in \mathcal{K}^C \tag{24}$$

$$\sum_{n \in \mathcal{N}_E} \sum_{f \in \delta_n^- \cap (\mathcal{A}^{CF} \cup \mathcal{A}^G)} x_{f,k} = 1 \qquad \forall k \in \mathcal{K}^C \tag{25}$$

$$\sum_{f \in \delta_n^+ \cap (\mathcal{A}^{CF} \cup \mathcal{A}^G)} x_{f,k} - \sum_{f \in \delta_n^- \cap (\mathcal{A}^{CF} \cup \mathcal{A}^G)} x_{f,k} = 0 \qquad \forall n \in \mathcal{N} \setminus (\mathcal{N}_S \cup \mathcal{N}_E), k \in \mathcal{K}^C \tag{26}$$

$$x_{f,k} \in [0,1] \qquad \forall f \in \mathcal{A}^{CF}, k \in \mathcal{K}^C \tag{27}$$

$$z_{p,r} \in [0,1] \qquad \forall r \in \mathcal{R}, p \in \mathcal{P}_r \tag{28}$$

$$z_r^{NS} \in [0,1] \qquad \forall r \in \mathcal{R} \tag{29}$$

The objective function, shown in Equation (19), consists of three components: a fixed cost for operating a full-freighter, a variable cost for transporting requests on full-freighters and a penalty cost for not delivering the cargo requests. Many of the constraints are inherited from the full-arc based model. Equation (22) and Equation (23) are new constraints specific to the sub-problem. Equation (22) guarantees that each request $r \in \mathcal{R}$ follows a path. Equation (23) ensures that the weight of cargo requests carried on cargo flight arc $f \in \mathcal{A}^{CF}$ does not exceed the capacity available.

As mentioned, instead of solving the model with all possible paths, column generation is used to generate good paths. First, the binary constraints on the decision variables is relaxed (Equations (27) to (29)). This will allow dual variables to be generated for each constraint. There are two important types of dual variables. Firstly, let σ_r represent a dual variable associated with the constraint shown in Equation (22). Since this constraint is created for each individual request, there exists a dual variable σ_r for each request $r \in \mathcal{R}$. Secondly, let π_f denote a dual variable associated with the constraint shown in Equation (23). Since this constraint is created for each cargo flight arc, there exists a dual variable π_f for each $f \in \mathcal{A}^{CF}$.

The objective of the pricing problem is to identify good paths for the cargo requests. In mathematical terms, this involves finding paths with negative reduced costs, as this would result in a decrease in the overall cost objective. The reduced cost of a path is calculated by subtracting the product of the dual variable associated with the constraint and the coefficient for that constraint from the objective coefficient for that path. For instance, suppose a potential path p' for request r uses cargo flight arcs $\{f_1, f_2, ..., f_n\}$. The objective function coefficient for such this path is $W_r \times \text{MinVar} \times (d_1 + d_2 + ... + d_n)$, where d_f is the total distance of cargo flight arc $f \in \mathcal{A}^{CF}$. Such a path variable would be present in the respective request constraint (Equation (22)) for request r with the coefficient of 1. It will also be present in the arc capacity constraints (Equation (23)) for each arc that it uses in $\{f_1, f_2, ..., f_n\}$. Hence, the reduced cost for this path variable is: $W_r \times \text{MinVar} \times (d_1 + d_2 + ... + d_n) - W_r \sum_{f \in \{f_1, f_2, ..., f_n\}} \pi_f - \sigma_r$. This can be simplified to:

$$\text{Reduced Cost } = W_r \times \text{MinVar} \times \left[\left(d_1 - \frac{\pi_{f1}}{\text{MinVar}} \right) + \left(d_2 - \frac{\pi_{f2}}{\text{MinVar}} \right) + ... + \left(d_n - \frac{\pi_{fn}}{\text{MinVar}} \right) \right] - \sigma_r \quad (30)$$

In essence, the distance of the cargo flight arcs is adjusted by the dual variables from the arc capacity constraints from Equation (23). The aim of the pricing problem is to find the best paths to add (i.e. paths with the most negative reduced cost). This can be achieved by adjusting the distances of the cargo flight arcs with the dual variables and then solving a shortest path problem to find the shortest path. By doing this, the first term in Equation (30) will be as small as possible and the reduced cost will also be as small as possible.

The general procedure is as follows. First, the binary constraints are relaxed for the decision variables and sub-problem is initialized with one path for each cargo request. The sub-problem is solved and the dual values are used to adjust the distances of the cargo flight arcs. Thereafter, an attempt is made to generate a new path for each cargo request. Only paths which have a negative reduced cost are added to the sub-problem. Afterwards, the sub-problem is resolved with the new path additions. Consequently, a second round of pricing is performed with the new dual values from the new solution. This entire process is repeated until there are no more paths that can be added to the sub-problem (because no path has a negative reduced cost).

At this stage, the binary constraints are relaxed in the sub-problem. It is likely that the $x_{f,k}$ decision variables are fractional. The traditional approach would be to impose the binary constraints after the pricing is complete and solve the sub-problem. However, this is not done in this approach. Instead, all aircraft arcs $f \in \mathcal{A}^{CF}$ where $x_{f,k}$ is greater than a particular threshold ($= 0.05$) are selected to form a subset $\mathcal{A}_R^{CF}$. The arc based model is solved using the subset of cargo flight arcs $f \in \mathcal{A}_R^{CF}$, instead of considering all the cargo arcs $f \in \mathcal{A}^{CF}$.

After solving the arc-based model, there may remain undelivered cargo requests. Subsequently, the sub-problem is solved again, this time considering only the undelivered requests, aiming to generate new paths for them. Additionally, the available capacity of the arcs is scaled according to the weight of the undelivered cargo requests. For example, if the total weight of all requests is 100 tons and 25 tons of cargo requests have not been delivered, the arc capacities are scaled down by 0.75 (100 - 25 / 100). The new subset, $\mathcal{A}_R^{CF}$, consists of the arcs from the current solution and the newly selected arcs from the sub-problem. The arc-based model is then solved again using this updated subset of cargo flight arcs. This entire process, involving solving the sub-problem, solving the arc-based model, solving the sub-problem with undelivered requests, and solving the arc-based model again, is referred to as a stage. Multiple stages can be performed. After each stage, the sub-problem is solved again with all the requests. The arc capacities do not need to be adjusted since all the requests are priced again. However, the arc capacities of the arcs in the current solution are adjusted to reflect the remaining capacity left in the current routing and scheduling.

Based on this, the **D**etour Ratio Reduced **C**olumn **G**eneration **R**educed **A**rc **B**ased **M**odel (D-CGR-ABM) is a model that incorporates the detour ratio reduction and the column generation-based selection of a subset of cargo flight arcs, as described above.

3.5 Request Clustered, Detour Reduced, Column Generation Arc Size Reduction (CD-CGR-ABM)

The request clustering, as described in Section 3.3.1, can be combined with the D-CGR-ABM model to give the Request **C**lustered **D**etour Ratio Reduced **C**olumn **G**eneration **R**educed **A**rc **B**ased **M**odel (CD-CGR-ABM). This model essentially incorporates all the modifications discussed earlier, including request clustering, detour ratio reduction, and column generation-based arc selection. Figure 3 shows the flow diagram of this model.

First, the static passenger flights only arc based model is solved to determine which cargo requests should be allocated to these passenger flights. Cargo requests that were not allocated on passenger flights are clustered together. After that, the sub-problem is solved to create a subset of cargo flight arcs $f \in \mathcal{A}_R^{CF}$ that are then used to solve the arc based model. Notice the use of the word "activate" in the Figure 3. Essentially, the arc-based model is initialized with all the cargo flight arcs. Arcs that are not in $\mathcal{A}_R^{CF}$ are deactivated by adding the following constraint:$\sum_{k \in \mathcal{K}^C} x_{f,k} \leq 0$ for all cargo flight arcs $f \in (\mathcal{A}^{CF} - \mathcal{A}_R^{CF})$. If a deactivated arc is to be reactivated, the corresponding constraint is removed.

The sub-problem and the reduced arc based problem are solved until a stop condition is reached. In this case, the model stops after 3 stages are performed. After this, the cargo request clusters are disaggregated into the individual cargo requests. The static arc based model is solved again, with the passenger flights arcs and the cargo flight arcs that are a part of the solution. Also, it should be noted that the detour ratio reduction is applied to all individual arc-based models throughout the process.

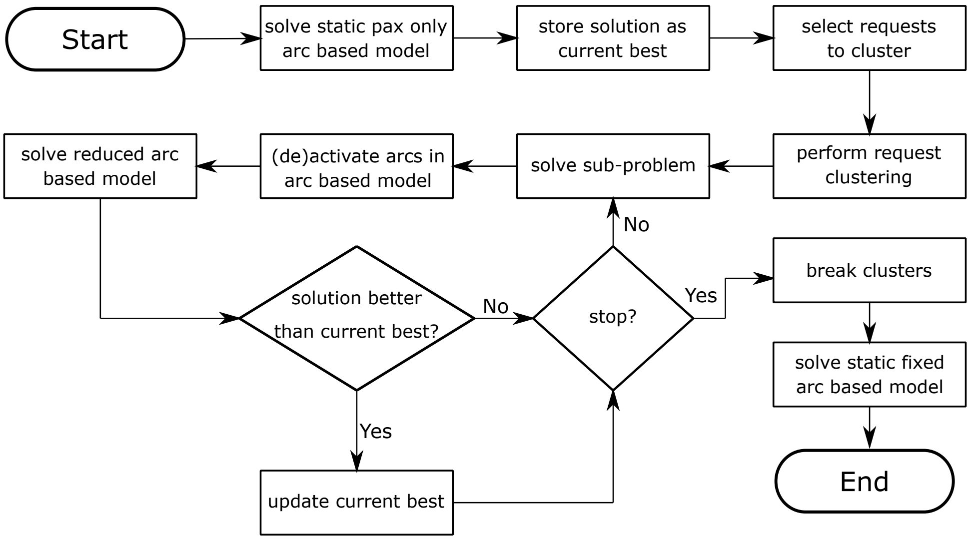


Figure 3: Request Clustered Detour Ratio Column Generation Reduced Model

4 Description of the Case Studies

In the field of cargo operations, a significant portion of data is confidential, making it challenging to access real-world data-sets. To overcome this limitation, a set of 11 synthetic problem instances were created to test the various models on. These instances varied in size, representing a range of scenarios. Each problem instance was characterized by a list of airports, a time window, list of passenger flights, cargo requests, and cargo freighters. Furthermore, to incorporate passenger flights into the instances, a representative airline was selected for each problem instance. Since each representative airline operates from a specific hub, such as Frankfurt (FRA) for Lufthansa, the hub airport was included in the list of airports for each instance.

The instance's list of airports is used to generate all possible permutations of OD (origin-destination) pairs. Each cargo request will be assigned an OD pair. In the analysis of cargo demand, it is observed that the demand for cargo transportation is predominantly unidirectional, with goods typically moving from manufacturing sites to customers. To reflect this, the probability of assigning a specific OD pair to a cargo request is not uniform across all OD pairs. Instead, the probability is determined by trade data, with OD pairs that have high trade exports between their origin and destination countries being given higher probabilities. The trade data is collected from the UN Comtrade database for the year 2019, just before the corona pandemic started [United Nations, Department Of Economic and Social Affairs, 2023].

It is important to note that a significant portion of global trade is moved via maritime transport. For example, countries with high crude oil exports will have high export numbers. However, since crude oil is predominantly transported by ships, there may not be a high demand for air cargo from such countries. Therefore, when analyzing trade data, it becomes important to consider trade classification and focus only on products that are typically shipped by air cargo.

There are many different types of trade classifications in use today. One such classification type is the 2-digit Standard International Trade Classification (SITC) system, which categorizes trade using two digits ranging from 00 to 99 [United Nations, 2006]. When utilizing the UN Comtrade database, trade entries associated with classifications that typically do not involve air cargo transportation are excluded from the dataset. For example, categories like "32 - coal, coke and briquettes" and "35 - electric current" are removed from the dataset.

Finally, the task of assigning release and due times to the cargo requests is addressed. Inspiration is drawn from [Brandt and Nickel, 2019], where an example demonstrates the hourly outgoing cargo capacity of a combination airline throughout a day (refer to Appendix B). This example reveals peak periods of cargo handling and capacity, as well as times of low outgoing capacity. Based on this information, the following approach is adopted for assigning release and due times.

The release and due time of a cargo request must fall within the following time points during the day: 08:00, 13:00, 17:00, and 22:00. These times correspond to the peak cargo handling capacity. It is important to note that these times are in local time, not UTC time. Using these time points, all possible combinations of day and time are created, resulting in every feasible release and due time pair for a cargo request. A pair is considered feasible if the time interval between the release time and due time is 4 hours longer than the minimum travel time required from the origin to the destination. Subsequently, each cargo request is randomly assigned a specific feasible release time and due time pair. In summary, each instance has a predetermined number of cargo requests and types. Each cargo request is assigned an origin-destination (OD) pair, followed by the assignment of a release time and due time pair.

Table 5 summarizes the various instances and provides key information for each instance, including the total number of cargo requests per type, the total number of full freighters, the representative airline, and the total number of passenger flights. All problem instances have a time window of 4 days. In order to gather relevant information about passenger flights, historical data was examined, specifically focusing on past scheduled passenger flights operated by the representative airline between the respective origin-destination (OD) pairs, including details such as the aircraft type used.

Table 5: Overview of Synthetic Instances

Instance	**Airports**	**Total Full-Freighters**	**Total Containers**	**Total Pallets**	**Airline**	**Total Pax. Flights**
1	FRA, JFK, BOM	1	60	70	LH	10
2	IST, LHR, BOM	1	80	90	TK	31
3	DXB, DEL, ATL, SIN, LHR	2	180	195	EK	67
4	CAN, DEL, MNL, NRT, SIN	2	225	210	CZ	5
5	HND, KUL, CGK, MEX, LAX, BOM	3	310	240	NH	9
6	IST, JFK, CDG, FCO, TLV, BOM, PEK	3	355	265	TK	125
7	DXB, ATL, LOS, DEL, AMS, CAN, MNL, HND	4	400	360	EZ	50
8	CAN, ATL, FRA, DOH, BOM, CGK, SGN, HND, SYD	4	395	365	CZ	9
9	CDG, BOS, MAD, LHR, DXB, SIN, BOM, ICN, ADD, MEX	5	610	340	AF	109
10	FRA, CMN, LHR, TLV, JFK, CAN, NRT, LOS, IST, DEL, KUL, GRU	6	600	390	LH	72
11	IST, BOM, SIN, ICN, HND, JFK, AMS, CDG, JNB, LOS, DXB, GRU, BOG, EZE, MEL, BKK	8	830	505	TK	152

The values for the cost parameters, such as FixC_t and VarC_t, are primarily derived from [Van der Meulen et al., 2020]. However, it should be noted that most of the costs provided in the reference are specific to the B747-400ERF aircraft. Since the problem instances in this paper used other types of aircraft as well, some adjustments were made to the cost figures. Appendix C provides detailed information on the adjusted cost figures.

Additionally, it should be noted that different airports have varying methods for calculating landing charges.

However, in this study, a single charge was assumed for all airports. The landing fee calculation was inspired by the method used by Paris Charles de Gaulle Airport (CDG). The landing fee is calculated as $304.38 + 4.251 \times t$, where t represents the Maximum Takeoff Weight (MTOW) of the aircraft in tons [Aeroports De Paris, 2023]. Typically, airports will impose additional charges based on factors such as the time of day and the noise category of the aircraft. In this study, a simplified surcharge is used, where using the runway between 22:00 and 06:00 costs 40% more.

Lastly, a time discretization of 2 hours (i.e. 120 minutes) is used for all the models.

5 Results

This section presents the results of the various models. Section 5.1 gives an illustrative example of a small problem instance and compares the solutions of the various models. Section 5.2 talks about the results of the clustering algorithm. Finally, Section 5.3 compares the optimization results across all the problem instances. Table 6 summarizes the main differences between the models described in Section 3.

Table 6: Overview of Models

Model	Arc Based	Detour Ratio Reduction	Clustered Requests	Column Generation Reduction
F-ABM	Yes	No	No	No
CD-ABM	Yes	Yes	Yes	No
D-CGR-ABM	Yes	Yes	No	Yes
CD-CGR-ABM	Yes	Yes	Yes	Yes

5.1 An Illustrative Example

This section presents an illustrative example, Instance 2, to better explain the problem and the various models used to solve it. This problem instance consists of three airports: **(1)** London Heathrow (LHR) **(2)** Istanbul Airport (IST) **(3)** Mumbai CSMIA (BOM). The time-window spans 4 days from 10-07-2022 21:00 UTC to 14-07-2022 21:00 UTC. The representative airline is Turkish Airlines, and there are a total of 31 passenger flights scheduled during the time window. These passenger flights are between LHR and IST. There are no passenger flights connecting BOM and IST. All the passenger flights are operated by wide-body aircraft. Additionally, there is also one A330-200F full-freighter available for transporting cargo requests. There are a total of 170 cargo requests that need to be transported, out of which 80 are containers and 90 are pallets. The breakdown of the origin-destination pair of the cargo requests is as follows: IST-LHR 42, LHR-IST 37, BOM-LHR 36, LHR-BOM 27, BOM-IST 18 and IST-BOM 10. These cargo requests have different release and due times. The goal is to route and schedule the full-freighter such that the operating costs are minimized.

Figure 4 illustrates the full-freighter routing solution for this particular problem instance when solved using the F-ABM model. It also shows the used passenger flights in the solution. The vertical lines represent the boundaries of each day (in UTC). Additionally, the load factor is displayed alongside the arcs. Given that there are many passenger flights between IST and LHR, the model chooses to exclusively operate the full-freighter between IST and BOM.

The problem instance was solved using the other models as well. Figure 5 depicts the dendrogram of the BOM to LHR cargo requests obtained during the request clustering process for the CD-ABM and CD-CGR-ABM models. While each origin-destination (OD) pair has its own dendrogram, only the one for BOM-LHR is presented here for conciseness. Since there are no passenger flight only connections between BOM and LHR, all of the 36 cargo requests are clustered. The dendrogram can be pruned at various cluster distances, resulting in different numbers of clusters. Each cluster can be evaluated using the silhouette score. Figure 6 depicts the variation of the silhouette score as a function of the total number of clusters. The maximum silhouette score, approximately 0.56, is attained with 7 clusters. This corresponds to a pruning distance of 2280 in the dendrogram. This leads to the formation of 7 cargo request clusters from the 36 original cargo requests.

When using the D-CGR-ABM or CD-CGR-ABM models to solve the problem instance, only a portion of the cargo flight arcs is chosen for each stage. This selection is done by solving a sub-problem using column generation. For example, Figure 7 shows the progression of the objective value of the relaxed sub-problem when solved using column generation in the D-CGR-ABM model. As more paths for cargo requests are generated in the pricing rounds, the objective value (i.e., cost) decreases. It is important to note that the sub-problem is relaxed, among other things, resulting in a lower cost value compared to the actual cost. When using the D-CGR-ABM model, the following fractions of cargo flight arcs are selected for each stage: Stage 1: [14%, 19%],

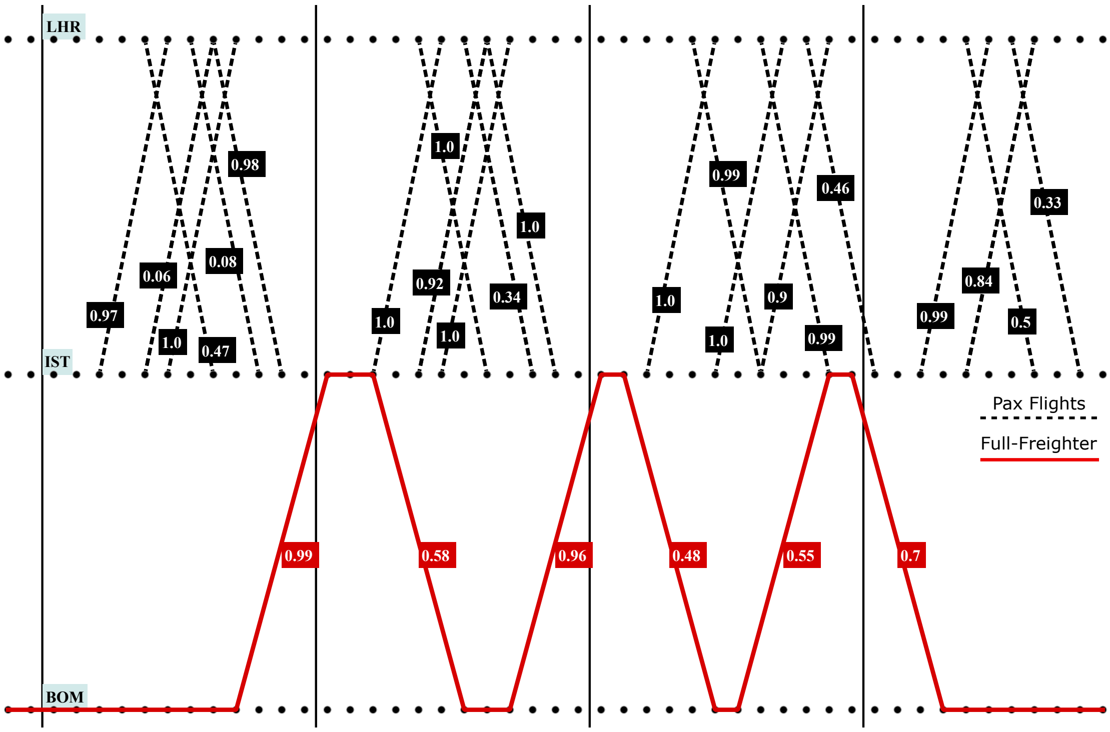


Figure 4: Routing Solution F-ABM Model

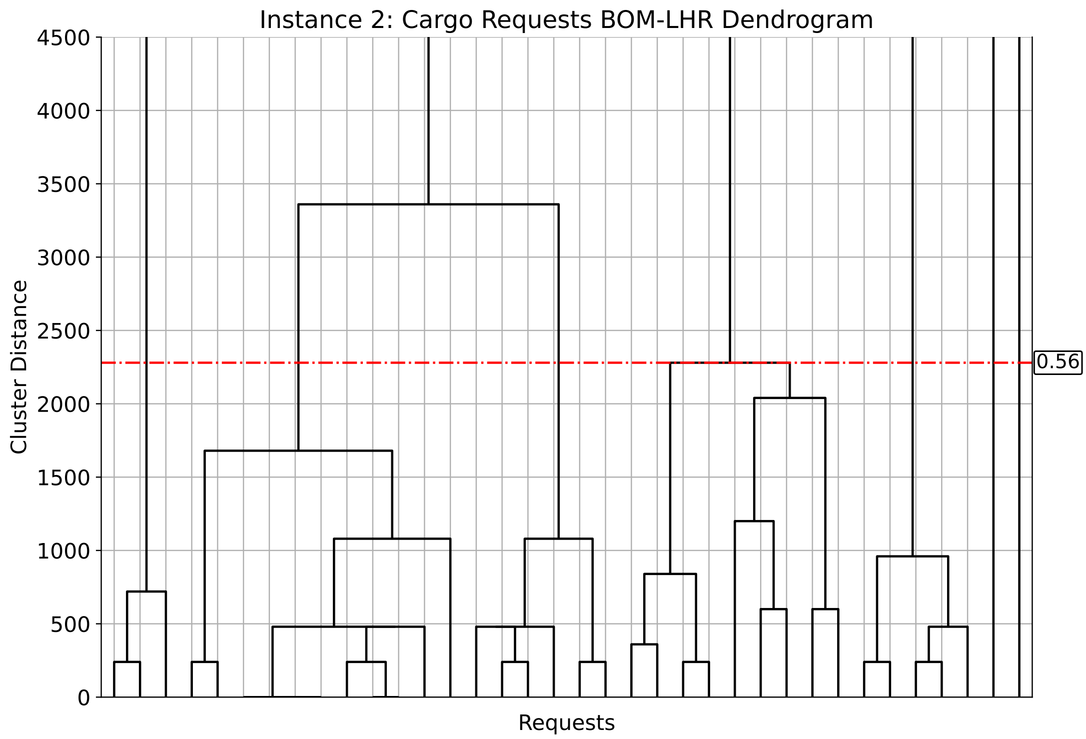


Figure 5: Example Dendrogram Showing Hierarchical Clustering

Stage 2: [29%, 32%], Stage 3: [29%, 31%]. As detailed in Section 3.4, during each stage, the arc-based model is solved twice, using different subsets $\mathcal{A}_R^{CF}$ of the cargo flight arc $\mathcal{A}^{CF}$s.

Table 7 presents a comparison of the results of the various models. As anticipated, the F-ABM model gives the lowest cost, followed by the D-CGR-ABM model in second place, the CD-ABM model in third place, and finally the CD-CGR-ABM model. It is interesting to note that the models which use cargo request clustering (i.e. CD-ABM and CD-CGR-ABM) provide solutions in which the full-freighter does not exclusively operate between IST and BOM (refer to Appendix D). It is somewhat expected that the CD-CGR-ABM model would provide the highest cost solution, as it incorporates the most modifications. These modifications reduce the

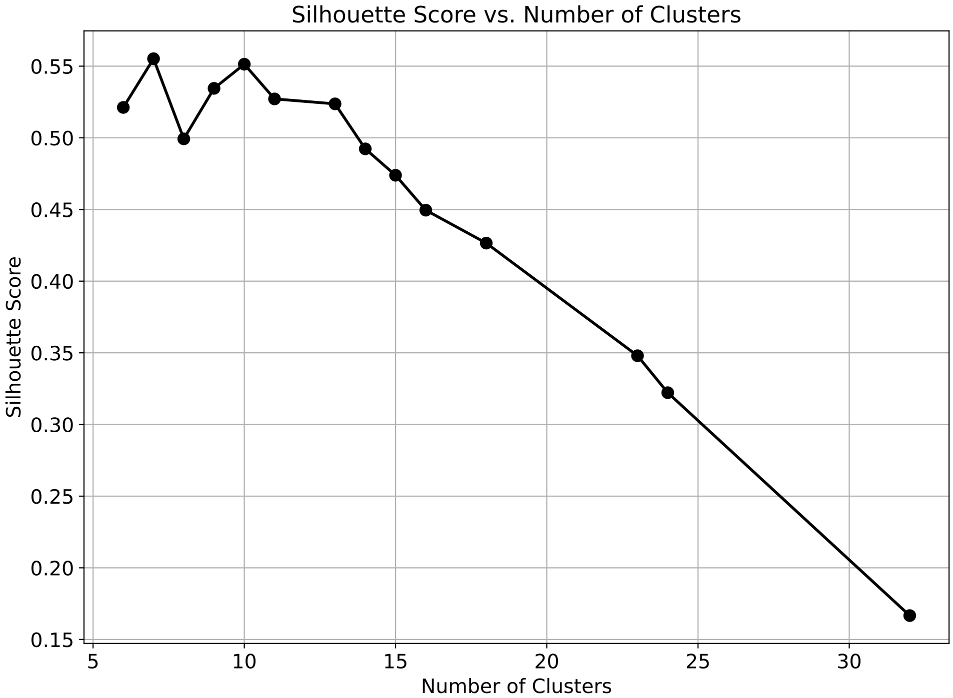


Figure 6: Silhouette Score vs. Number of Clusters

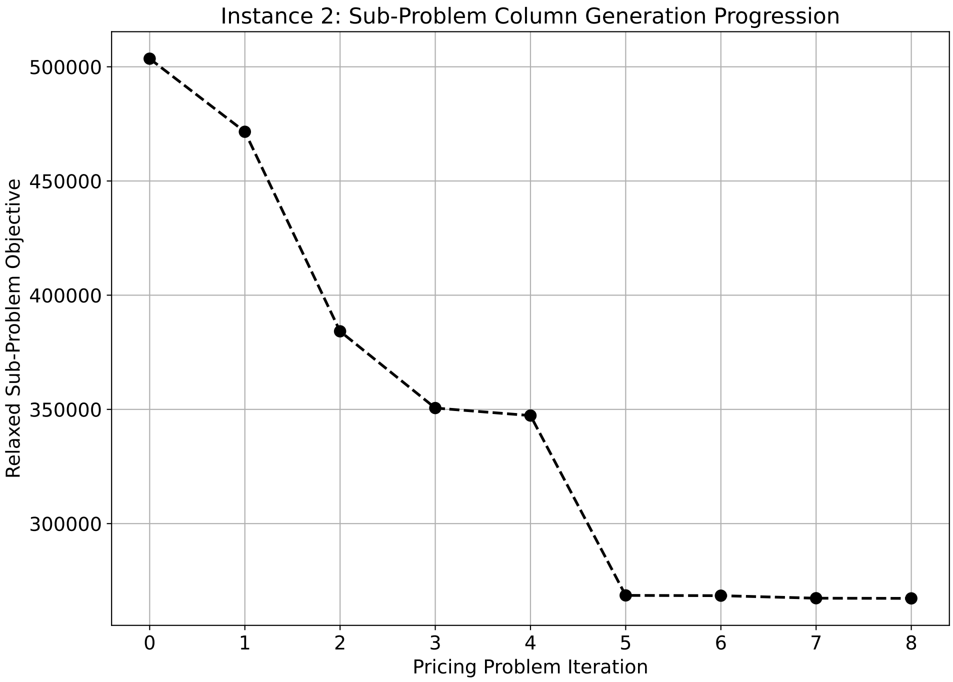


Figure 7: Objective Value Progression of Sub-Problem for Instance 2 with D-CGR-ABM

problem size, but lead to a higher cost. Additionally, the column generation modified models (D-CGR-ABM and CD-CGR-ABM) have longer runtimes compared to the F-ABM model. While this may seem counterintuitive, it is reasonable considering the relatively small size of the problem instance. Referring back to the concept of "stages", the column generation models solve the reduced arc-based model multiple times using different subsets of cargo flight arcs. For small instances, this can be considered an overkill, it is easier to simply solve the F-ABM model. However, for larger instances, the column generation models do reduce the runtime (see Section 5.3). Additionally, the F-ABM model delivers the highest number of requests and achieves the highest load factor for the full-freighter. This likely results in the relatively lower cost.

Table 7: Results Comparison of Various Models

	Cost (EUR)	Cost vs. F-ABM Cost	Time (sec)	Requests Not Delivered	Average Full-Freighter Load Factor
F-ABM	609,941	-	21.00	22	0.71
CD-ABM	687,766	13%	0.24	33	0.57
D-CGR-ABM	633,775	4%	74.97	24	0.69
CG-CGR-ABM	691,343	13%	50.70	34	0.55

5.2 Clustering Algorithm Performance

Table 8 presents the performance of the clustering algorithm. As the instances grow in size, the clustering time also increases. The compression ratio, defined as the ratio of total cargo requests to the total number of clusters, has an average value of 2.11 across all the instances. This indicates that the clustered request set $\mathcal{R}^C$ contains approximately half the number of elements as compared to the original set $\mathcal{R}$.

Table 8: Clustering Algorithm Performance

Instance	Cluster Time (sec)	Total Cargo Requests	Total Clusters	Compression Ratio
1	0.06	130	63	2.06
2	0.06	170	100	1.70
3	0.12	375	178	2.11
4	0.15	435	142	3.06
5	0.13	550	293	1.88
6	0.19	620	301	2.06
7	0.23	760	357	2.13
8	0.27	760	279	2.72
9	0.27	950	472	2.01
10	0.35	990	391	2.53
11	0.50	1335	564	2.37

5.3 Comparison of Models

This section presents a comparison of the results achieved by the various models. Tables 9 to 12 provide the optimization results for the F-ABM, CD-ABM, D-CGR-ABM, and CD-CGR-ABM models, respectively. All the experiments were performed on a computer with an Intel i5-10400 CPU (2.90GHz) processor and 32 GB of RAM. Python and Gurobi Optimizer [Gurobi Optimization, LLC, 2023] are the main tools used to code and run the models. Table 9 displays the optimization results for the F-ABM model, with 6 columns indicating the problem instance, the best incumbent achieved, the best bound, optimality gap, the time to the best incumbent, and the total optimization time. When solving the F-ABM model, two stop conditions were defined: (1) termination if the optimality gap reaches 2% and (2) termination upon reaching a specified time limit. The time limits varied depending on the problem instance, with durations set at 3 hours (18000 seconds) for instances 1-4, 6 hours (21600 seconds) for instances 4-8, and 7 hours (25200 seconds) for instances 9-11. Instance 3 and from instance 6 onwards, all optimization runs were terminated due to reaching the time limit. Instances 6 and 8-11 still had significant optimality gaps at termination, which is reasonable considering their relatively larger sizes. The F-ABM model will require more time to close the gap. Notably, for Instance 11, the F-ABM model was unable to establish a bound, and the best incumbent reported in the table represents a scenario where all cargo requests are solely transported using scheduled passenger flights. This also explains the short time to best.

Table 10 presents the optimization results for the CD-ABM model, with most columns being similar to Table 9. The time columns, namely "Best Time" and "Total Time", include the time required for cargo request clustering. Notably, the optimality gap in Table 10 is significantly smaller compared to Table 9. This gap specifically represents the optimality gap after solving the static fixed arc-based model (refer to Figure 2), which only considers the cargo flight arcs selected during the model's clustering phase. This is different than the F-ABM model, which incorporates all cargo flight arcs.Table 10 also includes the percentage cost increase

Table 9: Results F-ABM Model

Instance	Incumbent	BestBd	Gap	Best Time	Total Time
1	857,140	842,905	2%	7.00	10.88
2	609,941	598,709	2%	21.00	40.25
3	2,491,659	2,423,899	3%	15,913.00	18,000.00
4	1,708,042	1,678,575	2%	5,736.00	5,736.30
5	2,760,092	2,705,425	2%	2,455.00	20,201.41
6	4,267,309	2,882,836	32%	21,059.00	21,600.00
7	5,272,735	5,093,491	3%	14,359.00	21,600.00
8	6,423,022	5,392,912	16%	20,762.00	21,600.00
9	8,178,699	2,485,659	70%	6,346.00	25,200.00
10	9,109,436	4,393,838	52%	19,044.00	25,200.00
11	13,791,271	-		18.00	25,200.00

when using the CD-ABM model as compared to the F-ABM model. The time limits for the clustered arc based model remain the same as those of the F-ABM model (see Figure 2). The total runtime of the CD-ABM model is shorter compared to the F-ABM model; this is expected given the smaller model size. However, the solutions are more costly for instances 1-5 and instance 7. For the remaining instances, the CD-ABM model gives lower-cost solutions. This observation aligns with the understanding that the F-ABM model likely ran out of time, resulting in a inferior solution as compared to the CD-ABM model.

Table 10: Results CD-ABM Model

Instance	Incumbent	BestBd	Gap	Best Time	Total Time	% Cost Increase vs F-ABM
1	890,440	889,636	0.09%	0.24	0.75	3.89%
2	687,766	687,389	0.05%	0.24	0.58	12.76%
3	2,542,522	2,530,243	0.48%	48.95	50.85	2.04%
4	2,048,701	2,048,701	0.00%	36.73	37.75	19.94%
5	2,841,379	2,829,786	0.41%	142.75	143.71	2.95%
6	3,645,015	3,639,515	0.15%	16,662.55	21,603.67	-14.58%
7	5,297,374	5,274,997	0.42%	927.05	1,199.67	0.47%
8	6,140,154	6,120,832	0.31%	2,856.30	2,856.53	-4.40%
9	4,060,504	4,057,762	0.07%	11,115.02	25,205.12	-50.35%
10	6,393,593	6,387,220	0.10%	8,577.28	8,577.45	-29.81%
11	13,412,539	13,350,209	0.46%	22,012.27	25,215.40	-2.75%

Table 11 gives the optimization results for the D-CGR-ABM model. Recall the idea of "stages" described in Section 3.4. Each stage consists of solving the sub-problem, solving the reduced arc-based model, solving the sub-problem with undelivered requests, and solving the reduced arc-based model again. Table 11 presents the optimization results for the first stage and the best optimization results achieved after 3 stages. The choice of 3 stages serves as a stop condition for the model. As expected, the solution improves when comparing Stage 1 to the Best Stage (i.e. Stage 3). Additionally, the percentage cost increase when using the D-CGR-ABM model compared to the F-ABM model is shown. Similar to the CD-ABM model, the F-ABM model gives superior solutions for smaller instances, while the D-CGR-ABM model excels for larger instances.

Table 12 showcases the results for the CD-CGR-ABM model, with columns matching those in Table 11. Also, the time taken to perform the clustering is included in the time columns.

Table 13 compares the cost and time to reach the best solution among the various models with that of the F-ABM model. Several nuanced observations can be made from the results. In general, the models with the

Table 11: Results D-CGR-ABM Model

					% Cost Increase vs F-ABM	
Instance	**1st Stage**	**1st Stage Time**	**Best**	**Best Time**	**1st Stage**	**Best**
1	881,710	5.00	881,710	15.08	2.87%	2.87%
2	650,128	4.55	633,775	74.97	6.59%	3.91%
3	2,528,605	480.62	2,527,719	1,128.36	1.48%	1.45%
4	2,073,539	59.68	1,907,949	618.92	21.40%	11.70%
5	2,790,756	64.70	2,790,615	175.53	1.11%	1.11%
6	3,746,012	3,576.55	3,575,650	8,565.25	-12.22%	-16.21%
7	5,317,914	571.93	5,261,868	1,985.14	0.86%	-0.21%
8	6,037,139	849.52	5,942,921	1,922.76	-6.01%	-7.47%
9	4,196,279	3,253.53	4,099,080	9,101.63	-48.69%	-49.88%
10	6,679,456	5,601.73	6,597,519	17,810.72	-26.68%	-27.57%
11	11,605,746	6,403.37	10,108,789	33,230.91	-15.85%	-26.70%

Table 12: Results CD-CGR-ABM

					% Cost Increase vs F-ABM	
Instance	**1st Stage**	**1st Stage Time**	**Best**	**Best Time**	**1st Stage**	**Best**
1	885,550	2.51	885,550	16.82	3.31%	3.31%
2	1,050,872	3.54	691,343	50.70	72.29%	13.35%
3	3,276,209	7.75	2,606,830	96.62	31.49%	4.62%
4	3,328,630	14.68	2,033,268	282.39	94.88%	19.04%
5	3,705,451	31.19	2,916,050	295.55	34.25%	5.65%
6	4,604,790	130.45	3,609,040	1,995.69	7.91%	-15.43%
7	6,336,853	96.96	5,373,075	1,078.03	20.18%	1.90%
8	6,470,484	62.94	6,106,053	485.32	0.74%	-4.93%
9	5,735,579	81.25	4,449,227	1,296.28	-29.87%	-45.60%
10	7,373,784	257.47	6,679,743	2,255.47	-19.05%	-26.67%
11	9,814,495	452.24	9,545,617	4,946.13	-28.84%	-30.79%

modifications (i.e. CD-ABM, D-CGR-ABM, CD-CGR-ABM) have a shorter time to best as compared to the F-ABM model. However, there are a few exceptions to this trend. For instance, in cases of instances 1 and 2, the column generation models (for best stage) have longer runtimes compared to the F-ABM model. This can be attributed to the small size of these instances, enabling the F-ABM model to quickly find the optimal solution, while the column generation models require solving the sub-problem and the reduced arc-based model multiple times.

Theoretically, the CD-CGR-ABM model uses a subset of cargo flight arcs, while the CD-ABM model uses all the cargo flight arcs. Therefore, the cost of a solution given by the CD-ABM model can never be worse than that given by the CD-CGR-ABM model. Note that both of these models perform cargo request clustering. However, there are instances where the cost of the CD-ABM model is negligibly worse than that of the CD-CGR-ABM model. For example, a difference of 0.85 percentage points for Instance 6 (when comparing with the F-ABM model). This is likely because the clustered arc-based model part of the CD-ABM model (see Figure 2) has a stop condition if the optimality gap is 2%. As a result, the CD-CGR-ABM model may find a slightly better solution in certain cases if it happens to select better cargo flight arcs.

For the column generation models, D-CGR-ABM and CD-CGR-ABM, it is evident that the best stage has a lower cost compared to the first stage. However, the difference in costs between these stages is not consistent across all instances. In certain cases, such as instance 3 for D-CGR-ABM, the difference in cost is relatively small (= 0.03 percentage points). Therefore, one could choose to only perform 1 stage and save time. However, there are instances, like instance 5 for CD-CGR-ABM, where the difference in cost is significant (= 28.6 percentage points). Thus, here it makes sense to do more than 1 stage. This tradeoff highlights that while performing more stages improves the objective, it may not always be worth it as it also costs more time.

Table 13: Results Comparison with F-ABM Model

Instance	CD-ABM		D-CGR-ABM				CD-CGR-ABM			
			1st Stage		Best		1st Stage		Best	
	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
1	3.89%	-97%	2.87%	-29%	2.87%	115%	3.31%	-64%	3.31%	140%
2	12.76%	-99%	6.59%	-78%	3.91%	257%	72.29%	-83%	13.35%	141%
3	2.04%	-100%	1.48%	-97%	1.45%	-93%	31.49%	-100%	4.62%	-99%
4	19.94%	-99%	21.40%	-99%	11.70%	-89%	94.88%	-100%	19.04%	-95%
5	2.95%	-94%	1.11%	-97%	1.11%	-93%	34.25%	-99%	5.65%	-88%
6	-14.58%	-21%	-12.22%	-83%	-16.21%	-59%	7.91%	-99%	-15.43%	-91%
7	0.47%	-94%	0.86%	-96%	-0.21%	-86%	20.18%	-99%	1.90%	-92%
8	-4.40%	-86%	-6.01%	-96%	-7.47%	-91%	0.74%	-100%	-4.93%	-98%
9	-50.35%	75%	-48.69%	-49%	-49.88%	43%	-29.87%	-99%	-45.60%	-80%
10	-29.81%	-55%	-26.68%	-71%	-27.57%	-6%	-19.05%	-99%	-26.67%	-88%
11	-2.75%		-15.85%		-26.70%		-28.84%		-30.79%	

The question of which model is best is difficult to answer as it primarily depends on the problem instance being solved. For very small instances, the F-ABM model tends to provide the best results within a reasonable timeframe. However, for larger instances, both the CD-ABM model and the D-CGR-ABM model are viable options. In most cases, the D-CGR-ABM model creates solutions with lower costs. However, it is important to note that the CD-ABM model generally has shorter runtimes compared to the D-CGR-ABM model. When dealing with larger instances and the need for faster results, the CD-CGR-ABM model proves to be an efficient choice. Although it may not necessarily provide the lowest cost solution, it offers a shorter runtime. In fact, in the case of the largest instance (i.e., Instance 11), the CD-CGR-ABM did provide the lowest-cost solution. This is likely because the instance was so large that the F-ABM, CD-ABM, and D-CGR-ABM models ran out of time.

As with the illustrative example in Section 5.1 (i.e., Instance 2), numerous managerial insights can also be drawn from the solutions for each problem instance. As mentioned, one of the key features of combination carriers is their ability to leverage both the qualities of full-freighters and passenger flights when transporting cargo. For example, in the case of the illustrative example in Section 5.1, there are 50 cargo requests that have a multi-leg journey, using both full-freighters and passenger flights to reach their destination in the F-ABM model. Similarly, for Instance 9, which also has a significant number of passenger flights, there are 125 cargo requests that rely on both full-freighters and the belly-space of passenger flights for transportation in the CD-ABM model. This demonstrates how the decision tool can create synergies between full-freighters and scheduled passenger flights when designing a routing and scheduling solution for full-freighters.

6 Conclusion & Future Work

The air cargo industry plays a crucial role in global supply chains, facilitating the efficient transportation of valuable goods across the world within short timeframes. Combination carriers, which use both dedicated full-freighter aircraft and the belly space of scheduled passenger flights, play a significant role in the air cargo sector.

This research introduces a decision tool designed to assist combination carriers in optimizing their operations. The decision tool uses anticipated cargo demand data at the Unit Load Device (ULD) level as an input to create a routing and scheduling solution for full-freighters, while taking into consideration the available belly space on already scheduled passenger flights. The cargo requests can be transported on full-freighters, scheduled passenger flights, or a combination of both (in the case of multiple legs). The decision tool uses a time space network design to create a full arc based model (F-ABM). Given the long runtimes of the F-ABM model, several modifications are introduced to reduce the model size. The CD-ABM model incorporates cargo request clustering and detour ratio reduction techniques to address this. The D-CGR-ABM model also employs detour ratio reduction and also uses a column generation-based approach to select a subset of cargo flight arcs. Finally, the CD-CGR-ABM model combines all the aforementioned modifications into a single model, integrating detour ratio reduction, cargo request clustering, and column generation for the selection of cargo flight arcs. In essence, the choice of which model to use is left to the user of the decision tool.

The various models are tested on 11 synthetic problem instances of varying sizes to assess their performance. For small instances, the F-ABM model provides the best solution in terms of cost, despite its long runtime. However, for larger instances, the F-ABM model often runs out of time and finds inferior solutions as compared to the other models. Among the other models, the D-CGR-ABM model generally generates superior solutions compared to the CD-ABM model. However, its runtime is often longer than that of the CD-ABM model, although still shorter than the runtime of the F-ABM model. On the other hand, the CD-CGR-ABM model exhibits the shortest runtime, but its solutions tend to have higher costs.

The ultimate choice of which model to use ultimately depends on the user of the proposed decision tool. Opting for the F-ABM model for small instances, while employing the CD-ABM or D-CGR-ABM models for larger instances, is likely the most suitable approach. In the case of very large instances, the CD-CGR-ABM model could be particularly useful. Furthermore, if there are severe runtime restrictions, using the CD-CGR-ABM model for all instances will suffice and give satisfactory solutions.

The decision tool generates solutions that create synergies between full-freighters and scheduled passenger flights. There are numerous examples of cases where a cargo request is transported using both a full-freighter and a scheduled passenger flight to reach its destination.

This decision tool has multiple potential use cases. In the short term (a few months before operations), combination carriers can use it to optimize full-freighter routing and scheduling based on anticipated cargo demand. Furthermore, this tool can provide insights into which airports are best served by full-freighters, considering the already scheduled passenger flights available for cargo transportation. Some airports may not need to be served by full-freighters, this can have an impact on the required ground personnel at those airports. This was observed in a three-airport illustrative example, where the two airports that were well-connected by passenger flights did not have any full-freighters scheduled between them. In a more long term setting, the tool can simulate various future scenarios to support strategic decisions, such as fleet planning. For example, it can help combination carriers determine if their current full-freighter fleet composition is sufficient. If it turns out that a significant number of cargo requests are not being delivered with the current full-freighter fleet composition, it may suggest that investing in an additional full-freighter would be a worthwhile investment.

There are certain limitations to the research conducted that warrant discussion. It is important to address these limitations and explore avenues for future research to address them.

First, the various models should be tested on a real life dataset. An effort was made to generate synthetic problem instances that incorporate many real-life characteristics. For instance, trade data was used to generate asymmetric demand between airports, and real-life passenger flight information was also used. However, it is possible that certain characteristics were not fully accounted for. Therefore, it is essential to test the various models on real-life problem instances. To accomplish this, collaboration with a combination carrier would likely be necessary to access their datasets.

Second, in this research, the various models were tested on 11 different synthetic instances. While general trends could be observed in the results across the models, the decision tool needs to be tested on a greater variety of instances. For example, one instance may have many cargo requests and a long time window, but fewer full-freighters. In another case, there may be a large number of full-freighters and a short time window. It is possible that certain modifications (e.g. cargo clustering, column generation) perform better on specific types of instances. Once the models have been tested on a broader range of instances, a classifier can be trained to determine the most suitable modification to apply for a given new instance.

Also, a more involved use of machine learning can also be explored. In this research, a subset of cargo flight arcs was created using column generation. However, an alternative approach could involve employing supervised machine learning techniques. A classifier can be trained to determine whether a given cargo flight arc is useful or not for routing decisions. This classifier should be trained on real-life datasets. It is likely that training such a classifier will require a lot of time. However, once trained, the classification process should be quite quick. The performance of such a classifier can be compared with that of column generation.

Also, while it has been shown that using column generation to select cargo flight arcs for an arc-based model can be a useful approach to reduce the problem size, the exact algorithm of the model needs to be fine-tuned. In this research, the concept of "stages" was introduced, but there may be potentially better approaches to consider.

Finally, stochasticity can be incorporated into the model to account for uncertainties in operations. For instance, the availability of belly space in scheduled passenger flights and the release times of cargo requests is typically uncertain before operations. Therefore, they can be modeled as random variables. A 2-stage stochastic programming approach can be used to solve such a problem. In the first stage, the routing and scheduling of the full-freighters can be performed. In the second stage, the allocation of cargo requests to the flights, including both passenger and full-freighter flights, can be carried out. The decision tool can then provide solutions that are optimal in a certain percentage of cases, such as 95%. It is important to note that a solution optimal in 99% of cases will generally be more costly as compared to a solution optimal in 95% of cases. Additionally, incorporating stochasticity will likely result in longer runtimes.

References

[Aeroports De Paris, 2023] Aeroports De Paris (2023). FEE SCHEDULE FOR SERVICES RENDERED. https://www.parisaeroport.fr/docs/default-source/professionnel-fichiers/services-aux-compagnies-aeriennes/adp---tarifs-2023-homologues_eng.pdf?sfvrsn=e7f2edfa_0. Accessed: 2023-05-14.

[Airbus, 2022] Airbus (2022). Etihad airways scales up its cargo operations with airbus' new generation a350f freighter. https://www.airbus.com/en/newsroom/press-releases/2022-08-etihad-airways-scales-up-its-cargo-operations-with-airbus-new. Accessed on 2023-05-14.

[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.

[Bushey et al., 2022] Bushey, C., England, A., and Pfeifer, S. (2022). Qatar Airways orders first 777X freighters from Boeing. *Finanial Times*.

[Crabtree et al., 2020] Crabtree, T., Hoang, T., Gildemann, G., and Collingwood, J. (2020). World Air Cargo Forecast 2020–2039. Technical report, The Boeing Company.

[Delgado and Mora, 2021] Delgado, F. and Mora, J. (2021). A matheuristic approach to the air-cargo recovery problem under demand disruption. *Journal of Air Transport Management*, 90(October 2020):101939.

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

[Feng et al., 2015] Feng, B., Li, Y., and Shen, Z. J. M. (2015). Air cargo operations: Literature review and comparison with practices. *Transportation Research Part C: Emerging Technologies*, 56:263–280.

[Gurobi Optimization, LLC, 2023] Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.

[Hong et al., 2023] Hong, S.-J., Kim, W., and Niranjan, S. (2023). Challenges to the air cargo business of combination carriers: Analysis of two major korean airlines. *Journal of Air Transport Management*, 108:102360.

[IATA, 2015] IATA (2015). The value of air cargo air cargo makes it happen. https://www.iata.org/contentassets/4d3961c878894c8a8725278607d8ad52/air-cargo-brochure.pdf. Accessed on 2023-05-14.

[IATA, 2023] IATA (2023). IATA - Unit Load Devices. https://www.iata.org/en/programs/cargo/cargo-operations/unit-load-devices/. Accessed: 2023-05-14.

[Lange, 2019] Lange, A. (2019). Does cargo matter? The impact of air cargo operations on departure on-time performance for combination carriers. *Transportation Research Part A: Policy and Practice*, 119(December 2017):214–223.

[Reis and Silva, 2016] Reis, V. and Silva, J. (2016). Assessing the air cargo business models of combination airlines. *Journal of Air Transport Management*, 57:250–259.

[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.

[Shparberg and Lange, 2022] Shparberg, S. and Lange, B. (2022). Global Market Forecast 2022. Technical report, Airbus.

[Singapore Airlines, 2022] Singapore Airlines (2022). Singapore airlines firms up order for seven airbus a350f freighter aircraft. https://www.singaporeair.com/en_UK/be/media-centre/press-release/article/?q=en_UK/2022/January-March/ne0222-220216. Accessed on 2023-05-14.

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

[United Nations, 2006] United Nations (2006). Standard international trade classification revision 4. ST/ESA/STAT/SER.M/34/Rev.4.

[United Nations, Department Of Economic and Social Affairs, 2023] United Nations, Department Of Economic and Social Affairs (2023). UN Comtrade Database. `https://comtradeplus.un.org/`. Accessed: 2023-04-01.

[United Nations Department of Global Communications, 2023] United Nations Department of Global Communications (2023). EU airlines join forces with UNICEF to transport COVID-19 vaccines around world. `https://www.un.org/en/coronavirus/eu-airlines-join-forces-unicef-transport-covid-19-vaccines-around-world`. Accessed: 2023-05-14.

[Van der Meulen et al., 2020] Van der Meulen, S., Grijspaardt, T., Mars, W., Van der Geest, W., Roest-Crollius, A., and Kiel, J. (2020). Cost figures for freight transport—final report. *Research to Progress*, pages 1–85.

[Xiao et al., 2022] Xiao, F., Guo, S., Huang, L., Huang, L., and Liang, Z. (2022). Integrated aircraft tail assignment and cargo routing problem with through cargo consideration. *Transportation Research Part B: Methodological*, 162:328–351.

[Yan and Chen, 2008] Yan, S. and Chen, C.-H. (2008). Optimal flight scheduling models for cargo airlines under alliances. *Journal of Scheduling*, 11(3):175–186.

[Yan et al., 2006] Yan, S., Chen, S.-C., and Chen, C.-H. (2006). Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E: Logistics and Transportation Review*, 42(5):409–430.

Appendices

A Time Space Network

The basic structure of the time space network is shown in Figure 8. Cargo flight arcs represent the movement of a full-freighter between two different airports. Passenger (referred to as "pax") flight arcs connect two different airports using a passenger flight. Ground arcs connect two adjacent nodes within the same airport. In certain cases, two nodes in the time space network can be connected by both a cargo flight arc and a pax flight arc. Additionally, there can be multiple pax flight arcs connecting two nodes. No service arcs represent the direct movement of a request from its origin node to its destination node, indicating that the request could not be delivered. Therefore, the use of no service arcs has high costs.

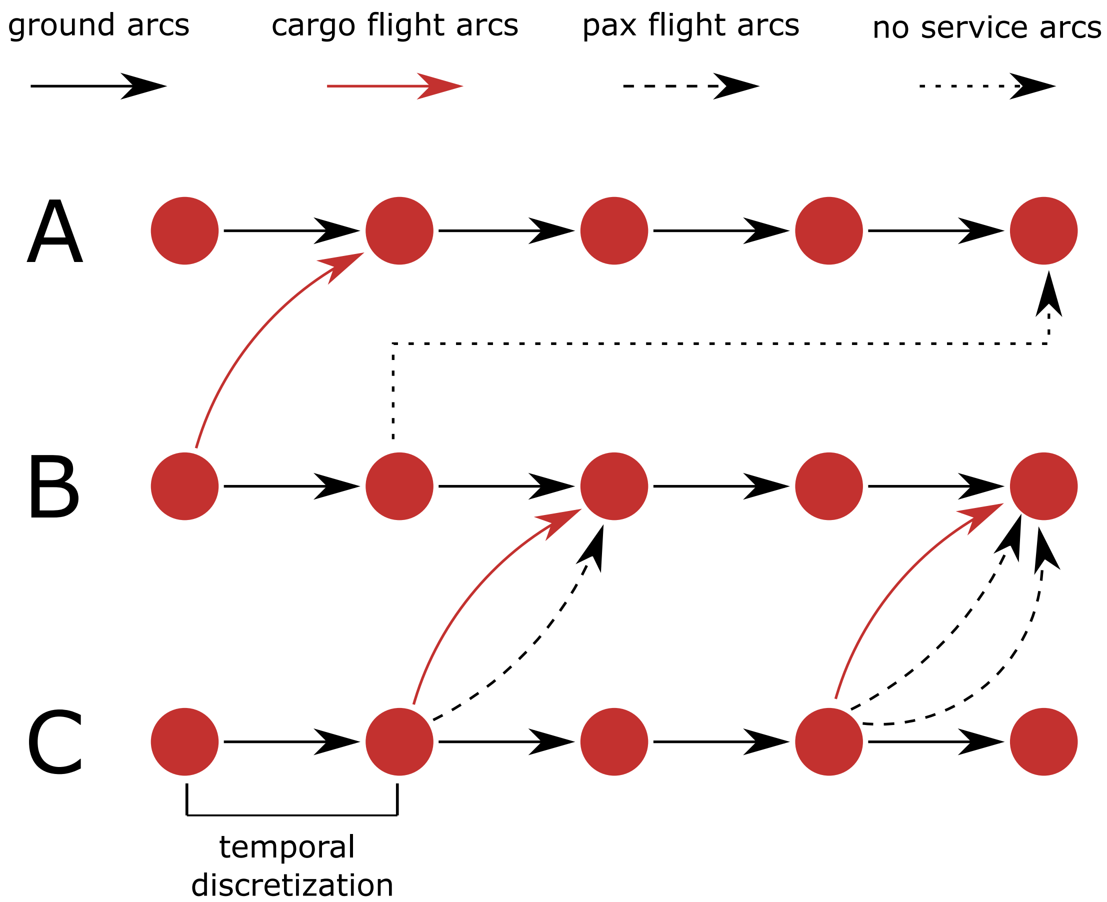


Figure 8: Time Space Network Diagram

B Cargo Capacity During the Day

Figure 9 shows the outgoing hourly cargo capacity of Lufthansa Cargo in Frankfurt. There are peaks during the day at 13:00 and 22:00 hours. This served as an inspiration when generating the release and due times for cargo requests.

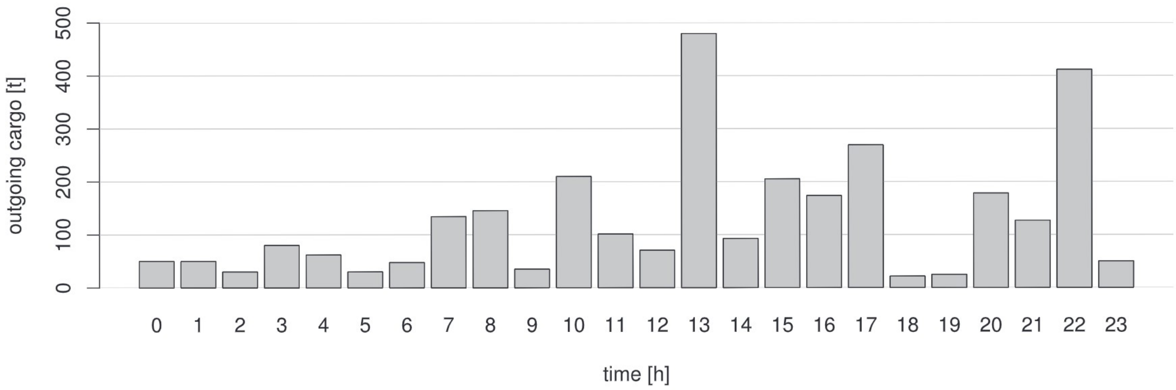


Figure 9: Hourly cargo capacity of Lufthansa Cargo out of Frankfurt on 2016-10-21 (Fig. 5. in [Brandt and Nickel, 2019])

C Cost Figures for Aircraft

Table 14 presents the cost figures for the various aircraft used in the problem instances. Aircraft that have a higher MTOW also have higher landing fees (e.g., B747F vs A321P2F). Furthermore, the variable cost (indicating fuel consumption) is higher for larger aircraft.

Table 14: Cost Figures for Aircraft

Aircraft	**PassengerAircraft**	**WideBody**	**VariableCost**	**FixedCosts**	**CrewCosts**	**LandingFees**
A220	TRUE	FALSE	0.09			
A320	TRUE	FALSE	0.07			
A330	TRUE	TRUE	0.10			
A340	TRUE	TRUE	0.11			
A350	TRUE	TRUE	0.09			
A380	TRUE	TRUE	0.11			
B737	TRUE	FALSE	0.09			
B747	TRUE	TRUE	0.11			
B767	TRUE	TRUE	0.08			
B777	TRUE	TRUE	0.10			
B787	TRUE	TRUE	0.09			
E190	TRUE	FALSE	0.10			
B777F	FALSE	TRUE	0.09	2.11	1.81	1782.94
MD11F	FALSE	TRUE	0.11	1.25	1.81	1520.11
B767F	FALSE	TRUE	0.08	1.32	1.81	1098.89
B747F	FALSE	TRUE	0.11	2.51	1.81	2025.18
A350F	FALSE	TRUE	0.09	2.19	1.81	1660.45
A321P2F	FALSE	FALSE	0.07	0.81	1.81	701.85
A330P2F	FALSE	TRUE	0.10	1.54	1.81	1324.62
A330-200F	FALSE	TRUE	0.10	1.45	1.81	1294.86

D Routing Solution of the CD-ABM Model for Instance 2

Figure 10 shows the routing and scheduling solution for Instance 2 when solved using the CD-ABM model. Unlike the solution of the F-ABM model, here the full-freighter does not exclusively fly between IST and BOM. It is also worthwhile to note that the load factors for the full-freighter flights between IST and LHR are rather low, likely because a lot of the cargo requests are already being transported by the scheduled passenger flights.

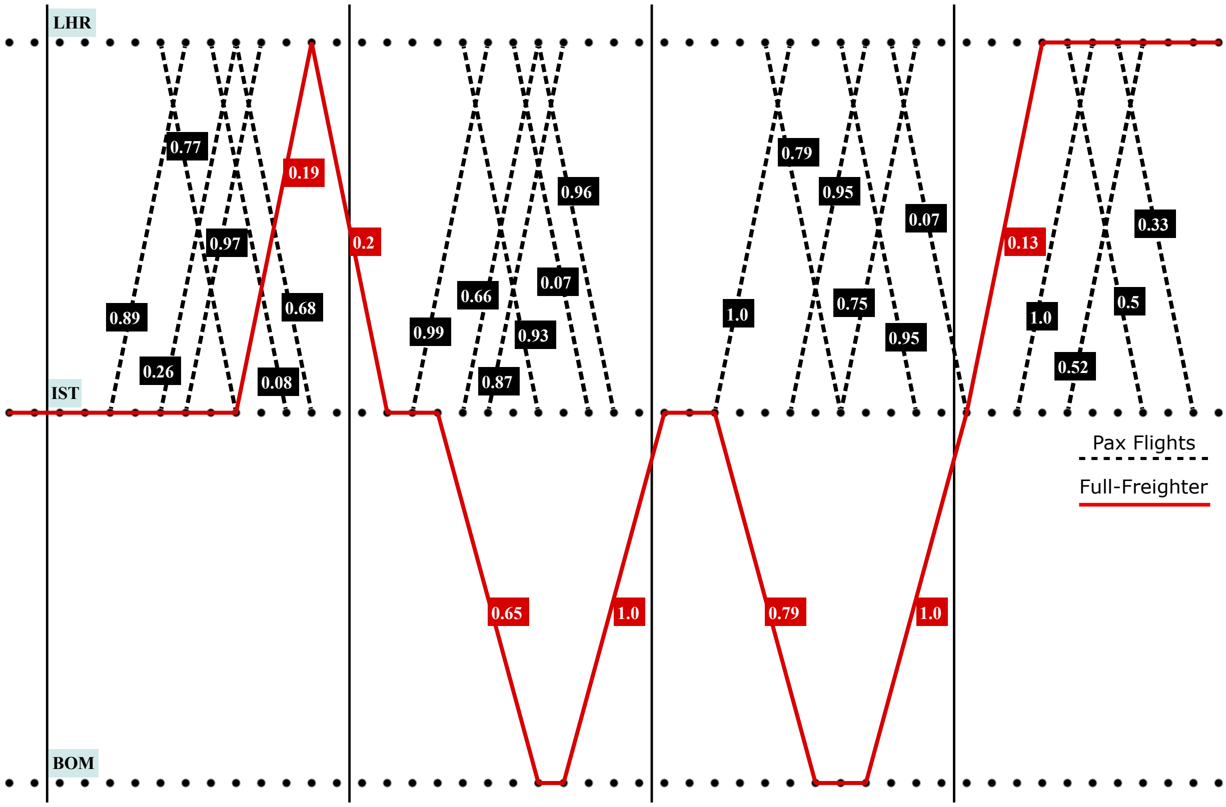


Figure 10: Routing Solution of CD-ABM Model for the Illustrative Example

II

Literature Study
previously graded under AE4020

Optimizing Cargo Operations for Combination Airlines

Literature Study (AE4020)

Sidharatha Thakur (4541340)

Optimizing Cargo Operations for Combination Airlines

Literature Study (AE4020)

by

Sidharatha Thakur (4541340)

Student: Sidharatha Thakur (4541340)
Thesis Supervisor: Dr. A. Bombelli TU Delft, daily supervisor

cover page image taken from Versleijen (2017)

TU Delft

Contents

Acronyms

ACSRP air cargo schedule recovery problem
APK available passenger kilometres
ATK available tonnes kilometres

BSC balanced score card
BSCC Basic Service Combination Carrier

cdf cumulative distribution function
cg centre of gravity

FAM fleet assignment model
FEC Federal Express Company
FSCC Full Service Combination Carriers

KPA Key Performance Areas
KPI Key Performance Indicator

MAC Mean Aerodynamic Chord
MACBETH Measuring Attractiveness by a Categorical Based Evaluation Technique
MCDA Multi-Criteria Decision Analysis
MSIP multistage stochastic integer programming problem

RMP restricted master problem
RPK revenue passenger kilometres
RTK revenue tonne kilometres

SFSCC Separate Profit and Loss Full Service Combination Carrier

ULD Unit load devices

1 Introduction

Air cargo transport is a vital part of the global economy and supply chain. Many different kinds of products have to be transported quickly and air cargo transport helps achieve this. Combination airlines play a key role in transporting cargo. These types of airlines have both a passenger network and a cargo network. Therefore, they can synergize and use both these networks to transport the cargo. However, supply side and demand side uncertainties can cause disruptions to the operations of combination airlines.

Supply side uncertainties refer to the uncertainty in the available belly-space of a passenger aircraft that can be used to transport the cargo. Demand side uncertainties refer to the uncertainty in the cargo booking requests. Most cargo carriers rely on manual planners to deal with the uncertainties on a daily basis (Delgado et al., 2020). These manual planners reroute and reschedule the cargo requests and flights. This process can be time consuming and will likely not yield the optimal solution. Therefore, the objective of this thesis is the following.

Research Objective

The objective of this research is to reroute & reschedule cargo requests and cargo aircraft in a cost-effective manner with a tactical perspective (3 to 7 days). This will be accomplished by designing a computational tool which can proactively and reactively deal with the supply and demand side uncertainties.

This literature review aims to present relevant literature which can help with the research objective of building a computational tool.

This literature review is structured as follows. Chapter 2 gives an overview of air cargo operations. It will provide the necessary background information that will be useful in this research project. Chapter 3 will present the research design. It will elaborate on the research objective, present research questions. It will also formulate sub-questions that will help guide this literature review. Chapter 4 will describe the most relevant optimization techniques that can be used to build the proposed computational tool. Chapter 5 will present the various modelling techniques to model the supply side and demand side uncertainties. Chapter 6 will describe the planning process followed in air transport operations, it will also describe various optimization models which have been used in planning air cargo operations. Chapter 7 will present models that have been used for weight and balance control in aircraft. Adhering to weight and balance requirements is essential in safely operating an aircraft. Based on the findings of the literature review, Chapter 8 will make a preliminary assessment on the best way to create the solution tool proposed in the research objective. Chapter 9 will conclude by providing a succinct answer to each sub-question that was formulated in the research design.

2

Overview of Air Cargo Operations

This chapter gives an overview of various aspects in air cargo operations. Section 2.1 gives an introduction to air cargo operations. Section 2.2 outlines the differences between the passenger business and cargo business. Section 2.3 looks at the types of cargo that is often transported using aircraft. Section 2.4 explains the importance of Unit Load Devices (ULDs). Section 2.5 looks specifically at the operations of a combination airline (a type of air cargo carrier).

2.1. Introduction to Air Cargo Operations

The air cargo industry transports cargo, airmail and freight by using aircraft. There are five main participants in the air cargo shipment process (Feng et al., 2015). These are the shipper, consignee, road transporter, freight forwarder and air cargo carrier. The shipper wants to ship cargo and the consignee is the entity which will receive the cargo. The road transporter is responsible for the ground transportation of the cargo (e.g. between an airport and a destination warehouse). The freight forwarder is an agent which organizes the entire shipment process. They are responsible for arranging all the documentation, preparing the shipment, booking capacity on an aircraft etc. The air cargo carrier flies the cargo from an origin airport to a destination airport. In most cases, the air cargo carrier interacts mainly with the freight forwarder.

Crabtree et al. (2020) divides the air cargo carriers into four groups. These are the belly-only carriers, cargo specialists carriers, combination carriers and integrated express carriers. The belly-only carriers offer air cargo capacity in the lower deck (also called belly) of their passenger aircraft. The main benefit of this type of carrier is that it allows cargo transport between origin-destination pairs which perhaps have large passenger demand but not a large enough cargo demand. The downside is that the typical single-aisle regional jets used in passenger transport often cannot accommodate pallets/containers in their lower holds, they are also not able to carry large/oddly sized cargo. The second type of air cargo carrier are the cargo specialist air carriers. They have a fleet of full freighters, they can use the main deck of the aircraft to carry large amounts of cargo. This allows them to carry large and oddly sized cargo. They can operate on a scheduled or on a chartered basis. There is often not enough demand between an origin-destination pair to fill a large cargo aircraft. Therefore, these freighters add stopovers in their flight. This allows them to load cargo at other airports and thereby increase the capacity utilization of the aircraft. Unlike passengers, multiple stopovers will not be an inconvenience to the cargo. Adding multiple stopovers allows the aircraft to carry less fuel in each flight leg, this increases the cargo carrying capacity in each leg.

The third type of air cargo carriers are the combination carriers. These operate both passenger aircraft and cargo aircraft. These use the belly-space of their passenger aircraft and also dedicated freights to carry the cargo. They can realize synergies in their passenger and cargo networks. For instance, they can feed cargo from their passenger network to their cargo network. The last type of air cargo carriers are the integrated express carriers. They operate a fleet of trucks and aircraft which enables them to provide a door-to-door service. This means that they pick-up cargo from the shipper and deliver it to the consignee. In essence, they combine the roles of the road transporter, freight forwarder and air cargo carrier. Integrated express carriers aim to provide short cargo transport times. To do this, they often impose size and handling requirements on the cargo (Brandt, 2017).

Figure 2.1 provides a revenue breakdown of the various types of air cargo carriers in 2019. One observes that the market is dominated by combination carriers (36% of the revenue) and integrated express carriers (42% of the revenue). Furthermore, almost 90% of the revenues come from air cargo carriers which operate freighter aircraft.

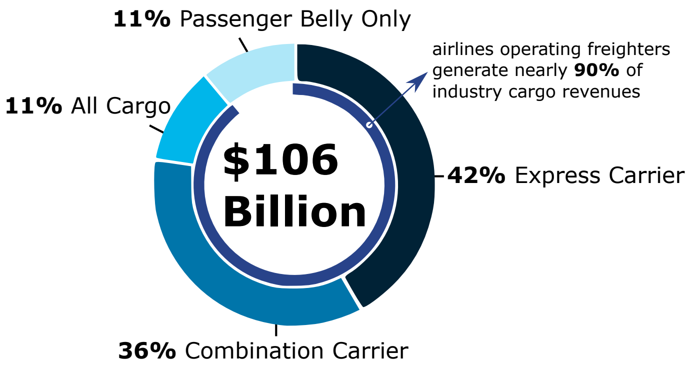


Figure 2.1: Breakdown of 2019 air cargo revenues by carrier type (Crabtree et al., 2020, page 19)

The following sections describe key properties of the air cargo industry. This will give a better picture of the industry and how it differentiates itself from the air passenger business.

2.2. Comparison between passenger and cargo business

There are notable differences in the passenger and air cargo business (Kasilingam, 1997). The standard unit in the passenger business is a seat; passenger airlines sell seats to customers. On the other hand, air cargo carriers work with a multi-dimensional system, they sell capacity on an aircraft. The capacity is determined by the volume, shape and weight of the cargo. This multidimensional approach can cause complications. For instance, some cargo may be light enough to be loaded on an aircraft but not conform to volume requirements and thus cannot be loaded.

The loading process also requires more attention in air cargo transport. Passengers, often, can board and de-board the aircraft on their own. In most cases, they do this in a responsible and orderly manner. On the other hand, cargo has to be loaded and offloaded by ground handlers. Therefore, the optimal loading procedure is relevant for air cargo carriers.

The passenger network is bi-directional. Most of the passengers have a return ticket to their origin airport. Cargo is often unidirectional. There can be a lot of cargo demand from one airport to another and virtually no demand for the return flight. This is especially true when there are producers/factories in one part of the world who supply products via air to customers in another part of the world. For instance, this is often the case for cargo transport between Asia (suppliers) and Europe.

Lastly, there is comparatively more predictable booking behaviour for passengers as compared to cargo. Passengers often book and pay for their flight months in advanced. Only a handful of the passenger re-book or cancel their flight. The booking behaviour is different in the air cargo industry. Freight forwarders can book capacity 6 to 12 months before departure. However, on the day of the flight, they might not show up with cargo or come with less cargo than expected. Furthermore, capacity on the aircraft can also be bought on the short-term spot markets. Thus, new capacity bookings can be made hours before the departure of the aircraft. This makes it difficult for air cargo carriers to gauge the actual capacity bookings for their flight. Furthermore, for combination airlines which also use belly-space to transport the cargo, it is difficult for them to know how much belly-space is available. This is because the check-in cargo of the passenger has priority over the cargo when being transported on the passenger flight. The size of the check-in cargo of the passenger is only known moments before departure. This creates an uncertainty in the belly-space capacity of the passenger aircraft. For instance, if there is limited belly-space capacity, some cargo will not be able to be transported on the passenger aircraft.

2.3. Types of Cargo

The typical price of transporting cargo via a ship is around 0.04 EUR/kg. This is significantly lower than transporting it via air, this costs around 1.84 EUR/kg (Gerber, 2017). One must ask what the main advantages are of air cargo which warrant such a high price. The obvious difference is the shipping time, air cargo enables shippers to transport their cargo very fast over large distances. Peter Gerber, formerly CEO Lufthansa Cargo, outlines four situations in which air cargo is necessary for the shipper (Gerber, 2017):

- **time-to-market**: Some goods require a short time-to-market. For example, the latest technology gadgets or fashion trends must reach the market quickly. These companies typically use air cargo to ship these goods. Transporting goods by ship would simply take too much time.
- **just-in-time logistics**: Complicated machine parts, critical spare parts are key in ensuring the optimal functioning of modern day production chains. These parts are also often shipped by air cargo.
- **security or safety demands**: High value goods like money, jewellery, gold etc require secure and quick transport. Dangerous cargo like radioactive materials also must be shipped securely. These are also typically transported by air cargo.
- **perishables**: Some cargo can degrade quickly and must be shipped in a short time frame using air cargo. Fruits, vegetables, flowers, vaccines are typical examples of perishables.

2.4. Unit Load Devices

Unit load devices (ULD) are devices on which the cargo is placed. These devices are then loaded onto the aircraft and secured with latches. Using ULDs streamlines the aircraft cargo loading and offloading process. There are typically two kinds of ULDs; containers and pallets. Containers have a fixed enclosure in which cargo is put. Since the aircraft fuselage is round, the contours of the containers are designed such that the volume inside the container is maximized. The contours of the container determine the aircraft on which it can be loaded. Furthermore, the contour also determines the locations where it can be loaded on the aircraft. For instance, an upper deck container will not fit on the lower deck. Using a container can also allow for temperature control when transporting frozen goods, medicine etc.

The other type of ULD is a pallet. This consists of a base on which cargo is manually packed. Since there is no limiting enclosure, using a pallet allows for greater freedom in the packing process. It is easier to pack larger shipments on a pallet. Once loaded, the pallet is secured by a net and straps. When loading containers or pallets, it is desirable to have the centre of gravity of the loaded ULDs close to the centre of base of the ULD.

Figure 2.2 shows examples of containers and pallets. Figure 2.2a shows an example of a container. The container has an angled side, this suggests that it will be loaded in the lower deck of the aircraft. The shapes and dimensions of the containers are standardized. Each container type has an IATA code, the container in Figure 2.2a has the IATA code “AKE”, it is also known as the “LD-3” container (VRR; WestJet). Figure 2.2b shows an example of a pallet. The pallet is built by placing cargo on a base and then securing it with an air pallet net.

There are two main processes associated with ULDs (Brandt, 2017). First, the ULDs have to be packed with cargo. This is known as the build-up process. One must ensure that the total weight of the packed ULD does not exceed the maximum permissible weight. Once the ULDs are loaded, they have to be put onto the aircraft. This is the aircraft loading process. Here, one must take into consideration the weight and balance requirements for the aircraft. The loaded aircraft cannot be too heavy and it must have a centre of gravity which is within safety limits.

(a) Containers (b) Pallets

Figure 2.2: Examples of Containers and Pallets (WestJet)

2.5. Combination Airlines

The focus of this literature review is the operations of combination airlines. Their operations are comparatively more complicated than the other types of air cargo carriers as they have to maintain both a passenger network, a cargo network and also optimize the synergies between these networks.

Reis and Silva (2016) provide a nice overview of the business models of combination airlines. The paper argues that there are three types of combination airlines. The Basic Service Combination Carrier (BSCC) have the lowest commitment towards carrying cargo. They are analogous to the belly-only airlines discussed in Crabtree et al. (2020). These combination carriers carry cargo in the available belly-space of their passenger airlines. They do not operate additional freighters. They have a small sales teams which handles the cargo division. Full Service Combination Carriers (FSCC) have larger cargo departments than their BSCC counterparts. They extensively use their passenger aircraft and network to carry cargo, however they may also operate ad-hoc freighters to fulfil some cargo demand. Separate Profit and Loss Full Service Combination Carrier (SFSCC) have the largest cargo operations across the different classifications. They operate a fleet of passenger aircraft and full-freighters to carry the cargo. Table 2.1 summarizes the three types of combination carriers and their properties.

Table 2.1: Types of Combination Carriers (Reis and Silva, 2016, adapted from Table 1)

	BSCC	**FSCC**	**SFSCC**
Product differentiation	Basic product	Medium range	Broad range
Capacity management	Hardly present	Basic	Complex
Competitive behviour	Market breaker	Follower	Market maker
Route network	Follows PAX network	Mainly follows PAX Network	Very large
Hub	PAX hub	Mainly PAX hub	Mainly PAX hub
Cargo Fleet	Only belly-space	Belly-space and ad-hoc freighters	Belly-space and freighters

The authors in Reis and Silva (2016) interviewed key staff members at major airlines to learn about their cargo operations. Based on these interviews, the authors were able to classify the airlines in one of the three aforementioned categories. This is shown in Table 2.2. This table gives a good overview of the different kinds of combination carriers that operate in the real market. Airlines which fall under the SFSCC category are the main focus of this paper as they have extensive passenger and cargo operations.

Table 2.2: Classification of some combination carriers (Reis and Silva, 2016, Table 5)

Type	**Airlines**
BSCC	TAP Cargo, Brussels Airlines Cargo, SATA Cargo
FSCC	Turkish Cargo, SWISS WorldCargo, Finnair Cargo, IAG Cargo
SFSCC	AF-KLM Cargo, Emirates SkyCargo, Lufthansa Cargo

The increased complexity of the operations performed by combination airlines also has an impact on the performance of the airline. Lange (2019) uses a sample of American airlines and explores the relationship between a combination airline and its departure on-time performance. Departure on-time performance is used as opposed to arrival on-time performance as there are comparatively more factors which affect arrival delay than departure delay. For example, taxiing time, airtime and landing affect arrival delay and not departure delay. Therefore, departure delay is a better variable for measuring the effect of increased complexity, in passenger and cargo loading, on the performance of the airline. The paper had 3 main findings. First, combination airlines have more departure delays than their pure passenger counterparts. Second, widebody aircraft have increased departure delays as compared to narrow body aircraft. Lastly, the belly hold capacity has no effect on the departure delays of passenger airlines, whereas it does affect the departure delays of combination airlines. Based on these results, the paper concludes that combination airlines, which load both cargo and passengers, have increased departure delays. One the one hand, combination airlines can generate more revenue by carrying cargo, however this also negatively affects their performance. This suggests that it is essential to further optimize the operations of a combination airline in order to improve its performance. One such improvement could be to better deal with the uncertainties prevalent in the operations of a combination airline. Specifically, dealing with the uncertainty in the available belly-space of passenger aircraft and the uncertainty in the cargo booking can help improve the performance of combination airlines.

2.6. Concluding Remarks

This chapter has provided an overview of various aspects in air cargo operations. It elaborated on the various entities in the air cargo shipment process. Furthermore, it also looked specifically at combination airlines and their operations. The uncertainty in the available belly-space on passenger aircraft combined with the uncertainty in the cargo demand (i.e. booking requests) complicates the operations for a combination airline. The next chapter elaborates on these complications and establishes the research objectives.

3

Research Design

This chapter describes the proposed research design. Section 3.1 states the research problem and objective. Section 3.2 describes the research question and sub-questions which will help guide the research. Section 3.3 gives the problem definition and operationalisation, this talks more in detail on the problem setting and assumptions. Finally, Section 3.4 describes the key indicators which can be used to judge the solution to the problem.

3.1. Research Problem and Objective

Chapter 2 highlighted two sources of uncertainty faced by combination airlines. First, the uncertainty in the belly-space of passenger aircraft. This is considered as a supply side uncertainty. Second, the cargo booking demand uncertainty. This is considered as a demand side uncertainty. These two uncertainties can adversely affect the operations of the combination airlines. If the demand is high and capacity is low, the aircraft will not be able to transport some cargo requests. This can result in it paying a penalty. On the other hand, if the demand is low and the capacity is high, the aircraft is flying with a low load factor. Both scenarios can lead to the airline making a loss.

Most cargo carriers redesign their schedule on a daily bases to deal with the uncertainties. This process is done manually by experienced planners. They aim to reroute cargo, reroute aircraft, add new flights etc. (Delgado et al., 2020). This manual approach can help in resolving the problem, however the solution will likely be sub-optimal. Furthermore, it may take a relatively long time to manually perform the rerouting and scheduling. This can be a problem since most modifications have to be done in a short time frame. Given these problems, the research objective can be defined as follows:

Research Objective

The objective of this research is to reroute & reschedule cargo requests and cargo aircraft in a cost-effective manner with a tactical perspective (3 to 7 days). This will be accomplished by designing a computational tool which can proactively and reactively deal with the supply and demand side uncertainties.

3.2. Research Questions

The aforementioned research objective leads to a number of to a number of research questions and sub-questions which will help guide the thesis. The main research questions are formulated as follows:

Question 1

How can a combination airline incorporate current and potential disruptions, caused by uncertainties, in a computational tool in order to cost-effectively reroute and reschedule its cargo requests and cargo flights?

Question 2

What is the improvement in performance for a combination airline by introducing a computational tool to reroute and reschedule its cargo requests and cargo flights given current and potential disruptions caused by uncertainties?

Both questions look somewhat similar but there are subtle differences between them. Question 1 focuses more on the design of the computational tool. Question 2 focuses more on the performance of the tool. It looks at how the tool can help bring improvements (financial gains, reduced costs etc) to a combination airline. This implies that the performance of the combination airline when using the tool should be compared to when it's performance when it does not use the tool.

3.2.1. Sub-Questions

There are a number of sub-questions that arise from the research questions. These sub-questions will help guide this thesis. This literature review aims to select literature which answer these sub-questions. Answering these sub-questions will help in the construction of the computational tool. The sub-questions are:

Sub-Question 1: **What are the key performance indicators which can help evaluate the performance of the computational tool and the performance of the cargo carrier?**

One needs to construct a set of indicators which will help compare the performance of the combination airline when using the tool to when it does not use the tool. Furthermore, one also needs to evaluate the performance of the tool itself and see if incorporating this tool to the daily operations of the combination airline is feasible. For instance, a computational tool may give very optimal solutions but if it takes 3 days to generate this solution, then using the tool is not realistic.

Sub-Question 2: **What are the most relevant optimization techniques which can be used in constructing the computational tool?**

There has been a lot of progress in the field of operations research. There are numerous techniques which can be used as the backbone of the computational tool. This sub-question aims to find the relevant techniques that can be used in the computational tool.

Sub-Question 3: **How can the supply side and demand side uncertainties be modelled?**

Being able to accurately model the uncertainties is necessary in order to design the computational tool. The uncertainties are divided into supply side (e.g. belly-space capacity) and demand side (i.e. cargo booking requests). This sub-question will look at the techniques that can be used to model these uncertainties.

Sub-Question 4: **What are the current, if any, deterministic or stochastic models which deal with planning operations of a cargo carrier?**

This sub-question aims to find papers which present models to deal which help plan the operations of a cargo carrier. It will be helpful if these papers also considered the uncertainties in their models. This sub-question is different than Sub-Question 2. There, one aims at finding optimization techniques, whereas in this sub-question, one aims at looking at how these techniques are used in models pertaining to air cargo operations.

Sub-Question 5: **What are the various ways the weight and balance of the aircraft can be ensured when carrying the cargo?**

The computational tool must ensure that the rerouting of cargo does not result in a situation where the aircraft cannot take off because its too heavy or not properly loaded. This sub-question will look at the current techniques for weight and balance control in aircraft.

3.3. Problem Definition and Operationalisation

The proposed problem definition and operationalisation is now discussed. The computational tool help improve the operations of a combination carrier. The carrier will have scheduled passenger flights and also operates cargo flights. The schedule of the passenger flights is assumed to be given. Furthermore, the tail assignment for the passenger flights is also assumed to already have been done. Tail assignment refers to assigning individual aircraft in the fleet to flights that are to be performed. The airline will have uncertain demand for its cargo services, this demand will only reveal itself very shortly prior to its dispatch date. The cargo will have a source airport, destination airport, staring time, destination time and request characteristics (e.g. weight and volume). The starting time represents the time from when it can be picked up from its source airport. The destination time represents the time by which it must reach its destination airport.

The airline can carry the cargo in the available belly-space of its passenger aircraft or in dedicated freighter aircraft. In a passenger aircraft, the check-in luggage of a passenger has priority over the cargo. The cargo can have multiple stopovers and aircraft changes, as long as it arrives at its destination airport on time. The aim is to schedule and route the cargo aircraft such that the cargo reaches its destination on time. One key advantage of combination airlines is that they can use both their passenger and cargo networks to carry the cargo. Therefore, the routing of the cargo aircraft should not be independent of the already established passenger aircraft routing and scheduling. The freighter aircraft should be routed in such a way that the combined cargo carrying capacity of the cargo and passenger aircraft is maximized.

As mentioned, there are two sources of uncertainties in this problem. The first is the available belly-space in the passenger aircraft that can be used to carry the cargo. The weight of passenger check-in luggage is not known before hand and therefore it could be that some cargo cannot be loaded on a passenger aircraft because there is no space for it. The second source of uncertainty is the cargo booking itself. The exact amount of booked cargo is often only known very shortly before a flight. Therefore, having a fixed routing schedule for cargo aircraft is not advisable as demand changes very frequently. It is imperative that these sources of uncertainties are modelled appropriately, this will ensure the validity of the model.

There are some assumptions to the model. Note that during the course of this thesis project, these assumptions may change in order to simplify or increase the complexity of the model. These assumptions are listed as follows:

1. Given the short term nature of the uncertainties, the time horizon is between 3 days and a week.
2. The fleet of the airline is assumed to be known. This will help determine the cargo carrying capacity of the individual aircraft.
3. The set of airports, both cargo and passenger, from which the airline operates is also assumed to be known in advanced. This will help determine the distances and flying times.
4. It is assumed that there will always be available crew to man the additional cargo flights.
5. The beginning and ending locations of the cargo aircraft are fixed. Although, this is based at the fleet level, thus at the end of the time horizon, a particular aircraft type should be available at the desired airport.
6. The belly-space and cargo booking uncertainties are very short term in nature, whereas the passenger aircraft routing is much more long term. Hence, it is decided that the routing and tail assignment of passenger aircraft has already been done at this stage. This cannot be changed.
7. In the first iteration of the model, it is assumed that the airports do not have take-off and landing slot restrictions. This will help simplify the scheduling of the cargo flights. Although, if time permits, the author would like to relax these slot constraints in the model.

8. Finally, given the uncertainties, it is possible that some of the cargo will not be delivered. Therefore, model should strive to ensure that least a certain percentage of the cargo demand, for example 95%, is fulfilled. The 95% parameter can be tweaked and a sensitivity analysis on this can be performed.

3.4. Performance Indicators

One must construct criteria which will help evaluate the solution given by the model. In air transport, these are often done using Key Performance Areas (KPA) and Key Performance Indicators (KPI). KPAs refer to broad areas which a company must focus on. For instance, financial performance, sustainability are examples of KPAs. KPAs are composed of KPIs, these are numerical indicators which evaluate the performance of a company for that KPA. For instance, the KPA financial performance can be measured using a KPIs such as operational profit, operational revenue etc (Cachola, 2017). In evaluating the model, The selection of KPAs and KPIs must be specific to air transport. There are numerous sources of literature which discuss the relevant KPIs in the air cargo industry. These are described next.

Cachola (2017) investigates how to evaluate the performance and efficiency of air cargo carriers. The paper proposes several KPIs that are relevant for air cargo carriers. It uses Multi-Criteria Decision Analysis (MCDA) to assign relative weights to the KPAs and KPIs and thereafter rank them. The MCDA tool used was “Measuring Attractiveness by a Categorical Based Evaluation Technique” (MACBETH). Thereafter, the paper used the performance evaluation technique developed to evaluate the performance and efficiency of the operations of two large cargo airlines, namely Cargolux and Lufthansa Cargo.

Chen et al. (2008) develops KPIs for air cargo transport using the balanced score card (BSC) method developed by Robert Kaplan and David Norton (Kaplan and Norton, 2005). The BSC method asks managers to construct KPIs for a business from four different perspectives. These perspectives are financial, customer, internal processes and organizational capacity.

Wouter (2017) gives a valuable presentation on “The Strategy of Air Cargo airlines”. It discusses the various business models these cargo airlines employ. The presentation also gives an extensive list of 45 KPIs that can be used to evaluate air cargo carriers. The author uses these KPI to evaluate the strategy of 47 air cargo carriers.

The air cargo KPIs mentioned in the aforementioned papers are analysed, the most relevant ones are presented below under their respective KPA headings. As mentioned, these KPIs will help gauge the solution returned by the model. The KPIs of the airline when using the proposed computational tool can be compared to when it does not use the computational tool. The hope is that using the computational tool leads to better KPI results.

3.4.1. Operational

The most simple operational KPI is the total cargo carried. From this, one can construct the revenue tonne kilometres (RTK), this is equal to the total freight carried (in tonnes) multiplied by the distance flown (in kilometres). The available tonnes kilometres (ATK) is the overall capacity of cargo that can be carried (in tonnes) multiplied by the distance flown (in kilometres). The RTK and ATK are analogous to the passenger transport KPIs of revenue passenger kilometres (RPK) and available passenger kilometres (APK) respectively. The RTK effectively measures that total cargo that is carried. The ATK measures the total cargo carrying capacity of the airline. The load factor can be expressed as RTK divided by ATK, it can be thought of as the ratio between the actual cargo carried and the cargo carrying capacity. The cargo airline should aim for a higher load factor.

The average aircraft utilization is a fleet level operational KPI. It measures the average amount of time an aircraft is used for transporting the cargo. Higher utilization results, quite clearly, to a better usage of the aircraft, the fixed costs required to acquire the aircraft are spread over more trips (Mirza, 2008).

One of the model aims is to fulfil the cargo demand. Therefore, there should also be some KPIs which measure the effectiveness of performing this task. The average transport time will indicate how long it took for the cargo to arrive at its destination airport, measured from the time it was ready for delivery at the source airport. The percentage of cargo demand not fulfilled will measure how much of the cargo demand was not fulfilled. The average airport dwell time will measure the average amount of time the cargo packet was sitting idle at an airport. The average commute time will refer to the average amount of time the cargo was being flown in an aircraft. The transport time is composed of the dwell time and commute time, it could be that a package has a long transport time simply because it had to wait a long time at an airport (high dwell time) or because it had many stopovers (high commute time).

3.4.2. Financial

The operating costs, revenue and profit are the key financial KPIs for the solution. The airline must aim to reduce costs, increase revenue and profit. These financial measures are often expressed as a ratio of either the RTK or ATK. This will help compare these measures among different cargo carriers. For instance, the operating costs are often expressed as a percentage of the ATK, this essentially measures the cost incurred by the airline to produce on unit of cargo carrying capacity. Similarly, the revenue is often expressed as a ratio of RTK.

3.4.3. Environmental

Given the advent of global warming and climate change, it is imperative to minimize the adverse effects of transportation on the climate. Environmental KPIs aim to measure the impact of the model on the climate. These include the total fuel cost and total carbon dioxide emissions per transported tonne. The model should aim to generate solutions which minimize these KPIs. Practically, this can be measured by looking at the total fuel cost of the solution returned by the model.

3.4.4. Computational Tool Performance Indicators

The performance of the computational tool itself must also be evaluated. For instance, if a tool takes a very long time to return an optimal solution, then this is not a desirable tool to use in real life operations. Therefore, the main performance metric for the tool is the runtime; lower runtimes are desirable. Furthermore, the optimality gap is also a valuable indicator. The optimality gap measures how far the returned solution is from the optimal solution. A high optimality gap is not desirable. Lastly, the computational tool will be run on various different uncertainty instances. Therefore, one must be able to measure the average effectiveness of the tool. This can be measured by a success ratio which is the ratio between the total cargo requests fulfilled with the total cargo requests that had to be fulfilled.

3.5. Concluding Remarks

This chapter described the research design. It defined the research objectives and formulated various research questions and sub-questions to help guide this thesis project. It also defined the problem and its assumptions that will be solved by the proposed computational tool. Lastly, it also answered the first sub-question by defining various performance indicators which can be used to judge the computations tool and its potential benefits for the combination carrier. The next chapter aims at answering the second sub-question, it looks at the various optimization techniques that can be used to design the computational tool.

4

Relevant Optimization Techniques

Significant progress has been made in the field of operations research. Mathematicians and industry alike have developed new techniques to solve the large variety of optimizations problems that exist in the real world. This chapter discusses some the techniques used to solve optimization problems that exist in transportation and specifically airline scheduling planning. It is assumed that the reader is familiar with the Simplex method and duality, both of which form the basics of optimization. Some of the techniques described in this section build upon the ideas behind the Simplex method and duality. Many of these techniques described in this chapter are versatile and can be fine-tuned with variations in their execution.

Section 4.1 gives an overview of branch and bound, the technique used to solve mixed integer problems. Section 4.2 explains the procedure of column generation. Section 4.3 describes decomposition techniques, furthermore it also explains the Dantzig-Wolfe and Benders decomposition. Section 4.4 discusses stochastic programming techniques, these can be used to solve problems which have uncertainty in some of their parameters.

4.1. Branch and Bound

Branch and bound is used to solve mixed-integer problems. It was originally developed by Ailsa Land and Alison Harcourt in 1960 (Jünger et al., 2009). It starts by relaxing the integer constraints and solves the problem. Thereafter, it selects a decision variable which should be an integer but is currently not an integer. It proceeds by branching on this variable and solving two new sub-problems, these are similar to the original problem but have additional constraints. As an example assume that x_1 should be an integer but it currently has a value of 15.4. If the algorithm decides to branch on x_1, it will solve two new sub-problems which are similar to the original problem. One of the sub-problems will have the additional constraint that $x_1 \geq 15$ and the other sub-problem will have the constraint that $x_1 \leq 16$. These new sub-problems will be additional nodes in the branch and bound tree. This process of expanding on variables is repeated for the new nodes in the tree. If one of the nodes has all the required integer variables as integers, then one obtains a solution to the problem. If this solution is better than the current best solution (also called incumbent), then the incumbent is updated. If the solution is worse than the incumbent, one stops branching from that node; the node is fathomed. Also, if a node gives an infeasible solution, then one stops branching on that node and the node is fathomed.

There can be variations in the branch and bound algorithm. For instance, the node selection process can be changed. In most cases, depth-first search is used, this means that the algorithm will continue exploring a branch until it has to backtrack and explore another branch. However, one can also use breadth-first search, here all the nodes at a particular level are analysed before going to the next level. Both depth-first search and breadth-first search are apriori rules, the node selection order is decided beforehand. One can also use adaptive rules, here the node selection process is not established beforehand, rather a node is selected based on current node parameters such as bounds (Wolsey and Nemhauser, 1988). Another variation can be the choice of the branching variable. For instance, one can decide to branch on the variable which is the furthest away from an integer (i.e. maximum integer infeasibility). Alternatively, one can also decide to use branch on the variable which is the closest to an integer (i.e. minimum integer infeasibility). There can be other, more complex, ways to select the branching variable. Choosing the right branch and bound algorithm design can significantly reduce the runtime of the algorithm. Morrison et al. (2016) provides an extensive overview of the various advances and variations in the branch and bound algorithm design.

4.2. Column Generation

Column generation can be a useful algorithm to solve large scale linear programming problems. It was first formulated by Gilmore and Gomory (1961) to solve the cutting stock problem. The basic idea behind column generation is to start off with a linear programming model which only has a small subset of all possible decision variables (columns). This is called the restricted master problem (RMP). Thereafter, decision variables which can help improve the objective function are added to the model. This idea is analogous to how the simplex method works. In each iteration of the simplex method, it calculates the reduced cost ($c_j - z_j$) of all non-basic variables. Based on a desirable reduced cost (e.g. negative reduced cost for minimization problems), a decision variable enters the basis. Column generation works in the same way, however instead of calculating the reduced cost for all non-basic variables, the algorithm simply generates a decision variable which will have a desirable reduced cost. This desirable decision variable is added to the RMP. This iterative process is repeated until the algorithm cannot find any more desirable decision variables to add to the RMP. The desirable decision variables are generated by solving the pricing problem. The main benefit of column generation is that the model is solved only with a handful of decision variables, this reduces the computational time and effort required to solve the model.

The reduced cost for variable j is calculated as $c_j - \boldsymbol{\pi}^T \boldsymbol{a_j}$ where c_j is the coefficient of the decision variable in the objective function, $\boldsymbol{\pi}$ is the vector of dual variables and $\boldsymbol{a_j}$ is the vector of coefficients that j has in the constraints (Irnich and Desaulniers, 2005). A sub-problem (also called pricing problem) aims to generate a column such that its reduced cost is negative (if dealing with a minimization problem). If it cannot do so, then the current optimal solution from the RMP is the optimal solution for the entire problem. There are numerous ways to model the pricing problem, this depends largely on the underlying problem and application. The pricing problem can be modelled as a Knapsack problem, shortest path problem etc. One way to accelerate the column generation procedure is to add multiple decision variables in one iteration to the RMP. For instance, it could be that there are multiple decision variable which have a negative reduced cost, all of them can be added to the RMP. Doing this leads to a lower number of iterations. The interested reader will find an elaborate explanation of column generation, its applications and variations in the book Irnich and Desaulniers (2005).

4.3. Decomposition Techniques

Solving a series of smaller problems can be more efficient than solving one large problem, that is in essence the idea of decomposition. Decomposition refers to solution techniques which aim to breakdown, or decompose, a large (non)-linear model into smaller sub-models which contain only some of the decision variables and constraints. These smaller problems are easier to solve than the large problem. The most convenient problem occurrence is when the linear model can entirely be decomposed into smaller problems. Consider the abstract linear model shown in Equation (4.1).

$$
\begin{aligned}
\max \quad & a_1x_1 + a_2x_2 + b_1y_1 + b_2y_2 \\
& c_1x_1 + c_2x_2 \leq h_1 \\
& c_3x_1 + c_4x_2 \leq h_2 \\
& d_1y_1 + d_2y_2 \leq h_3 \\
& d_3y_1 + d_4y_2 \leq h_4 \\
& x_1, x_2, y_1, y_2 \geq 0
\end{aligned}
\tag{4.1}
$$

There are four decision variables, namely x_1, x_2, y_1, y_2. The model has an objective and some constraints. Notice that the problem can be decomposed into two problems, one which has only the

x decision variables and another which has only the y decision variable. This is shown in Equation (4.2).

$$\begin{aligned}\max \quad & a_1x_1 + a_2x_2 \\ & c_1x_1 + c_2x_2 \le h_1 \\ & c_3x_1 + c_4x_2 \le h_2 \\ & x_1, x_2 \ge 0\end{aligned} \qquad \begin{aligned}\max \quad & b_1y_1 + b_2y_2 \\ & d_1y_1 + d_2y_2 \le h_3 \\ & d_3y_1 + d_4y_2 \le h_4 \\ & y_1, y_2 \ge 0\end{aligned} \tag{4.2}$$

Note that both the objective and the constraints are decomposed. Solving these individual decomposed sub-problems will give the optimal values for the decision variables. One quick way to gauge the decomposability of a problem is to examine the coefficient matrix of the constraints. The coefficient matrix of the the constraints of Equation (4.1) is shown in Equation (4.3). One clearly notices the block structure of the matrix. This suggests that the linear problem can be decomposed.

$$\underbrace{\begin{bmatrix} c_1 & c_2 & & \\ c_3 & c_4 & & \\ & & d_1 & d_2 \\ & & d_3 & d_4 \end{bmatrix}}_{\text{block structure}} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \le \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} \tag{4.3}$$

It is seldom the case that a linear problem can be perfectly decomposed into smaller sub-problems. There are always complicating factors that make doing this difficult. However, in some cases, these complications can be managed. “Complicating constraints” and “complicating variables” are types of complications which are often found in linear problems and prevent perfect decomposition. Complicating constraints are constraints which prevent the problem from being decomposed. Suppose a constraint $c_5x_1 + d_5y_1 + d_6y_2 \le h_5$ is added to Equation (4.1). This constraint has variables of type x and y. This prevents the decomposition of the problem. The general structure of the constraint matrix for a problem with complicating constraints is shown in Figure 4.1. It has a block like structure which indicate that some constraints are only relevant for some decision variables. However, it also has some complicating constraints, these incorporate decision variables from different blocks. Relaxing these complicating constraints would result in a decomposable model.

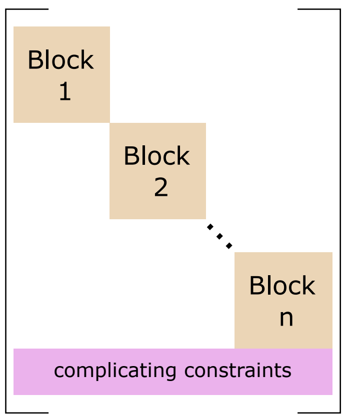


Figure 4.1: Constraint matrix for a problem with complicating constraints

Complicating variables are decision variables which appear in multiple blocks. Consider the problem shown in Equation (4.4).

$$\begin{aligned} \max \quad & a_1x_1 + a_2x_2 + b_1y_1 + b_2y_2 + k_1z_1 \\ & c_1x_1 + c_2x_2 + e_1z_1 \leq h_1 \\ & c_3x_1 + c_4x_2 + e_2z_1 \leq h_2 \\ & d_1y_1 + d_2y_2 + e_3z_1 \leq h_3 \\ & d_3y_1 + d_4y_2 + e_4z_1 \leq h_4 \\ & x_1, x_2, y_1, y_2, z_1 \geq 0 \end{aligned} \tag{4.4}$$

The variable z_1 is a decision variable, notice how it appears in every constraint. This makes it difficult to decompose the model. The general structure of the constraint matrix for a problem with complicating variables is shown in Figure 4.2. A block like structure is noticeable, however there are some decision variables which are in multiple constraints across the blocks.

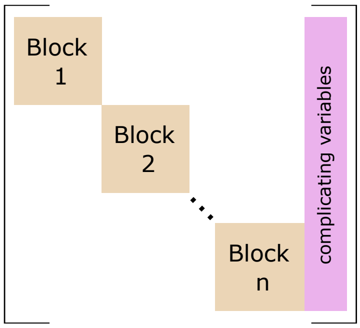


Figure 4.2: Constraint matrix for a problem with complicating variables

A lot of research has been done in developing decomposition techniques to solve models with the aforementioned complications. Table 4.1 and Table 4.2 give an overview of the various techniques that can be used to solve decomposition problems. Table 4.1 assumes that all decision variables are continuous, whereas Table 4.2 assumes that there are some non-continuous decision variables. An asterisk ("*") next to the name indicates that the technique may fail to give a solution.

Table 4.1: Solution techniques for continuous models (Conejo et al., 2006, adapted from table 1.29)

	Linear	**Non-Linear**
Complicating Constraints	Dantzig-Wolfe Augmented Lagrangian	Dantzig-Wolfe Lagrangian Relaxation Augmented Lagrangian Optimality Conditions
Complicating Variables	Benders' Decomposition	Benders' Decomposition

Table 4.2: Solution techniques for non-continuous models (Conejo et al., 2006, adapted from table 1.29)

	Linear	**Non-Linear**
Complicating Constraints	Branch and Price Lagrangian relaxation* Augmented Lagrangian*	Dantzig-Wolfe Lagrangian relaxation Augmented Lagrangian Optimality conditions
Complicating Variables	Lagrangian relaxation Augmented Lagrangian Benders' Decomposition	Benders' Decomposition

The following subsections describe the general working principles of Dantzig-Wolfe decomposition and Benders' decomposition. The former is applicable to models with complicating constraints and the latter is applicable to models with complicating variables. These two solution methods form the basics of solution techniques often encountered in practice and literature.

4.3.1. Dantzig-Wolfe Decomposition

The Dantzig-Wolfe decomposition was developed by George Dantzig and Philip Wolfe (Dantzig and Wolfe, 1960). The algorithm relies on the theorem that any point in a bounded convex region can be expressed as a convex combination of its extreme points. For instance, suppose $p1$, $p2$ and $p3$ are the extreme points of a bounded convex region, then any point in the region can be expressed as $w1p1 + w2p2 + w3p3$, where $w1 + w2 + w3$ add up to 1 (i.e. a convex combination). The general idea behind Dantzig-Wolfe decomposition is to first relax the complicating constraints. This will result in a relaxed feasible region for the problem. Thereafter, find an optimal convex combination of extreme points of the relaxed feasible region such that the resulting convex combination optimizes the objective of the original problem and also satisfies the complicating constraints. The Dantzig Wolfe decomposition very cleverly uses column generation to determine which corner points (from the relaxed feasible region) to add, such that the resulting convex combination maximizes the objective and adheres to the complicating constraints.

This algorithm starts with relaxing the complicating constraints. This allows for decomposing the problem into sub-problems. These sub-problems are given an arbitrary objective and optimized. This can be repeated with different sub-problem objectives to obtain an initial set of solutions which satisfy the relaxed optimization problem. Suppose, for example, that x_1 and x_2 are two solutions to the relaxed optimization problem. Note that x_1 and x_2 essentially represent corner points in the relaxed feasible region. Also, assume that there are n complicating constraints $C_1(x)$ to $C_n(x)$ with right-hand sides b_1 to b_n. The term $C_1(x_1)$ represents the value of complicating constraint 1 for solution x_1.

Thereafter, a master problem is solved. Here, the aim is to determine weights u_1 and u_2, such that the convex combination u_1x_1 + u_2x_1 is maximized. The constraints in the master problem ensure that the resulting convex optimization point satisfies that complicating constraints. For instance, $u_1C_1(x_1) + u_2C_1(x_2) \leq b_1$ ensures that the resulting convex combination satisfies the first complicating constraint. Such a constraint is introduced for all the complicating constraints. A final constraint ensures that the weights u_i add up to 1, $u_1 + u_2 = 1$. This master problem is solved and the dual variables are stored.

Let the dual variable for the i^{th} complicating constraint in the master problem be denoted as λ_i and the dual variable for the final weighing constraint (i.e. $u_1 + u_2 = 1$) be denoted as σ. Suppose there is another point x_j in the relaxed feasible region which could be added to the master problem. This can only improve the solution if its reduced cost is negative (assuming a minimization objective of the original problem). Let c_j be the objective value for the solution x_j and h_{ij} be the value of the i^{th} complicating constraint for the solution x_j. Thus, the reduced cost is given as $c_j - \sum_i \lambda_i h_{ij} - \sigma$. Alternatively, the pricing problem for the column generation consists of finding a point x_j such that the reduced cost is minimized across the relaxed feasible region. If there is no such x_j with a desirable reduced cost, then the column generation procedure terminates. One key point is that c_j and h_{ij} can both be decomposed, thus when solving the pricing problem, one can exploit the block structure of the problem. The Dantzig-Wolfe decomposition algorithm is a very fascinating algorithm and one could write a whole paper on its workings. The interested reader is directed towards section 2.4 in Conejo et al. (2006) which provides numerous examples to help understand the algorithm. Note the importance of column generation in this algorithm. As mentioned, this helps in reducing the computational effort required to solve the model.

4.3.2. Benders' Decomposition

Complicating variables can also prevent the decomposition of a problem. Jacques F. Benders, a Dutch mathematician, devised a decomposition technique which can be applied to problems with complicating variables, it is known as Benders' (or Benders) decomposition (Benders, 1962). One key theorem is that if a problem has complicating variables, its dual will have complicating constraints (Conejo et al., 2006). Therefore, given a problem with complicating variables, one can formulate the dual problem and apply the Dantzig-Wolfe decomposition to the dual problem. In fact, Benders' decomposition is very closely related to the Dantzig-Wolfe decomposition, applying the Benders' decomposition to a problem is equivalent to applying the Dantzig-Wolfe decomposition to the dual (Rahmaniani et al., 2017). In the Dantzig-Wolfe decomposition, one adds decision variables to the master problem whereas in Benders' decomposition, one adds constraints (called cuts) to the master problem.

The general idea behind Benders'decomposition is that the problem is split into a master problem and a sub-problem. The sub-problem is similar to the original problem except the values for the complicating variables are fixed. In each iteration, the sub-problem is solved and based on the solution new constraints (cuts) are added to the master problem. Then the master problem is solved again (with the new cuts), doing this will give new values for the complicating variables. These new values for the complicating variables are used to solve the sub-problem again and so on. The solution of the sub-problem is the upper-bound and the solution to the master problem is the lower bound. If the difference between the upper bound and the lower bound is within a pre-specified tolerance level, then the algorithm terminates.

Benders' decomposition is best explained with an abstract example taken from Pereira and Pinto (1991). Consider the linear problem shown in Equation (4.5).

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 \\ & A_1x_1 \geq b_1 \\ & E_1x_1 + A_2x_2 \geq b_2 \\ & x_1, x_2 \geq 0 \end{aligned} \tag{4.5}$$

The decision variable x_1 can be assumed to be the complicating variable. This linear problem can be split into two problems, a master problem and a sub-problem. These are shown in the set of Equations 4.6. The dual formulation of the sub-problem is also given.

$$\begin{array}{lll} \textbf{Master Problem} & \textbf{Sub-Problem} & \textbf{Dual of Sub-Problem} \\ \min \quad c_1x_1 + \alpha & \min \quad \alpha(x_1) = c_2x_2 & \max \quad \pi(b_2 - E_1x_1) \\ \quad A_1x_1 \geq b_1 & \quad A_2x_2 \geq b_2 - E_1x_1 & \quad \pi A_2 \leq c_2 \\ \quad x_1 \geq 0 & \quad x_2 \geq 0 & \quad \pi \geq 0 \end{array} \tag{4.6}$$

Note that the master problem is similar to the original problem but it only has the constraints related to the complicating variable x_1. It also has a decision variable α which essentially refers to c_2x_2. The α forms the link between the master problem and the sub-problem. Note that the sub-problem is a function of x_1. This is because the value of x_1 is fixed when solving the sub-problem, there is a x_1 term on the right hand side of the constraint. Thus, one needs a value of x_1 to solve the sub-problem, that is, the complicating decision variable value has to be fixed.

The algorithm starts by generating a trial value for x_1, this value is used to solve the sub-problem. Let the trial value of x_1 be denoted x_1^1 to indicate that it is the trial value for the first iteration. Thereafter, the sub-problem $\alpha(x_1^1)$ is solved, it has a solution and its corresponding dual problem also has a solution, this is denoted by π^1. The 1 super-script indicates that it was the dual solution in the first iteration where $x_1 = x_1^1$. Note that if the initial value of x_1 was different, the dual solution π^1 could also have been different. However, the feasibility of π does not depend on the value of x_1 as x_1 is not in the constraints of the dual problem. Therefore, one can say that given π^1 is a dual solution when using $x_1 = x_1^1$, it will also be a feasible dual solution for other values of x_1. Furthermore, by the weak duality theorem, one can assert that $a(x_1) \geq \pi(b_2 - E_1x_1)$ for all values of x_1. Thus, since π^1 is a feasible solution to the dual, regardless of the value of x_1, combing this fact with weak duality allows us to construct the constraint $a(x_1) \geq \pi^1(b_2 - E_1x_1^1)$. This constraint is added to the master problem in the form $\alpha \geq \pi^1(b_2 - E_1x_1^1)$. This is known as an optimality cut. In the next iteration, the master problem is solved (with the new optimality cut) and a new value for x_1 will be generated, this will be used to solve the sub-problem again. This entire process is repeated until the difference between the upper and lower bound is less than the tolerance. Note that it could be that the sub-problem is infeasible. If this is a case, a different type of constraint would be added to the master problem to prevent the infeasibility, this is called a feasibility cut. The interested reader can find a more extensive and intuitive understanding of Benders' decomposition in Murphy (2013). This work omits rigorous mathematical proofs and is also accessible for readers with a limited math background.

Over the years, there have been numerous variations and extensions of the Benders' decomposition method, for instance applications to non-linear and integer problems. Furthermore, generating cuts can be inefficient and lead to many iterations, researchers have focused on techniques to improve the optimality and feasibility cuts. Rahmaniani et al. (2017) provides a good literature review on the various applications of Benders' decomposition and its extensions.

4.4. Stochastic Programming

Uncertainty and randomness are a part of everyday life. This also holds for optimization models. Many future events are not known at the present and this can be a cause of uncertainty in the optimization model. The classic example is that of a farmer who must decide today how much of a crop to grow. The yield and ultimate profit of the farmer depends considerably on future uncertainties such as the weather. Although, the weather can be predicted quite accurately for a short forecast horizon (1-2 days), it is quite difficult to do this accurately for a long-time horizon required for farming. Therefore, the weather is an uncertainty in the optimization model of the farmer.

Stochastic programming presents techniques which aim to solve optimization problems in which some parameters in the model are uncertain. The uncertainty can be in the objective function parameters or in the constraints. In order to solve the problem, it is essential that the uncertainty can somehow be expressed mathematically. This can be done by defining (joint)-probability distributions of the uncertain parameters. Alternatively, one can also define a set of possible future scenarios and incorporate them in the model. These scenarios represent the possible future outcomes of the uncertainty. Incorporating many scenarios in the model will make it more realistic, however it will also make the model difficult to solve.

The decision on whether to include uncertainty in an optimization model depends primarily on what is being modelled. There are broadly 4 ways to address uncertainty in optimization. The first is to assume that uncertainty does not exist. Expected values of the uncertain parameters are used when solving the model. This approach assumes that the optimization model is deterministic. The upside of this approach is that this greatly simplifies the model, it will reduce the computational effort required to solve the model. The downside is that the actual realization of the uncertainty will likely deviate from the expected realization used when solving the model. This can adversely impact the feasibility of the solution. In real life, an infeasible solution can lead to high costs.

The second approach is to assume a worst-case scenario and solve the model accordingly. Doing this ensures that the solution is feasible for all possible scenarios in the future. However, since the worst-case scenario is assumed, it is likely that the solution will be expensive. The other two approaches are chance constraints and multi-stage stochastic programming models.

4.4.1. Chance Constraints

The third approach is to use chance constraints. In this method, the modeller ensures that the solution is feasible with a pre-defined probability. For instance, the modeller can choose 95% and ensure that the solution is feasible for 95% of the time. This corresponds to a probability of failure of 5%. Requiring a lower probability of failure will increase the cost of the solution. Having a probability of failure of 0% is analogous to modelling using the worst-case scenario approach. The cumulative distribution function is used to model chance constraints.

The cumulative distribution function (cdf) of random variable ω is denoted as $F_\omega(x)$. It represents the probability that ω is less than or equal to x (denoted at $\Pr(x \geq \omega)$). the inverse cumulative distribution function (also called quantile function) is denoted as $F_\omega^{-1}(p) = x$, it takes in a probability p and gives a number x, such that ω is less than or equal to x with a probability of p. Suppose an optimization model has the following constraint: $a_1x_1 + a_2x_2 \geq \omega$, where ω is random. A chance constraint with confidence level ϵ requires the constraint to be met with a probability of ϵ, in this example it can be written as $\Pr(a_1x_1 + a_2x_2 \geq \omega) \geq \epsilon$. This can also be written in terms of the cdf as $F_\omega(a_1x_1 + a_2x_2) \geq \epsilon$. Taking the inverse cdf on both sides of the equation gives $a_1x_1 + a_2x_2 \geq F_\omega^{-1}(\epsilon)$. Some probability distributions have a closed form expression for their inverse cdf. Suppose, in the example, that ω

is normally distributed with mean μ_ω and variance σ_ω^2. The inverse cdf can be written as equal to $F_\omega^{-1}(\epsilon) = \mu_\omega + \sigma_\omega \Phi^{-1}(\epsilon)$ where Φ^{-1} is the inverse cdf of the standard normal distribution (mean equal to 0 and variance equal to 1), it can be calculated analytically. The Φ^{-1} is often used in hypothesis testing and building confidence intervals. For example, a confidence level ϵ of 80% would lead us to defining the constraint as $a_1x_1 + a_2x_2 \geq \mu_\omega + \sigma_\omega \times 0.8416$. Once the confidence level is decided, the constraint is deterministic. In Python, the function "norm(x)" in the scipy.stats module can be used to compute the $\Phi^{-1}(x)$.

There are two potential drawbacks of using chance constraints. First, not all probability distributions can be incorporated in chance constraints. However, the normal distribution, which is a very common probability distribution, can be incorporated in chance constraints. Second, the constraints could make the feasible set non-convex. This can make it difficult to solve the linear model (de Oliveira et al., 2016).

4.4.2. Multi-Stage Stochastic Programming

One can also deal with uncertainty by using a multi-stage stochastic programming approach. The main idea behind this approach is to make an initial decision and consequently as more information is revealed (i.e. uncertainty is revealed), take additional (corrective)-decisions. The general idea behind the two-stage stochastic program is as follows. An initial decision is made using the information available at present. There are numerous future scenarios which can materialize. Once a scenario materializes itself, a second decision is made. The objective is to minimize the cost of the initial decision and the expected cost of the future decision. Note, the objective can also be changed into, for instance, maximizing a profit.

If the future uncertainty is continuous in nature, one can discretize the uncertainty with a fixed number of scenarios and their respective probabilities. More scenarios will allow the model to better incorporate the uncertainty, however it will also make the model larger and more time-consuming to solve. In mathematical jargon, the discretization of the future scenarios is necessary in order to derive the "deterministic equivalent form" of the stochastic programming problem (Murphy, 2013).

An abstract optimization model for the two-stage stochastic programming problem is presented in Equation (4.7).

$$
\begin{aligned}
\min \quad & \overbrace{c^T x}^{\text{first-stage}} + \overbrace{\sum_{\omega \in \Omega} p(\omega) \cdot q(\omega)^T y_\omega}^{\text{second-stage}} \\
& Ax \leq b \\
& T(\omega)x + W(\omega)y_\omega = h(\omega) \qquad \forall \omega \in \Omega \\
& y_\omega \geq 0 \qquad \forall \omega \in \Omega \\
& x \geq 0
\end{aligned}
\tag{4.7}
$$

The set Ω is the set of all possible future scenarios. The vector x is the vector of first stage decisions, y_ω is the vector of second-stage decisions that are made if scenario ω materializes. The model aims to minimize the cost of the first-stage decisions and the expected cost of the second-stage decisions. This is why the probability of scenario occurring (i.e. $p(\omega)$) is also in the objective function. The model also has first-stage constraints and second-stage constraints. Specifically, $T(\omega)$ is called the technology matrix and $W(\omega)$ is known as the recourse matrix. The cost $q(\omega)$ is known as the wait-and-see cost. Note that these parameters are denoted to depend on the uncertainty ω, but depending on the problem, they can also be independent of the uncertainty.

The structure of the model shown in Equation (4.7) is similar to the model shown in the sub-section about Benders' decomposition (see Equation (4.5)). In fact, the first stage decision variables x in Equation (4.7) can be seen as complicating variables in the model. Therefore, Benders' decomposition is an useful algorithm to solve two-stage stochastic programming models. Van Slyke and Wets (1969) was one of the first papers to use decomposition techniques to solve stochastic programming, therefore another name for this technique is the L-shaped method.

The general idea behind the L-shaped method is the same as Benders' decomposition. An initial choice for the first-stage decision variables x (i.e. complicating variables) is made. This is used to solve the sub-problems, each scenario ω gives its own sub-problem. If any of the sub-problems is infeasible with the current first stage solution, then feasibility cuts are added to the master problem. Thereafter, the master problem is solved again, this should give different values for the first-stage decision variables. This process is repeated until all the sub-problems are feasible. After feasibility cuts, optimality cuts are added to the problem to ensure that an optimal solution is reached. Note that unlike the earlier discussion (Section 4.3.2) on Benders' decomposition, here there are multiple sub-problems. Therefore, one can add cuts for each sub-problem (multicut approach) or for each iteration (unicut approach). In the unicut approach, the cuts for all the sub-problems are combined into one cut and added to the master problem.

Two-stage stochastic programming problems can be extended to have multiple stages. At each stage, more of the uncertainty is revealed and thus another decision can be taken. The general idea behind a multi-stage stochastic programming problem is shown in Figure 4.3.

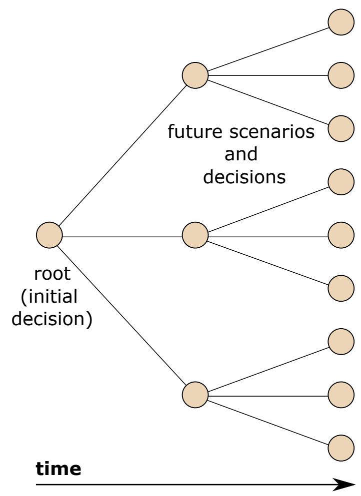


Figure 4.3: Multi-Stage Stochastic Programming Problem (Murphy, 2013, adapted from Figure 18)

An initial decision is taken. Multiple stages of future scenarios occur. Additional decisions can be taken at each of the future scenarios. Nested Benders' Decomposition (also known as Nested L-Shaped Method) can be used to solve a multi-stage stochastic programming problem (Birge and Louveaux, 2011). Observe that any consecutive pairs of stages can be seen as an isolated two-stage stochastic programming problem. Therefore, Nested Benders' Decomposition aims at applying the Benders' Decomposition multiple times (Murphy, 2013). The working principle behind this method is shown in Figure 4.4.

The method starts with an initial first-stage solution. This is used to solve the second-stage sub-problems. This will generate feasibility and optimality cuts for the root node master problem. Thereafter one can either continue deeper in the tree if there are any feasible second-stage decisions. The problem at the first stage will become the master problem and the third stage problem will be the sub-problem. Alternatively, one could also just go back to the root node from the first stage and re-solve the master problem with the new cuts that were generated from the first stage sub-problems. A good way to think of this method is as follows, cuts are passed backwards from the sub-problems to the master problem(s) and solutions are passed from the master problem(s) to the sub-problems. Note that there is flexibility in how one traverses the tree. One can either keep going deeper in the tree until the sub-problem is infeasible or one can always aim to first re-solve the master problems with the cuts. A combination approach can also be applied. There is no general rule, however different ways of traversing the tree will have an impact on the execution time of the algorithm. Murphy (2013) gives an extensive overview of using Benders' decomposition for stochastic programming problems. The interested reader is advised to refer to it for a more comprehensive understanding of Nested Benders' decomposition.

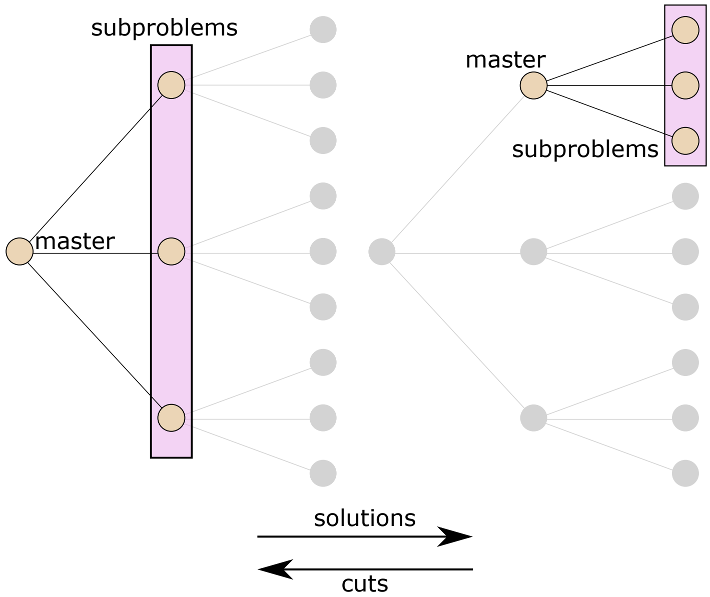


Figure 4.4: Working of Nested Benders' decomposition (Murphy, 2013, adapted from Figure 19)

There are two drawbacks of using multi-stage stochastic programming models. First, the future uncertainty is modelled using discrete scenarios. It could be that the actual realization of the uncertainty is not included in any of the modelled scenarios. The solution may not even be feasible for the actual realized uncertainty. Second, the problem may have integer decision variables, such a problem is known as the multistage stochastic integer programming problem (MSIP, Zou et al. (2016)). Due to the integrality constraints, the problem loses its convexity. The Benders' decomposition technique relies on using dual variables to generate the cuts, however these do not exist for problems with integer variables. Solving MSIPs is very challenging and is a topic for research. There are some basic techniques which have been developed. They primarily rely on relaxing the integrality constraints and branching techniques. They also have different cut generating techniques and add additional types of cuts to the problem. The interested reader can find more information about this in Chapter 7 of Birge and Louveaux (2011).

4.5. Concluding Remarks

This chapter gave a concise overview on various optimization techniques. Some of these techniques can form the backbone of the proposed computational tool. Using column generation and decomposition techniques will help reduce the computational time required to solve the problem. Given the uncertainty involved in the belly-space and cargo demand, it will be useful to use some of the stochastic programming tools (chance constraints and multi-stage) discussed in this chapter. To do this however, one needs to first be able to model these uncertainties in a mathematical manner. This is the topic of the next chapter.

5

Modelling Uncertainties

There can be two main sources of uncertainty in the cargo operations of combination airlines. First, the supply side uncertainties represent the uncertainties in the cargo carrying capacity of the air carrier. Section 5.1 describes various ways which have been used to model these supply side uncertainties. Second, the demand side uncertainty represents the uncertainty in the demand for air cargo carrying capacity from the airline. Section 5.2 gives an overview of the various methods used to model demand side uncertainties.

5.1. Supply Side Uncertainties

There can be numerous sources of supply side uncertainties. The most dramatic one is perhaps when a flight gets cancelled or an airport temporarily shuts down (e.g. due to bad weather). In such a case, the air cargo carrier cannot fly the aircraft and thus it cannot transport the cargo. Belly-space uncertainty is more common and relevant for combination carriers. In passenger aircraft, the belly refers to the lower deck of the aircraft where the passenger check-in cargo is stored during flight. It can also hold additional cargo if space permits. Therefore, combination airlines can use the available belly-space to transport cargo shipments. The exact weight and dimensions of passenger check-in cargo is uncertain until the passenger arrives at the airport. This causes uncertainty in the available belly-space for cargo transport. There are numerous ways to model this uncertainty.

Berdowski et al. (2009) conducted a survey to investigate the weights of passengers and their baggage at major European airports. There were eight airports in the study, Amsterdam Schiphol Airport (AMS) was also part of this study. Arriving passengers and their baggage was weighed at the baggage reclaim area. Similarly, departing passengers were weighed as close to the departure gate as possible, their check-in bag was weighed at the check-in counter. Additional information such as age, gender, season (Winter or Summer), route type (domestic, European or non-European) etc was also gathered. The results of the survey are shown in Table 5.1. The table shows the mean check-in baggage mass for various passenger types. It also gives the number of observations (n) and standard deviation. The confidence range can be used to construct a confidence interval (= sample mean $\pm$ confidence range).

This survey data can be used to design a probability distribution for the weight of passenger check-in cargo. Based on the number of passengers on a flight and belly capacity of an aircraft, one can model the available belly-space as a random variable with a specific probability distribution. Thereafter, one can either use chance constraints or discretize the probability distribution and create scenarios for multi-stage stochastic programming.

One drawback of this study is that it was conducted between 2008 and 2009. This means that the study is rather old. Nevertheless, it gives an quantitative insight into the uncertainty of checked luggage. This insight can help to model the total passenger cargo on flights and the resulting capacity available in the aircraft's belly to carry cargo.

Table 5.1: Checked baggage masses (in kg) by season and gender (Berdowski et al., 2009, table 4.9)

Season	Passenger Type	Mean	n	Std.dev.	Conf. Range (95%)
Summer	Male	16.9	5,162	5.8	0.16
	Female	17.0	4,172	5.7	0.17
	Child (2-12 years)	14.2	327	6.0	0.65
	Male & Infant (<2 years)	19.9	13	7.1	3.83
	Female & Infant (<2 years)	17.2	18	7.9	3.67
	Total	16.9	9,692	5.8	0.12
Winter	Male	16.5	7,391	5.9	0.13
	Female	16.8	5,08	5.7	0.16
	Child (2-12 years)	17.1	138	6.2	1.03
	Male & Infant (<2 years)	19.8	17	6.7	3.18
	Female & Infant (<2 years)	18.8	35	5.4	1.79
	Total	16.6	12,661	5.8	0.10
Total	Male	16.7	12,553	5.9	0.10
	Female	16.9	9,252	5.7	0.12
	Child (2-12 years)	15.1	465	6.2	0.56
	Male & Infant (<2 years)	19.8	30	6.7	2.41
	Female & Infant (<2 years)	18.3	53	6.3	1.71
	Total	16.7	22,353	5.8	0.08

Another way to model the capacity uncertainty is by using scenario trees. This was done in Delgado et al. (2019). This is a discrete time process to model the uncertainty. The idea behind a scenario tree was briefly explained in the section on stochastic programming (Section 4.4). The scenario tree approach is best explained using an example. Suppose there are 4 flights (**f1**, **f2**, **f3** and **f4**) and they depart in this respective order. Assume that the belly-space capacity of flights **f1**, **f2** and **f4** is uncertain, but it can only take two values. It can either be a optimistic value which means that there is relatively ample belly-space on the flight, or a pessimistic value which indicates that less belly-space is available. The belly-space of **f3** is assumed to be deterministic and known beforehand. The departure moment of each aircraft is essentially when information is revealed about the belly-space for that aircraft. Thus, each stage in the stochastic programming problem corresponds to the departure moment of a flight.

The scenario tree of the example is shown in Figure 5.1. It begins with the root node. At this point, no information is revealed yet about the available belly-space on the flights. At the next stage, flight **f1** is ready to depart and the available belly-space on this flight can either be a optimistic value or a pessimistic value. This results in two child nodes from the root node, one for the optimistic case and another one for the pessimistic case. Afterwards at the next stage, flight **f2** is ready to depart and its belly-space is revealed. This results in two child nodes for each node of stage **f1**. Since **f3** has a deterministic available belly-space, it is also grouped with flight **f2** in the same stage. Finally, flight **f4** is ready to depart, this results in two child nodes for node in stage **f2, f3**. A scenario is defined as a path from the root node to the last node. The dotted line in the Figure 5.1 represents scenario 3, here the belly-space on flights **f1**, **f2** and **f4** is optimistic, pessimistic and optimistic respectively. Thus, scenario trees can be used to discretize future possible scenarios that may occur in a discrete time stochastic process.

Note that the method used to model the uncertainty has a bearing on the algorithm that will be used to solve the model. For instance, using scenario trees will mean that one has to model using a multi-stage stochastic programming approach. Solving such problems can be challenging and this should be considered when deciding on how to model the uncertainty.

5.2. Demand Side Uncertainties

The demand for air cargo is uncertain. To understand this, one must understand the cargo booking dynamics in the air cargo industry. Although, different air cargo carriers will have their own policies

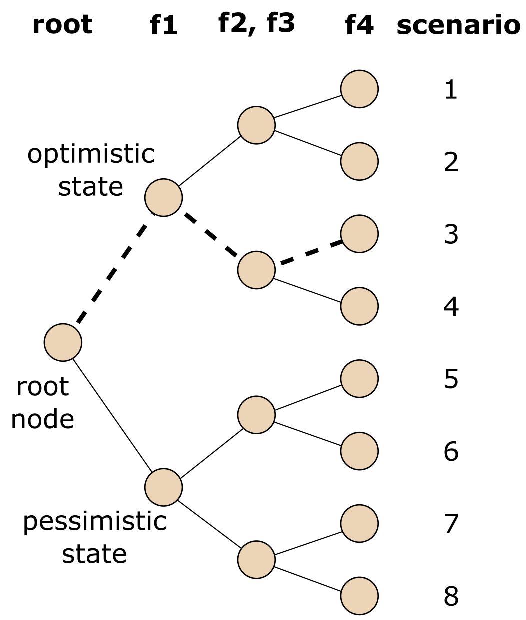


Figure 5.1: Scenario tree for belly-space capacity (Delgado et al., 2019, adapted from Fig. 2)

regarding cargo booking, the general structure is similar. Air cargo carriers sell cargo capacity either through long/medium term contracts or on the spot market. In the long/medium term contracts, freight forwarders can reserve cargo capacity on a particular route for particular flights/days of the week. The duration of such a contract varies between a few months to a year. Such contracts are typically negotiated with freight forwarders who have an established relationships with the air cargo carrier. Often, the freight forwarder will show up with less cargo than was originally booked. In fact, sometimes the freight forwarders may not even show up at all with cargo.

This is a problem for air cargo carriers as the unused capacity could have been sold to someone else. Air cargo carriers are reluctant to penalize freight forwarders for low capacity/no-show, largely because these freight forwarders have market power; they are large customers and they can start to look for other air cargo carriers if they are being penalized (Boonekamp et al., 2013). One solution for air cargo carriers is to overbook their flights. Many carriers overbook their flights in anticipation of the low capacity/no-show of freight forwarders. However, a downside of this is that in some cases the flight is actually overbooked and it cannot deliver some cargo. It will then have to pay compensation to the customer whose cargo it could not deliver.

Thus, the uncertainty in the show up capacity of freight forwarders combined with the overbooking practice of air cargo carriers causes uncertainty in the actual amount of cargo that will need to be transported. Furthermore, in the spot market, parties can book cargo capacity on a flight one-week to a few hours before departure. This short time period also introduces uncertainty in the actual amount of cargo to be transported. Lastly, there is also uncertainty in the dimensions of the cargo. Since a cargo has both volume, shape and weight, it could be that the cargo has awkward dimensions which prevent it from being packed on the flight. These are the main sources of demand disruptions for the air cargo carriers.

Delgado et al. (2020) and Delgado and Mora (2021) look at recovering from demand disruptions on an air cargo network. Both papers employ nice ways to model these demand uncertainties. These uncertainties are referred to as demand disruptions in these papers. Delgado et al. (2020) starts off by generating R cargo requests. In the paper, the authors used instances with $R = 20$ and $R = 30$. Each request is generated by randomly selecting a source airport and destination airport. Thereafter, the weight of the request was drawn from a uniform random distribution. The release date and due time for the request were also randomly selected. Thereafter, R^- requests were removed from the list of R requests, and R^+ new requests were appended to this list. The R^- varied between 7 and 15, whereas the R^+ varied between 7 and 22. Thus, if $R = 20$, $R^- = 7$, $R^+ = 10$, the model would first generate 20 requests, remove the last 7 requests and then add 10 new requests. The R^- essentially denotes the number of no-shows and the R^+ denotes the number of last minute requests.

Delgado and Mora (2021) has a similar way to model the demand disruptions. There is a base case of orders; consequently there can be disruptions to these orders. There can be two types of disruptions, either the original orders see a change in their weight or there are new orders added to the model. This is different as compared to Delgado et al. (2020), where it only considered no-shows and last minute requests. The updated weight of an order could be also zero, implying that the order is a no-show.

Both the papers only consider the disruptions in terms of changes in the weight of the cargo or last minute requests for cargo. They do not consider the disruptions caused, for instance, by volume changes. Doing this simplifies the problem, however it also makes it less realistic. Furthermore, both the papers are deterministic in nature, they assume that the demand disruptions are known and the models all aim to reschedule the flights to deal with the disruptions.

Some papers have aimed to model the cargo booking demand in a stochastic manner. Delgado et al. (2019) considers the cargo demand to be stochastic and models it using a normal distribution. Thereafter, this demand is incorporated in the model by using a chance constraint. Similarly, Han et al. (2010) aims to create a Markovian model to determine if cargo booking requests should be accepted or not. It assumes that the cargo booking requests have a weight and volume which is log-normally distributed. The cargo booking demand weight (measured in kg) is log-normally distributed with parameters μ and σ being equal to 6.237 and 0.938 respectively. The volume (measured in cubic meters) is log-normally distributed with parameters μ and σ being equal to 0.943 and 1.082 respectively (Han et al., 2010, from table 2).

5.3. Concluding Remarks

This chapter gave an overview of the techniques that can be used to model supply side and demand side uncertainties. Supply side uncertainties can be modelled with chance constraints or scenario trees. Demand side uncertainties can be modelled either by chance constraints or generating demand disruptions. The next chapter looks at how past work has incorporated these uncertainties when designing models to route cargo and cargo aircraft.

6

Cargo Carrier Optimization

This chapter presents various models that have been used in the planning process of air cargo carriers. Section 6.1 gives an introduction to four phase airline planning process. Section 6.2 gives an overview of the first papers which looked at optimizing air cargo operations. Section 6.3 looks at deterministic models which can be used in planning air cargo operations. Section 6.4 looks at stochastic models which can be used in planning air cargo operations.

6.1. Airline Planning Process

Chapters 6 & 7 of Belobaba et al. (2009) and Tamura and Inano (1987) provide a good overview of the different types of optimization models used in the scheduling and routing process of airlines. There are broadly 4 sequential steps required to successfully design the operations of an air carrier. Each step has its own mathematical model and solution technique(s). They are solved in the order described below, where the results of the previous model are the inputs for the next model.

1. **Schedule design problem:** The aim is to design an airline's flight schedule. To do this, the airline must perform route evaluation to judge which routes it should fly. Thereafter, it must determine how often it should fly these routes; this is known as frequency planning. Having an higher frequency reduces wait times, the airline may capture more traffic because of demand stimulation from the higher frequency. Afterwards, the airline must make a flight schedule of flight legs, with each leg having an origin airport, destination airport, departure time and arrival time. This is known as timetable setting. In most cases, airlines make incremental schedule changes as opposed to designing a new schedule from scratch. The former is computationally easier than the latter. Also, this ensures that there is come consistency in the new airline schedule.

2. **Fleet Assignment:** Given the aircraft schedule and airline fleet, this fleet assignment model (FAM) aims at assigning an aircraft type to these flight legs. The aims in this model are to minimize the spillage and reduce operating costs (conversely increase profits). Spillage occurs when the aircraft cannot accommodate the total demand for the flight leg. The airline must fly a larger aircraft to minimize the spillage. However, these larger aircraft have higher operating costs and thus a balance must be found. Time-space networks are used in fleet assignment, these help in representing the various flight legs. Each node in such a network presents a particular airport at a specific point in time. There are numerous constraints in the FAM. For instance, balance constraints ensure that the total number of aircraft of each type arriving and departing an airport are equal.

3. **Assigning Aircraft to Routes:** Given the flight schedule and the fleet assignment, the next step is to assign specific aircraft to the flight legs. A combination of flight legs to be operated by an aircraft is known as its route. Maintenance requirements are the main constraints in this problem. Aircraft have maintenance requirements, aircraft need to be at specific maintenance locations at specific times. The aircraft routing must ensure that these maintenance requirements are met. Grönkvist (2005) mentions that the aircraft assignment approach varies per airlines. Some

airlines plan their aircraft assignments well in advance, whereas others plan it quite late. This largely depends on the type and size of the air carrier. Airlines with a regular schedule, large fleet and large maintenance hangar capacity can afford to perform aircraft assignment quite late. Also, there are various names for the models which assign aircraft to the flight legs. These models have subtle difference in their objectives and inputs. Models include “Aircraft Routing”, “Aircraft Rotation Problem”, “Maintenance Routing Problem” and “Tail Assignment Problem”.

4. **Crew Pairing and Assignment Problem:** The final step is to assign crew to the flights. This is done by first constructing duty periods, this is a sequence of flights a crew member can do in a day. These duty periods are combined to form pairings. A pairing starts and ends at the base of the crew. The crew assignment problem assigns these pairings to individual crew members. There are various ways to do this. For instance in the US, the crew can bid for pairings and these are often granted based on the seniority of the crew member (Kasirzadeh et al., 2017). The crew pairing and assignment problem is not the focus of this research. The reader is referred to Kasirzadeh et al. (2017), this provides a comprehensive literature review on various crew scheduling models.

The above four phase planning process is a very broad characterization of the planning process. In fact, there are many papers which have been published for each phase. In fact each phase deserves a literature review on its own. The four phase planning process is also used by cargo carriers (Antes et al., 1998; Zils, 2000), however the number of papers constructing planning models for use in freight transport is a lot less as compared to passenger transport (Derigs and Friederichs, 2013). There are differences in the planning approach between cargo carriers and passenger airlines. For instance, the route selection for passenger airlines requires a long-term planning horizon. However, for cargo carriers, the fluctuating demand requires them adjust routes in short time horizon (Yan et al., 2006).

This research pertains primarily to the first three aspects of the planning process. Selecting the freight routes, the fleet flying the route and the aircraft rotations to successfully plan cargo operations which can augment the cargo carrying capacity of the passenger network for a combination airline. Since the research objective assumes that the planning of passenger flights is decided beforehand and cannot be rerouted, the focus of this chapter is the planning and routing for cargo carriers. Furthermore, the aim is to look at papers which propose integrated methods that combine schedule design, fleet assignment and rotation planning. The crew planning process is important but not relevant to this research. It is assumed that given a flight schedule, the airline will be able to successfully assign crew to the flights.

6.2. Historical Perspective on Air Cargo Planning

Chan and Ponder (1979) talks about the growth of the air freight industry in the United States of America. Specifically, it focuses on the operations of a then young start-up called Federal Express Corporation (FEC), today this company is the well-known FedEx Corporation. The paper talks about the operations which made FEC standout from its competitors. At the time, FEC focused on the overnight small-package delivery market. They mention the benefit of FEC having its own pick-up and delivery trucks. A key point in the paper was the hub-and-spoke system of FEC, with the hub located in Memphis, Tennessee. Since the point-to-point freight demand between two airports is not very high, operating a hub-and-spoke system allowed FEC to “bundle up the traffic, to cut down per-unit cost” (Chan and Ponder, 1979, page 224). The hub-and-spoke system also allowed FEC to deliver freight to smaller cities, where the total freight demand is likely not very high. For instance, at that time, 70% of the traffic carried by FEC was to cities outside the largest 25 metropolitan areas in the United States.

Chestler (1985) talks about the overnight small package delivery industry in the United States. It gives an overview of the industry and then specially analyses the impact of the hub location on the operations. There are many factors to be considered when deciding the location of a hub. The paper mentions that one of the reasons why FEC chose Memphis as its hub is because it has nice weather, thereby preventing unfavourable weather which can ground aircraft. However, because Memphis is rather far from the North-East and Midwest of the United States, FEC has to fly larger planes and this adversely affects the load factor. Competitors which closer hubs to the North-East can fly smaller planes and achieve an higher load factor. The paper also mentions the “bypass operation” that FEC uses to optimize operations. This is performed if there is sufficient demand between two cities A and B. At the source A, all the freight for destination B is loaded in the same container. When the flight from airport A reaches the hub, the container gets directly loaded on the aircraft for airport B. Thereby, by passing the sorting facility at the hub airport.

Marsten and Muller (1980) propose mixed-integer programming models for air cargo fleet planning. Assuming that the network, freight demands and aircraft fleet are known, the model aims to select the aircraft types to fly the routes such that the profit is maximized. This study was done for the "Flying Tiger Line", which was a an American cargo airline founded in 1945. The paper proposed three different models. The first model was applied to networks with a single hub. The second model was applied to networks with multiple hubs. Both of these networks operated in the night. The last model looked at a two-network system where there was one night network followed by a secondary daylight delivery network. This daylight delivery network has its own demand but it also aims to delivery any leftover freight from the night operations. The authors describe the network using the term "spider graph". This is essentially a graph where a node with a degree larger than two is considered as the hub node. The various models were coded in Fortran and solved using branch-and-bound. The authors conclude that the models are useful as a planning tool to test the effect of various network designs and aircraft characteristics on air cargo fleet planning.

Kuby and Gray (1993) compared two different hub-and-spoke network designs for air freight carriers. The first was a pure hub-and-spoke network design. Here, all the spokes were connected only to the hub. The second network is a hub-and-spoke with stopovers and feeders. Here, not all spokes are connected to the hub, some of them are connected to other spokes which are then connected to the hub. Doing this allows for less links in the network which essentially means less aircraft are used to traverse the network. Some spokes may not have enough freight demand to warrant a direct flight to the hub. By having stopovers, the aircraft can collect more freight and fly with an average higher load factor. Also, in some cases, the aircraft can directly deliver cargo from an previous spoke on the trip. For instance, if there is cargo to be delivered from nodes 1 to node 2 (assume these are nearby each other), taking the cargo from node 1 all the way to the hub and then sending it back to node 2 is quite inefficient. In the network with stopover model, this does not happen and the cargo gets directly delivered to node 2 from node 1. The paper compares the pure hub-and-spoke model with the hub-and-spoke with stopovers model. Data is used from the operations of Federal Express Company (FEC)'s operation in Western U.S. The authors find that the pure hub-and-spoke network would cost 73.59% more to operate than the hub-and-spoke with stopovers. Also, the average load factor on the pure hub-and-spoke model was 43.73%, whereas it was 74.7% for the hub-and-spoke with stopovers. This shows that having stopovers in the hub-and-spoke network of an air freight carrier can help reduce costs and boost the load factor. Although the exact increase depends on the airports and demand in the underlying network, a significant increase is likely.

6.3. Deterministic Models

This section gives an overview of papers which have developed deterministic models for the planning of air cargo carriers. This means that the total cargo to be carried, the profit and costs are all inputs to the model. Some of the papers develop the planning for cargo carriers from scratch whereas others reroute aircraft based on new information inputs. Furthermore, some papers aim to reroute & reschedule cargo and cargo flights based on demand disruptions that are caused by uncertainties.

Yan et al. (2006) designed a mixed integer program to construct a scheduling model which aims to combine airport selection, fleet routing and timetable setting for a cargo airline. This paper is very similar to their earlier paper (Yan et al.) which focused on airport selection, fleet routing and timetable setting for a combination airline. Another paper by the same author(s) (Yan and Chen, 2008) looked at optimal flight scheduling models for cargo carriers which are in an alliance with other cargo carriers. All the papers used the same basic solution technique. A time-space network forms the backbone of the model. A time space network has one axis to denote the time and another axis denotes the location. A node on this network represents an airport at a particular point in time. The paper has two types of time-space networks, one represents the flow of aircraft (called the fleet flow time space network) and the other represents the flow of cargo (called the cargo flow time space network). Each network has movement arcs (e.g. cargo movement, flight movement), holding arcs and cycle arcs. The arcs have a cost and an upper & lower bound on the capacity. The time space networks are used to construct sets which are then used to formulate a mixed-integer linear program with the objective to minimize the transportation costs. This large problem is NP-hard, solving this problem to optimality would require a prohibitively long amount of time.

The authors develop a family of heuristics to solve this large problem. In essence, all the heuristics aim to reduce the number of variables in the decision space in order to simplify the problem. For instance, the authors consider the “non-stop heuristic”, where all the flights considered are non-stop flight. Similarly the “one-stop heuristic” considers all flights with a maximum of one layover. Also, the “mixed-stop heuristic” looks at the origin-destination distance and determines if the flight should have no-stop, one-stop or multiple stops. Origin-destination pairs which are relatively close to each other will only be allowed to have non-stop flights, whereas other origin-destination pairs which are further away from each other may be allowed to have a layover(s). A final “improved mixed-stop heuristic” improves on the mixed-stop heuristic by solving a series of networks with different simplifications (non-stop, one-stop etc).

The model was run on a real life dataset from a major Taiwan airline. The results show that relying on a purely non-stop network is not advisable. All the other heuristics performed quite well and considerably reduced the problem size as compared to solving the entire problem with all the variables.

The main takeaway from this paper is the construction of the time-space networks and the various heuristics that can be used to solve the problem. The main drawback of the paper is that it considers a single fleet airline. Having multiple types of aircraft will result in a more complex problem as fleet assignment also needs to be done.

Derigs et al. (2009) build an integrated model which combines flight selection, rotation planning and cargo routing for air cargo operations. The model takes as an input a list of cargo flights. Each flight has a origin, destination, departure time and arrival time. Some of the flights have to be performed (mandatory) while others are optional. The aim of the model is to select the optimal set of flights to be performed, select the aircraft which will perform them and also route cargo through these flights so that it reaches its destination. The model's objective is to maximize the network-wide profit. This profit is composed of the revenues earned when serving the cargo demand and the various fixed and variable costs transportation costs for operating the aircraft and network.

Cargo units that are transported from an origin to a destination all traverse a list of flight legs. An origin-destination path is defined as a sequence of flight legs that a cargo unit can take to go from its origin to its destination. Every origin-destination pair has a set of origin-destination paths. The main decision variable is f_p which denotes the total flow (essentially total weight) of cargo that is transported on path p. Every path also has a margin m_p which denotes the revenue minus variable costs associated with transporting one unit of flow on path p. Therefore, term $m_p f_p$ represents the total margin (revenue - variable costs) for transporting cargo on path p. The model aims to select the origin and destination paths which will help maximize the network-wide profit. There are various constraints which ensure that the solution is feasible in real life, this includes ensuring that the total cargo transported is less than or equal to the cargo demand, flights are not over-loaded etc. The paper also formulates another model which is similar to the first model, except it also incorporates aircraft rotations in the model. The aim is then to select the optimal origin-destination paths and the aircraft rotations. The aircraft rotations are modelled as a binary variable, x_r is 1 if rotation r is selected.

Instead of running the model with all possible origin-destination path, the model uses column generation to construct valid origin-destination paths. The method aims to solve a shortest path problem to generate the origin-destination path columns. A similar method is used to generate columns for the model with aircraft rotations. Although, since this is a binary variable, one has to use branch-and-price to solve it. This method combines column generation with branch-and-bound. This does lead to relatively longer computational times as compared to the original model with only the origin-destination paths.

The authors designed a data generator to construct problem instances. All problem sets were solved in a reasonable amount of time, although the model with aircraft rotations took longer. The main takeaway from this paper is how to use column generation in the network planning for air cargo carriers. The main drawback of the paper is that it assumes a single fleet type for the aircraft. Some cargo carriers operate multiple types of aircraft and thus fleet assignment will also need to be incorporated. Doing this will make the model more realistic, albeit with an increase in the model size and computational effort required to solve the model.

Delgado et al. (2020) look at how to recover from demand disruptions on a cargo network. These demand disruptions are caused due the uncertainty in cargo booking demand described earlier (see Section 5.2). The authors note that there have been many papers which have studied the passenger airline recover problem, there one aims to minimize the passenger delay given an aircraft cannot perform its assigned flight leg(s). However, this paper was one of the first papers which looked recovery for freighters. The authors define the air cargo schedule recovery problem (ACSRP), this problem aims to reschedule freighter aircraft and reroute cargo given demand disruptions on cargo booking. The aim is to minimize the costs of rescheduling and rerouting the aircraft. The original routing of the aircraft is known. At the end of the recovery horizon, the aircraft should be a specific airport as defined in its original schedule.

The authors use a time-space network to formulate the problem. The network has two kinds of nodes, itinerary nodes denote airports for every time period. Request nodes denote the origin nodes and destination nodes of the request combined with the release and due-time of the request. The network has four types of arcs. Flight arcs connect two airports in the network. Ground arcs connect two nodes on the same airport, it represents the time an aircraft or a request spends on the ground at a particular airport. No-service arcs connect the origin nodes to the destination nodes of a cargo booking request. Request access nodes connect the request nodes to the itinerary nodes. The arcs have a cost, for instance the cost of a flight arc represents the cost of operating the aircraft. The authors also define a penalty function which denotes the cost of rescheduling, it focuses on the additional costs the airline incurs in reassigning its crew to the new schedule. The authors define three different ways to design the penalty function, each is based on a different crew management policy that an airline can have. The ACSRP is modelled as a mixed integer linear programming problem, with the objective to minimize the additional costs such as the new flight costs, cargo booking holding costs and the penalty costs of reassigning the crew.

The model was applied to the set of airports in North and South America. The model used different instances, these denoted different original schedules before the recovery horizon. Most of the parameters such as costs, fleet size were obtained from various airlines. Benchmark solutions were computed, these represented the case where the booking requests could be rerouted or cancelled, but the aircraft could not be rescheduled. The computational results showed that by rerouting aircraft and cargo, one can reduce the operating costs by 10% as compared to the benchmark solution. The average computational time was 16 minutes, this is reasonably short.

This paper showed that cost savings can be realized if one reschedules aircraft and reroutes cargo. One drawback of this paper is that it only assumed one set of demand disruptions, in real life the demand is more dynamic with disruptions that happen every day. One may have to solve this problem multiple times and not only once. This will bring with it more challenges.

Delgado and Mora (2021) use a metaheuristic approach to solve the air cargo recovery problem under demand disruption. It is similar to Delgado et al. (2020) described earlier. It aims to make last-minute adjustments to the schedule of cargo flights in order to recover from demand disruptions which arise due to the uncertainties associated with cargo booking demand. It also uses a time space network approach, with nodes and arcs. The linear model has a block structure with a complicating constraint. The model can be decomposed with sub-problems for each aircraft in the set of aircraft. There is a one complicating constraint which aims ensures that the total cargo that is picked up by any flight is less than or equal to the total demand for the cargo transport. Since the model has a complicating constraint, the authors use the Dantzig-Wolfe decomposition (see Section 4.3.1) to solve the problem. There is a restricted master problem (RMP) which has the complicating constraint and each cargo aircraft has its own sub-problem. The binary constraints in the model are relaxed for restricted master problem. Solving the restricted master problem gives dual variables which are used to solve the sub-problems. The solution of the sub-problems leads to adding columns in the RMP. Once no more columns can be added to the RMP, the binary constraints are reinstated and the problem is solved with all the generated columns. As mentioned, each aircraft has a sub-problem, the goal of a sub-problem is to find a optimal route for that aircraft. It uses a heuristic which starts with an initial route along with the list of orders which are picked up on that route. It then aims to change the route based on swapping orders, adding new orders or changing the position when the order is picked up on the route.

The performance of the metaheuristic was compared to two benchmark solutions. One benchmark solution solved the same problem but using a commercial Mixed Integer Programming (MIP) solver. This amounts to solving the problem with all possible columns. In the other benchmark solution, the aircraft could not be rerouted, only the cargo could be rerouted. Using metaheuristics was much faster than solving it in a conventional manner. Also, allowing the aircraft to be rerouted increased profit by 10-15% (depends on the instance) as compared to only allowing the cargo to be rerouted.

This paper showed that decomposition techniques can be used to recover from demand disruptions. They also significantly reduce the computation time. The drawbacks of this study are similar to that of Delgado et al. (2020). Actual demand disruptions are more dynamic in nature. One may need to solve this recovery problems multiple times.

6.4. Stochastic Models

Delgado et al. (2019) solves a multi-stage stochastic programming model which aims at allocating cargo to the belly-space of passenger flights. The available belly-space is uncertain and so is the cargo booking demand. The paper uses a scenario tree to model the available belly-space. This uncertainty modelling approach was discussed earlier in Section 5.1. The uncertainty in the cargo booking demand is modelled with a normal distribution and added as a chance constraint in the model. This was also discussed earlier in Section 5.2. The model assumes that the uncertain booking demand arises from spot market cargo bookings. This multi-stage stochastic programming model is solved using the Nested Decomposition technique (described earlier in Section 4.4).

Unlike other stochastic programming problems in fields such as energy planning, the stochastic programming models in air transport are not Markovian. Part of a cargo request could have been transported earlier and thus at every stage one must know how much cargo was already transported earlier in order to know how much cargo is still left to be transported. The lack of the Markovian property has implications for the Nested Decomposition method. Specifically, this has an implication on cut generation. Past decisions are required to be stored in order to generate the cuts.

The model is applied on medium and large instances which model 1-day operations of a commercial partner in South America. The large network has 103 requests that have to be transferred using 52 flights, 7 of the flights have uncertainty. An solution example will help understand what the model aims to do. Consider the network shown in Figure 6.1. This is a sub-network from a real solution. The arcs

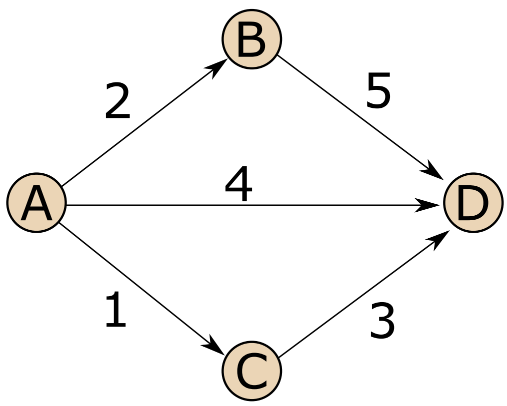
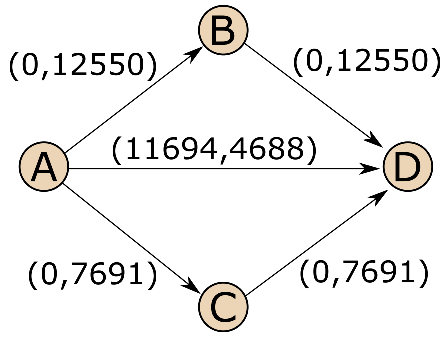
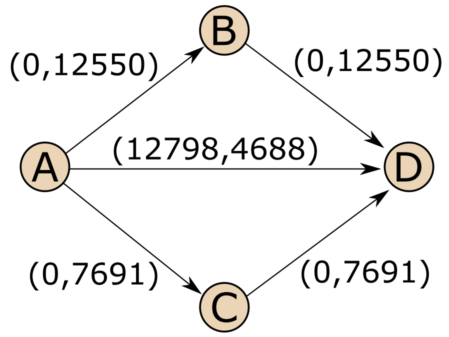


Figure 6.1: Example of sub-network (Delgado et al., 2019, adapted from Fig. 7)

represent the flight (denoted by numbers) and the nodes represent airports (denoted by letters). The belly-space capacity on flights 1 through 4 is uncertain. The uncertainty can either be optimistic (O), pessimistic (P) or average (A). There are two cargo requests (c_1 and c_2) that have to be transported. In the first stage decision, the model decided to transport 12798 kg and 24929 kg of c_1 and c_2 respectively. Figure 6.2 shows the actual network capacity based on two uncertainty realizations. The tuple next to the flight arcs denotes how much of c_1 and c_2 was carried by that flight. Figure 6.2a shows the situation



(a) P,P,*,P



(b) P,P,*,O

Figure 6.2: Two uncertainty realizations (Delgado et al., 2019, adapted from Fig. 8)

when the capacity on flight 1, 2 and 4 is pessimistic and the capacity on flight 3 is any realization (optimistic, pessimistic or average). The asterisk ("*") denotes any realization. Observe that even though the first stage decision decided to transport 12798 kg of c_1, it is only able to transport 11694 kg of c_1. The leftover 1104 kg cannot be transported and the airline will pay a penalty of this. Figure 6.2b shows the situation when the uncertainty realization is "(P,P,*,O)" (pessimistic, pessimistic, any, optimistic) for flights 1 to 4 respectively. Since flight 4's capacity is optimistic, it can carry more cargo. Hence, the 1104 kg of c_1 which could not have been transported in the case of "(P,P,*,P)" can be transported now since the carrying capacity for flight 4 is higher. The solutions show that the uncertainty realizations have a large impact of the solution, changes in the available belly-space capacity can lead to changes in the routing of multiple cargo requests. This shows that the model is able to effectively make routing decisions based on new information from the stages.

This model is aimed solely at routing cargo when the belly-space capacity of the aircraft is unknown. The model is able to provide near-optimal solutions for every belly-space capacity realization. Unlike the other models discussed earlier, this model does not reroute aircraft. Rerouting aircraft often requires combinatorial optimization techniques, these use binary and integer variables. These variables complicate the Nested Decomposition solution technique as the convexity of the problem is lost (refer to discussion in Section 4.4). Also, this model requires a relatively long time to execute. It takes 16 hours and 26 minutes to construct the cuts and it takes 9 minutes to solve the problem. This long execution time is not desirable when wanting to implement such a model in the daily operations of an air carrier.

6.5. Concluding Remarks

This chapter gave an overview of the airline planning process and also described various solution procedures that can be used in the integrated planning of cargo carriers. It looked at both deterministic models and stochastic models. Some deterministic models were also able to use column generation and the Dantzig-Wolfe decomposition in order to solve the problem. These techniques can help reduce the computational time and make the model applicable in real life operations. The use of a stochastic programming technique was also seen in this chapter. However, this did take a relatively long time to execute. The next chapter looks at weight and balance control, this is essential when loading and flying the aircraft.

7

Weight and Balance

Weight and Balance is an important part of air cargo operations. The total cargo loaded on the aircraft must be maximized, while ensuring that the aircraft is able to take-off. This imposes a constraint on the total weight of the cargo loaded on the aircraft. Furthermore, loading cargo on the aircraft shifts the centre of gravity of the aircraft. The loaded aircraft must have a centre of gravity which lies within safety limits. Also, cargo flights aim to have multiple stopovers in order to better utilize the aircraft. Therefore, it is desirable to load the cargo on the aircraft in a way that minimizes the handling requirements when loading and offloading the cargo at intermediate airports.

This chapter gives an overview of the relevant published work in the field of weight and balance control of aircraft. Section 7.1 describes the centre of gravity, its importance and how it is measured for aircraft. Section 7.2 gives a historical perspective on the first papers which explored weight and balance control. Section 7.3 lists the more recent papers in the field of weight and balance control.

7.1. Importance of Center of Gravity

The aircraft centre of gravity (cg) is an imaginary point where the entire aircraft weight can be assumed to act from. The point is three-dimensional, it has a longitudinal, lateral and vertical coordinate. However, the longitudinal location of the cg point is the most relevant when dealing with the stability and balance of the aircraft (Administration, 2016). The empty aircraft will have a certain cg location and this will shift as payload is loaded on the aircraft and as it burns fuel during flight. The computational way to measure the location of the cg is composed to three key steps. First, the total weight of the empty aircraft and all the items (e.g. fuel, cargo etc) put inside in it are calculated. Similarly, the total moment of the empty weight and all the payload items is calculated with respect to a reference datum; this can be the nose of the aircraft but can also differ among manufacturers. The moment arm is the length on the longitudinal axis. The cg location is found by dividing the total moment by the total weight (Administration, 2016).

The cg location is often expressed as a percentage of the aircraft's Mean Aerodynamic Chord (MAC). This is illustrated in Figure 7.1. The chord is a line between the leading edge and trailing edge of the wing. For most wings, the chord is not constant across the span of the wing. The average chord across the wing is known as the MAC. The cg coordinate is projected on the MAC; the distance behind the leading edge of this projection is Xt. The cg location is then expressed as a percentage of Xt divided by the length of the MAC. For most modern commercial aircraft, the cg location is between %20MAC and %30MAC (Liu et al., 2018).

The aircraft has forward and aft cg limits. The cg of the aircraft must be inside these limits during the flight. If the cg is too far aft (i.e. tail heavy), the aircraft is unstable. A too forward cg (i.e. nose heavy) will require more tail load to maintain level flight (Limbourg et al., 2012). This will increase the drag and consequently the fuel consumption (Sabre Airline Solutions, 2007). Liu et al. (2018) built a model to investigate the relationship between the cg location and fuel consumption. This model was applied to a B737-800 aircraft. The paper found that, for a given cruise Mach, the drag increases if the cg

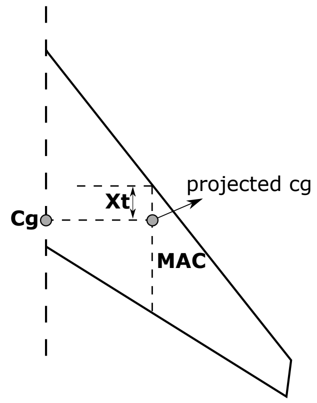


Figure 7.1: Center of Gravity Measurement (Liu et al., 2018, Figure 1)

location moves more forward (i.e. %MAC decreases). The higher drag will require more thrust from the engines, thereby increasing the fuel consumption. This study suggests that aircraft operators may desire having a more aft cg location, within the stability margins, as this decreases fuel consumption. One way to do this is by employing the appropriate aircraft loading strategy.

7.2. Historical Work on Weight and Balance

Martin-Vega (1985) provides a good overview of the various methods used in aircraft load planning before 1985. Around this time, manual loading planning was the norm, however computers were slowly being incorporated in the load planning process. There are two key insights made by the author. First, given the intricacy of the task, manual load planners' primary aim was to obtain a feasible load plan. Striving for optimality was a secondary aim. Second, manual load planning can be assisted by using computers but it cannot entirely be automated. There will always be a need for an experienced load planner.

Manual load planners had to take many factors into account such as dimensions of the cargo and the centre of gravity of the aircraft. Obtaining a precise centre of gravity location is difficult for a manual load planner. Therefore, they used the "pyramid loading" technique when loading the aircraft. In this technique, the heaviest items were placed in the centre of the aircraft and the load planner would work his/her way outwards by placing the next heaviest items alternately towards the front and back of the aircraft. This technique ensures that the centre of gravity of the aircraft during and after loading does not deviate significantly from the centre of the aircraft.

Larsen and Mikkelsen (1980) was one of the first papers to address the loading of cargo on aircraft. The authors created a tool called CARLO (CARgo LOading) which aimed to create feasible load plans given information about containers and pallets to be loaded. The authors created two heuristics which aim to minimize the handling costs at intermediate airports while respecting safety and feasibility constraints. The user of CARLO would have the choice of selecting either heuristic to create the load plan. The paper does not give a detailed description of how the heuristics work. The first heuristic is called the "INTERCHANGE" heuristic. It has a 3-phase working process. First, all the constraints are relaxed, and a load plan is created. In phase 2, the heuristic checks if the constraints are satisfied with the current load plan. If a constraint is not satisfied, then the heuristic proceeds to phase 3 where it performs pairwise interchanges between loaded cargo (i.e. swaps) in order to satisfy the violated constraint. Phase 2 is repeated after each successful interchange. The other heuristic is the "PUT" heuristic. Here, the constraints are never relaxed. The space available (slack) in each compartment of the aircraft is used to determine which item should be placed in the compartment next.

The CARLO tool was coded in FORTRAN-IV and solved using real life data from Scandinavian Airlines System. Both the heuristics performed well and were able to give a feasible load plan in under 1 second.

There are two limitations with this paper. First, it does not give a comprehensive explanation of the workings of the heuristics. This makes it difficult to replicate and use these heuristics. The second limitation is that it assumes that there are only 2 legs in a flight (i.e. one intermediate airport). As mentioned in the introduction, freighters often have multiple stop-overs. Increasing the number of stop-overs would "exponentially increase" the computation time Larsen and Mikkelsen (1980).

Another notable work is Brosh (1981). This paper presented, perhaps, the first mathematical programming approach when solving the load planning problem. In the problem, the aircraft has 5 cargo bays (see Figure 7.2), each with its own weight and volume restrictions. Furthermore, each bay has its own relative centre of gravity. The cargo to be loaded is not packed in ULDs, rather it has a weight and a density; the density is used to create the volume constraints. The author solves two versions of the problem. In the simplified version, the load is assumed to be homogeneous (i.e. equal density) and the objective is the maximize the total load carried. In the advanced version, the load is heterogeneous and the objective is to maximize the total profit obtained when carrying the load.

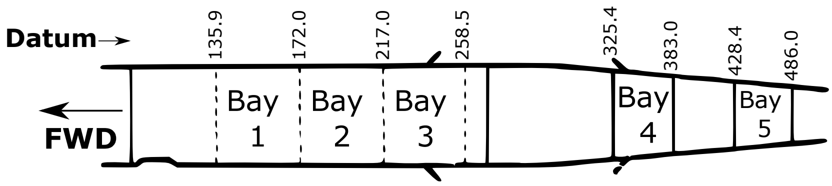


Figure 7.2: Plane layout (Brosh, 1981, adapted from Fig. 1)

The fundamental idea behind the solution approach is the centre of gravity envelope (see Figure 7.3). The loaded aircraft will have an associated total weight and centre of gravity location. This tuple should be in zone-1 of the centre of gravity envelope. The author approximates the boundaries of zone-1 region using piece-wise linear functions. These piece-wise linear functions are used to formulate constraints which ensure that the centre of gravity of the loaded aircraft is in zone-1 of the centre of gravity envelope. These constraints are non-linear. The solution is obtained by linearizing the non-linear constraints around an initial point Z and solving the model. The LP solution is used to generate a new approximating point, this is used to again linearize and solve the model in the next iteration. This is iteratively done until the solutions of the linear LP model converge with a certain tolerance δ. This solution technique was applied to a case study, the model took 3 iterations to converge.

As mentioned, Brosh (1981) was the first solution method which used mathematical programming to solve the aircraft loading problem. One drawback is that this model does not take into account ULDs (i.e. containers and pallets). Instead, the author assumes that the aircraft is loaded with different types of homogeneous cargo. This results in the model being a continuous optimization model (i.e. no integer/binary variables). Nowadays, a lot of the cargo is transported in ULDs and thus optimizing the placement of these ULDs within an aircraft is relevant. The ULDs themselves have their own set of constraints, for instance most ULDs can only be placed at specific locations on the aircraft. The papers introduced in the next section provide solution techniques which aim to optimize the placement of ULDs on aircraft.

7.3. Recent Work on Weight and Balance

Mongeau and Bes (2003) presents an optimization model which aims at loading ULDs to compartments in an aircraft. The aircraft is assumed to have multiple cargo holds which are subdivided into compartments. The ULDs are assumed to be of five types, some ULD types are small and thus require less space whereas others are large and require more space. Each compartment can hold different combinations of ULD types. For instance, a compartment may hold either 6 small ULDs are 2 large ULDs. Furthermore, every cargo hold has a maximum weight that can be loaded in it and the entire aircraft also has a maximum allowable freight weight that can be loaded. The binary decision variable $x_{i,j}$ is 1 if ULD i is loaded in compartment j. It could be that some ULDs are not loaded on the aircraft at all and stay on the ground. The expression for the cg is given in Equation (7.1). Note that M_a is the

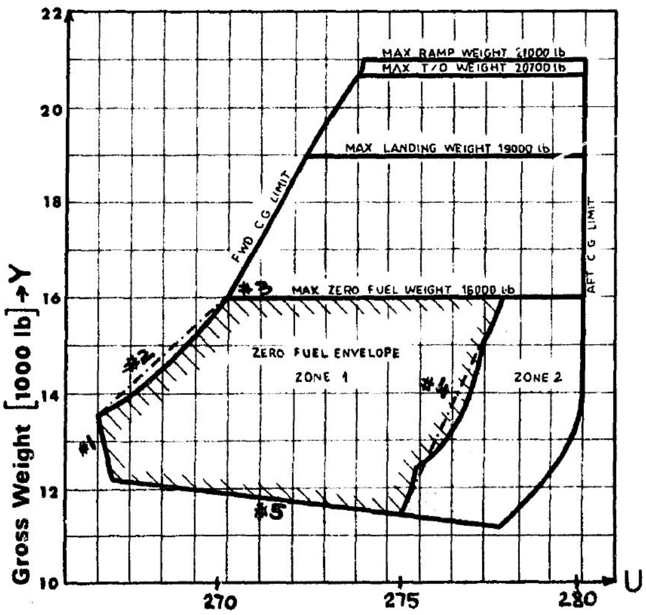


Figure 7.3: Center of Gravity envelope (Brosh, 1981, Fig. 2)

mass of the aircraft before loading, X_a is the longitudinal cg position of the aircraft before loading. M_i is the mass of ULD i and X_j is the longitudinal position of the geometric centre of compartment j.

$$
\begin{aligned}
M(x) &:= \sum_{i=1}^{N_{cont}} \sum_{j=1}^{N_{comp}} M_i x_{i,j} \\
CG(x) &:= \frac{M_a X_a + \sum_{i=1}^{N_{cont}} \sum_{j=1}^{N_{comp}} M_i X_j x_{i,j}}{M_a + M(x)}
\end{aligned}
\tag{7.1}
$$

Notice that expression in Equation (7.1) is derived by using the total moment over total weight method described in Section 7.1. It assumes that the cg of the ULDs all act through at the geometric centre of the compartment they are placed in. The expression for the cg location is non-linear as $x_{i,j}$ is both in the numerator and denominator of the expression. This non-linearity means that the cg expression cannot be directly used in the linear programming model. Instead, the model assumes that the cg must be at a predefined target cg location denoted by X_{target} with some allowable predefined tolerance ϵ. Therefore, the model uses the following constraint to ensure a desirable cg location is obtained:

$$
[M_a + M(x)][X_{\text{target}} - \epsilon] \leq M_a X_a + \sum_{i=1}^{N_{cont}} \sum_{j=1}^{N_{comp}} M_i X_j x_{i,j} \leq [M_a + M(x)][X_{\text{target}} + \epsilon] \tag{7.2}
$$

Note that the constraint in Equation (7.2) is linear. A similar constraint is also constructed to ensure that the cg location is ahead of the most aft allowable location. The objective of the model is to maximize the total weight loaded on the aircraft (i.e. $M(x)$ in Equation (7.1)) while striving for a optimal centre of gravity.

The model was run on six different test problems. Furthermore, the model was tested with two different cg deviation tolerance levels (i.e. ϵ). The runtimes were all under 10 minutes, this indicates that it is feasible to use this model in a real-life setting. Also, higher tolerance levels required more CPU time. Given that the paper was published in 2003, the increased computer power available today will likely require less time to solve this model.

There are some limitations to the paper. The centre of gravity is calculated based on information about each loaded compartment. However, each compartment itself can have multiple ULDs. Hence, the calculated centre of gravity is less precise than if it has been calculated by looking at the location of each ULD. Furthermore, the model does not take into account the loading/unloading effort required when an aircraft has multiple legs.

Limbourg et al. (2012) create a mixed-integer programming program to solve the weight and balance problem. Given a set of ULDs ($\mathbb{U}$) to load and a set of loading positions ($\mathbb{P}$) on an aircraft, the model aims to assign each ULD to a loading position. All the ULDs have to be loaded on the aircraft, this is different than Mongeau and Bes (2003), where some ULDs could be left on the ground. Every ULD has a type and every loading position has a set of ULD types that can be loaded on it. The calculation of the cg is done using the same method as in Mongeau and Bes (2003); total moment divided by total weight. However, since all the ULDs have to be loaded (i.e. none can be left on the ground), the total weight of the loaded aircraft is known before the loading process. Therefore, the denominator in the cg equation shown in Equation (7.1) is a constant and the cg expression is linear.

Given the linearity of the cg expression, the authors could have minimized the deviation of the cg location with respect to a desired location to be the objective of the model. However, similar to Mongeau and Bes (2003), this was instead included as a constraint. The objective was to minimize the moment of inertia of the loaded aircraft. The expression for the moment of inertia is shown in Equation (7.3). The term ID is the desired location, a_j is the arm of position j, w_i is the weight of ULD i.

$$\min \sum_{i \in \mathbb{U}} \sum_{j \in \mathbb{P}} w_i (a_j - ID)^2 x_{i,j} \tag{7.3}$$

The rationale behind optimizing on the moment of inertia is as follows. Only optimising the cg location with not ensure a desirable load distribution. It could be that weight is packed at the forward and aft ends of the aircraft. This would give a reasonable net cg location, however this would cause increased stress in the fuselage structure. Minimizing the moment of inertia will ensure a better packing of the weight. Furthermore, a lower moment of inertia leads to more manoeuvrability as the aircraft becomes easier to rotate.

The paper also includes some other constraints. It ensures that the lateral deviation of the cg location from the central line (also known as buttock line zero) is within limits. Also, for structural reasons, there should not be too much loading at different sections of the aircraft. Therefore, another set of constraints ensure that the combined loading at different aircraft sections is within allowable limits. This assumes that the weight of the ULD is uniformly distributed within each ULD.

The model was coded in Java and solved using IBM ILOG CPLEX. The main case study was to load 42 ULDs on a Boeing 747 aircraft. The solution was compared to that devised by a trained loadmaster. The model provided a solution which is always at least as good as the trained loadmaster's solution. Furthermore, the model only took 2 seconds to find a solution, whereas the trained loadmaster took 20 minutes. In another problem instance, the objective was changed to minimizing the cg deviation. The solution was compared to the original version which minimized the moment of inertia. It was found that the model which minimizes the moment of inertia was 16 times faster than the model which minimizes the cg deviation. This led the authors to conclude the minimizing the moment of inertia is comparatively better at driving the optimization process.

As compared to Mongeau and Bes (2003), this paper implemented the more realistic constraints encountered when loading an aircraft. The model was fast and gave desirable results. However, the model assumed that all the ULDs could be loaded in the aircraft. Although, this may be true for most cases, it could be a limiting if the number of ULDs is larger than the space available on an aircraft. For instance, a lot of ULDs arrive at a hub in order to be flown to their final destination. It may be necessary to optimally select which ULDs will be loaded on the aircraft. Also, as with Mongeau and Bes (2003), the model does not consider the loading/unloading effort required when an aircraft has multiple legs.

Vancroonenburg et al. (2014) builds on Mongeau and Bes (2003) and Limbourg et al. (2012). The objective is to load ULDs to positions on an aircraft. Unlike Limbourg et al. (2012), not all ULDs need to be loaded and some can be left on the ground. There is a profit G_i for transporting ULD i. This model has a multi-criteria objective. The first objective (O_1) is to maximize the total profit of transporting the ULDs. The second objective is to minimize both the longitudinal and lateral deviation of the cg from a target location. This deviation is modelled as the difference in rotational moment generated from the desired and actual cg location. Furthermore, in the model, maximizing cargo value has a higher priority and thus has a higher weight in the objective function.

The model has similar constraints as seen in Limbourg et al. (2012). These include constraints on the maximum weight transported and cumulative load constraints ensure that the load at certain areas in the aircraft are within limits. The model also has unique constraints not seen earlier. For instance, some ULDs may have dangerous goods, these cannot be placed at certain loading positions. The decision variables $x_{i,j}$ are only generated if ULD i can be placed at position j. Also, the ULDs are all allotted a priority number. Lower priority ULDs can only be loaded if the higher priority ones have been loaded. Furthermore, some ULDs may not be placed next to each other in order to prevent contamination; for example ULD with radio-active elements next to ULD containing food. If i_1 and i_2 are ULDs that should not be placed close to each other, then the following constraint ensures their separation:

$$x_{i_1,j_1} + x_{i_2,j_2} \leq 1 \qquad \forall j_1, j_2 \in \text{loading positions} : j_1 \text{ close to } j_2 \tag{7.4}$$

The model is coded in Java and solved using Gurobi. The dataset consists of 51 flights that were operated by the commercial research partner. The flights were done using aircraft from the Boeing 747-400 series (747-400ERF, 747-400F and 747-400SF). The CPU time was less than 10 seconds for all the problem instances and in fact averaged at about 2 seconds. The results were also compared with that obtained by an expert load planner. The comparison showed that the model obtained better longitudinal and lateral cg locations. The objective of maximizing profit was also done better by the model.

This model can be considered an extension of the model presented in Mongeau and Bes (2003) and Limbourg et al. (2012). It incorporates more real world constraints, this makes the model more realistic. The multi-criteria objective function is also unique to this model. One common result is that the CPU time is not very long. This was observed in this model and also in Limbourg et al. (2012). This is likely because the real life instances are limited in size. There are only a handful of loading positions on an aircraft. The number of ULDs that need to be transported is also limited, most often less than 100. As with all the previous models, this model does not consider the loading/unloading effort required when an aircraft has multiple legs and the ground operations involved with it.

Lurkin and Schyns (2015) builds upon the models described earlier (Mongeau and Bes (2003), Limbourg et al. (2012), Vancroonenburg et al. (2014)). The unique contribution is that this model also considers flights that have 2 legs with an intermediate airport. ULDs can be loaded/offloaded at this intermediate airport. This intermediate loading/offloading increases the complexity of the task. First, ULDs that do not have to be offloaded can be in the path of ULDs that need to be offloaded. Therefore, the ULDs that do not need to be offloaded are still offloaded and then loaded again. This increases the handling costs. Second, even if a ULD is not in the path of another ULD, it could be that it has to be moved around at the intermediate airport in order restore the desired cg location for the next leg of the trip. This will also increase the handling costs. Therefore, the aim should be to load ULDs in such a way that they are not in the path of ULDs that get offloaded at the intermediate airport and that optimum cg locations can be reached without moving them for the subsequent flight leg.

The main decision variable in the model is a binary variable $x_{i,j,k}$ which is 1 if ULD i is loaded in position j for flight leg k. Notice the additional parameter k in the decision variable, this was not seen in the earlier models where the decision variable consisted of only two indices i and j.

For each loading position j, the authors identify the closest exit door towards the nose of the aircraft and the closest exit door towards the tail of the aircraft. Thereafter, for each loading position j, the authors create a set $\mathbb{B}_j^N$ which represents all the loading positions situated between loading position j and the exit door towards the nose. Similarly, the set $\mathbb{B}_j^T$ represents all the positions situated between loading position j and the tail door. This is illustrated in Figure 7.4. For loading position a, the loading positions with the gray circles belong to $\mathbb{B}_a^N$. Similarly, the loading positions with the black squares belong to $\mathbb{B}_a^T$. Also, consider loading position b. Since this loading position is behind the last door, it does not have a door behind it pointing towards the tail. Hence, the $\mathbb{B}_b^T$ will be an empty set. The loading position with the gray triangle will be in $\mathbb{B}_b^N$.

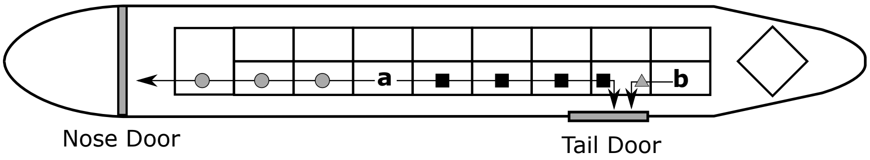


Figure 7.4: Different types of doors for the aircraft (Lurkin and Schyns, 2015, adapted from Fig. 4)

For all loading positions j, the model loops through the set $\mathbb{B}_j^T$ which represents all positions behind position j and the nearest tail door. If there is any loading position j' which belongs to $\mathbb{B}_j^T$ and is offloaded through the nose door, then the ULD at position j will also have to be offloaded. This is intuitively shown in Figure 7.5.

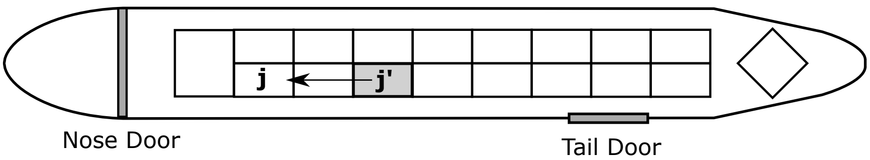


Figure 7.5: Unnecessarily unloading a ULD (Lurkin and Schyns, 2015, Fig. 5)

If the ULD at j' is to be offloaded at the intermediate airport through the nose door, then the ULD at position j would also have to be offloaded. Furthermore, if the current ULD at position j is a ULD whose destination is not the intermediate airport, then the offloading of the ULD at position j is deemed unnecessary.

The multi-criteria objective of the model is to minimize the sum of cg deviations across all the flight legs and the unnecessary offloading of ULDs. The cg location is found using the method described earlier; total moment divided by total weight. The model has several constraints which have been seen in the earlier models. The model was implemented on real-world instances provided by TNT-airways. The result was compared to that obtained by a load planner. The model solution is superior to the one obtained by the load planner. The cg deviations are significantly lower, so are the unnecessary offloading at intermediate airports. For most problem instances, the solution was found in under 10 minutes. This runtime is significantly higher than the computation times seen earlier, those were often under 1 minute. This is likely due to the increased size of the problem when considering the handling operations at intermediate airports.

Brandt (2017) presents a model which combines the main elements from Limbourg et al. (2012), Vancroonenburg et al. (2014) and Lurkin and Schyns (2015). Additionally, the model has some unique features not seen before. It allows for multiple legs (Lurkin and Schyns (2015) used 2 legs). It also allows the user to define loading positions which must be used before some other loading positions are used. For instance, loading positions next to the cockpit must be used if the positions behind them are also to be used. In the case of an accident (e.g. detachment of ULDs from loading positions), the ULDs directly in front of the cockpit will provide protection to the cockpit. This model is tested on various instances, 59% of the instances require a run time of 30 seconds or less.

7.4. Concluding Remarks

This chapter looked at the published research on weight and balance control when loading an aircraft. The papers presented had different assumptions and complexities. Some of the older papers did not use ULDs, whereas the newer models use ULDs in the formulation of the model. It seems that there are two main decisions that have to be taken before constructing a weight and balance model. First, will all the cargo have to be loaded on the aircraft or can some of it be left on the ground. Second, will the model consider intermediate loading/offloading of ULDs, doing this can lead to a better solution, albeit with a larger and slower model.

8

Potential Solution Approaches

Given the trove of literature presented in the preceding chapters, there are two potential approaches that can be taken in designing the proposed computational tool. The first approach can be classified as a predominantly reactive approach. It is described in Section 8.1. The other approach can be considered as a predominantly proactive approach. This approach is described in Section 8.2.

8.1. Predominantly Reactive Approach

In the predominantly reactive approach, the model would start with a routing for the cargo and cargo aircraft. As the disruptions occur, the model would seek to make changes to the initial routing in order to deal with the disruptions. This model is predominantly reactive as it waits for the disruptions to occur and reacts to them by changing the routing of the cargo and cargo aircraft. There can be some proactivity in the model by using chance constraints to model the passenger aircraft belly-space capacity. By using these chance constraints, the model is able to predict (with some confidence level) the available belly-space that it can use to transport cargo. Such a model can also be iteratively run whenever a new set of disruptions occur. This model would be Markovian in nature as the new routing would depend only on the current routing and the new disruptions.

Such a predominantly reactive model can be solved by conventional linear programming techniques. However, the approaches of Derigs and Friederichs (2013) and Delgado and Mora (2021) described earlier suggest that column generation and decomposition techniques can be used. This will reduce the solution tool runtime, thereby increasing the applicability of the model and also facilitate the integration of this model into the real life operations for a combination airline.

The advantage of such a model is that it will be able to deal with various kinds of disruptions. Furthermore, the solution techniques (e.g. column generation, decomposition techniques) are well established and have been used before for various kinds of models. This ensures that there are many resources that can be consulted when designing the solution tool. The main disadvantage is that the model is very reactive and it will have to be run multiple times to accommodate all the disruptions as they occur. Also, it seems intuitive that if the model is able to proactively set up a robust initial base routing, then the disruptions that occur in the future will be easier to absorb (Delgado et al., 2020). Since this reactive model will not do that, it is possible that the final solution is worse than a more proactive approach.

8.2. Predominantly Proactive Approach

The predominantly proactive approach would be to first decide on a set of scenarios that may occur. Based on the probability of the scenarios, one can set up the routing of cargo and cargo aircraft which best deals with the scenarios. This approach is akin to the multi-stage stochastic programming approach described earlier. Although this model is predominantly proactive, it also has a reactive component. At every stage when the disruption occurs, the model can take corrective (recourse) actions to deal with the disruptions. The main solution technique for this kind of problem is the Nested Decomposition solution technique.

The main advantage of the model is that it is predominantly proactive. As mentioned before, having a proactive approach will likely lead to a better solution to the problem. There are some disadvantages with this approach. First, one has to enumerate all the possible scenarios that can occur. This enumeration will always be a subset of all the possible scenarios that can occur in real life. Therefore, it is possible that a scenario occurs which the model had not considered and thus it cannot adequately respond to this scenario. This can be especially true for cargo demand. Its uncertainty can be hard to predict. Perhaps this is why most of the prediction models shown in the section on modelling cargo demand uncertainty (Section 5.2) had used deterministic modelling techniques. Also, multi-stage stochastic programming problems are very challenging to solve. This is complicated by the fact that routing aircraft problems typically involve integer/binary variables, this removes the convexity of the problem. It is even more challenging to apply Nested decomposition techniques for non-convex problems.

8.3. Integration of Weight and Balance

Most of the models described in Chapter 6 had a rudimentary way for ensuring weight and balance control. In fact, these models just ensured that the aircraft maximum payload weight was adhered to when loading the cargo. Some also ensured that the total volume of the cargo was less than the total volume available for cargo on board the aircraft. Most did not consider balance requirement of the aircraft. It is hoped that the proposed model can ensure balance control. Thus, the model will aim to transport all possible cargo that is available to be loaded on the aircraft, while ensuring that the loaded aircraft's cg location is within safety limits. The model may leave some cargo on the ground in order to ensure a feasible loading of the aircraft.

For the purposes of this research, it seems excessive to consider the more advanced weight and balance control measures. For instance, loading the aircraft in order to optimize handling costs at intermediate airports increases the computational time. Therefore, this can be omitted in the modelling process. Furthermore, the model will also not account for cg changes due to the burning of fuel during flight. It is assumed that if the aircraft has a safe cg position at take-off, this cg position will also be safe during all stages of the flight.

8.4. The Way Ahead

The main difference between the predominantly reactive model and the predominantly proactive model is that the proactive model will perhaps give a better solution although it will take more effort to solve. Since this research also wants to ensure that the computational tool can be applied by the combination airline, having a tool which takes a long time to return a solution is not desirable. Furthermore, given the timeline of the project, it seems prohibitively challenging to build a computational tool using the predominantly proactive approach. Therefore way ahead is to build the tool with the predominantly reactive approach.

The project will be divided in an initial phase and a final phase. Each phase will be roughly 3 months. An overview of this plan is given in Figure 8.1. The aim of the initial phase is to construct a model which

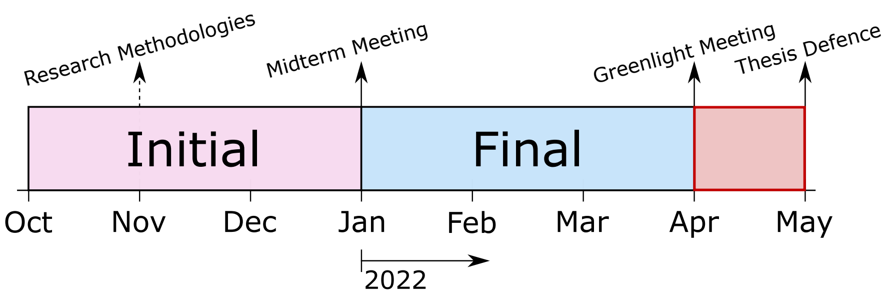


Figure 8.1: Overview of the Plan

can generate the problem instances. Thereafter, the solution tool will be constructed and tested on a small instance. The aim of the final phase is to add improvements to the solution tool which may have been recognized in the initial phase. A key desire is that the solution tool should have a short runtime.

Therefore, an effort will be made to optimize the model and speed it up. Thereafter, the solution tool will be tested on larger and more realistic instances. This will also help gauge the performance of the solution tool. In the end, the solution tool and its workings will be presented in front of a grading committee.

9 Conclusion

A combination airline operates both a passenger network and a cargo network. It aims to transport cargo in the lower deck (belly) of its passenger aircraft and also in dedicated freighter aircraft. The operations of a combination airline is complicated by supply side and demand side uncertainties. Supply side uncertainties refer primarily to the uncertainty in the belly space of the passenger aircraft. The check-in cargo of passengers has priority over other cargo on a passenger aircraft. The check-in cargo of passengers, and thus also the belly space that can be used to transport cargo, is only known shortly before departure of the passenger flight. Also, the cargo booking process is a complex process which brings with it a lot of uncertainty. The exact amount of cargo requests and the size of these requests is only known shortly before these requests arrive for departure. This leads to demand side uncertainties.

Given these operational uncertainties, the research objective of this project is defined as follows:

Research Objective

The objective of this research is to reroute & reschedule cargo requests and cargo aircraft in a cost-effective manner with a tactical perspective (3 to 7 days). This will be accomplished by designing a computational tool which can proactively and reactively deal with the supply and demand side uncertainties.

Two research questions can be formulated which will help guide this research process. The research questions are:

Research Questions

Question 1: How can a combination airline incorporate current and potential disruptions, caused by uncertainties, in a computational tool in order to cost-effectively reroute and reschedule its cargo requests and cargo flights?

Question 2: What is the improvement in performance for a combination airline by introducing a computational tool to reroute and reschedule its cargo requests and cargo flights given current and potential disruptions caused by uncertainties?

9.1. Sub-Questions

A number of sub-questions were formulated in order to guide the literature review process. These sub-questions were addressed in the various chapters of this literature review. The following is a short answer to each sub-question.

What are the key performance indicators which can help evaluate the performance of the computational tool and the performance of the cargo carrier?

There are numerous key performance indicators (KPI) that can help judge the performance of a cargo carrier. These indicators can be split along their key performance areas (KPA). The most common operational KPIs used in cargo operations are the revenue tonne kilometres (RTK) and the available tonnes kilometres (ATK). The former seeks to measure how much cargo was carried, while the latter seeks to measure how much cargo carrying capacity was delieverd by the cargo carrier. The ratio of the RTK to the ATK gives the load factor. Cargo carriers should aim to increase their load factor. Furthermore, there are also key financial performance indicators such as revenue, profit and cost. These are often expressed as a ratio of the RTK or ATK. The environmental KPIs such as total fuel cost are also relevant, especially as the world aims to reduce greenhouse gas emissions in a bid to tackle climate change.

Finally, the computational tool itself should have a quick runtime, this will help integrate it into the day-to-day operations of the combination airline. The tool should give near optimal solution, this can be measured by looking the optimality gap. Lastly, since the uncertainties change, the tool should ensure that a certain success ratio is reached in rerouting the cargo. As an example, 95% of the cargo (including with the uncertainties) should be able to reach its final destination.

What are the most relevant optimization techniques which can be used in constructing the computational tool?

The world of operations research has seen tremendous progress over the years. There are a host of optimization techniques that are available and can be used as the backbone of the proposed computational tool. Branch and bound is a very useful technique to solve mixed integer problems. It aims at first relaxing the integer constraints, solving the problem and then branching on variables which should be integer but are currently not. Column generation is a technique which aims to solve large linear problems. It aims to first start with a subset of decision variables and only add decision variables to the problem which will help improve the solution. This is done by solving a pricing problem. In the end, the final model contains only a subset of all decision variables. Since the number of decision variables is less, the runtime of the model reduces.

Decomposition techniques aim at decomposing the problem into smaller sub-problems. These smaller sub-problems are easier to solve. Often however, a problem cannot be fully decomposed into smaller sub-problems. Complicating constraints and variables are two complications that prevent a perfect decomposition. There are numerous techniques which aim to deal with these complications in order to make the model decomposable. The Dantzig-Wolfe decomposition aims at solving problems with complicating constraints. It relaxes the complications constraints and aims to construct a convex combination of the corner points of the relaxed feasible region such that the final solution is optimal and respects the complicating constraints. Benders' decomposition is used to solve problems with complicating variables. This method divided the problem into a master problem and sub-problems. The value of the complicating variables is an input to the sub-problems. Solving the sub-problems gives cuts which are added to the master problem. Solving the master problem with the new cuts gives new values for the complicating variables which can be used to solve the sub-problem again. This iterative process continues until the difference between the upper bound and lower bound is less than or equal to a predefined tolerance.

Stochastic programming techniques aim to deal with models which have stochasticity in their parameters. Chance constraints can be used to construct constraints which have an uncertain right hand side. These use the cumulative distribution function (cdf) of the uncertain parameter to build the constraints. Chance constrains ensure that the constraint is met with a certain confidence level (e.g. 95%). Multi-stage stochastic programming models aim to make a first stage decision and subsequent decisions as uncertainty is revealed. They aim to minimize the total cost of making a first stage decision and the expected cost of making future decisions. The mathematical structure of a multi-stage stochastic programming problem is very similar to that of a model with complicating variables. Therefore, Benders' decomposition can be used to solve multi-stage stochastic programming problems. If there are multiple stages in the problem, one can use Nested Benders' decomposition to solve the problem. This iteratively applies Benders' decomposition to various master problems and sub-problems. Solving multi-stage stochastic programming problems is challenging, this challenge is amplified when dealing with integer/binary decision variables. These variables remove the convexity of the problem, they do not have dual variables which can be used to add cuts to the master problem.

How can the supply side and demand side uncertainties be modelled?

Supply side uncertainties can be modelled by using chance constraints or scenarios trees. Using chance constraints requires one to have a realistic probability approximation of the uncertainty. Surveys conducted at airports have useful data which can be used to construct this probability model. Using a scenario tree is another approach, the aircraft belly-space is given a set of values it can take. Each stage in the scenarios tree corresponds to when uncertainty is revealed, this is when a flight departs. It should be noted that the choice of modelling dictates the solution technique. For instance, if one were to use a scenario tree approach, then a multi-stage stochastic programming solution approach would be used in solving the problem.

Demand side uncertainties are complicated to model given the underlying erratic cargo booking process. These uncertainties can be modelled by constructing demand disruptions. These take the existing set of cargo demand requests and add last minute requests, remove no shows and also update the weights & volume of the existing demand requests. One could also model demand side uncertainties by approximating probability distributions for the characteristics (e.g. weight, volume) of the demand requests.

What are the current, if any, deterministic or stochastic models which deal with planning operations of a cargo carrier?

Planning the operations of a cargo carrier is a four phase process. First one must construct a schedule. Thereafter, the fleet assignment is done. This involves assigning aircraft types to the flights. After that, particular aircraft are assigned to the flights. Finally, crew are assigned to man the flights.

The use of a time-space network is often done in these planning models. These networks have nodes which represent airports at a particular point in time, the arcs represent movement between these nodes. These time-space networks help construct sets which can be used in mixed integer linear programming models. The size of the problem is quite large, therefore heuristics are often used to aid the solution procedure. Column generation can also be used. In such models, a list of initial flight paths and aircraft rotations is established, after which new paths and rotations are added to the model by solving a pricing problem. The pricing problem is often a shortest path problem.

Some models have tried to deal with disruptions to the existing planning of airlines. More work has been done on passenger operations as opposed to cargo operations. Some of the models which focused on cargo operations assumed an initial set of demand requests which was disrupted with no-shows and new additions. The models aimed at rescheduling and rerouting aircraft and cargo in order to deal with the disruptions. These models used column generation and decomposition techniques in the solution process.

There are not many stochastic models which have been used in the operations of a cargo carrier. Such models typically use a multi-stage stochastic programming approach. Nested decomposition is used to solve these problems. However, such problems are not Markovian in nature and thus variations of the Nested Decomposition technique must be applied. Furthermore, these problems also typically have integer and binary variables, this requires further adapting the Nested Decomposition technique. Solving stochastic models is more time consuming as compared to a deterministic counterpart. This will adversely affect a model's ability to be integrated into the daily operations of a real combination airline.

What are the various ways the weight and balance of the aircraft can be ensured when carrying the cargo?

Weight and balance control is critical when loading and operating aircraft. The total weight loaded on the aircraft must not exceed the payload carrying capacity of the aircraft. Furthermore, the centre of gravity (cg) of the loaded aircraft should be within the required safety margins. Having a too aft cg location can cause the aircraft to become unstable. Most papers use a mixed integer programming approach to solve weight and balance control problems. A binary decision variable denotes if a particular unit load device (ULD) can be loaded at a particular position on board the aircraft. Constraints ensure that the weight of the loaded aircraft is less than the maximum allowed weight. Also, to reduce stresses in the aircraft structure, the weight of the various loaded aircraft compartments should not exceed a certain amount. Often, these models define a desired cg location and ensure that the loaded aircraft's cg location is within a certain tolerance of the desired cg location.

Since cargo carriers often have intermediate stopovers, some models also aim at reducing the cargo handling costs at these intermediate airports. For instance, loading ULDs such that only those which need to be removed at the intermediate airport are actually removed from the aircraft. To do this, the model must ensure that a ULD which does not need to be removed is not in the path of a ULD which needs to be removed at the intermediate airport. Accounting for these intermediate handling costs makes the model relatively more complex. This increases the computational time required to solve the model.

9.2. The Way Ahead

From the literature review, one can propose two different ways to model the solution tool. One is a predominantly reactive approach where the tool will react to demand disruptions. It will use chance constraints to model the belly-space capacity. The other is a predominately proactive approach where the tool will analyse the set of future possible scenarios and take a decision which minimizes the expected future cost of rescheduling and rerouting the cargo. Such a technique will likely use a multi-stage stochastic programming approach.

The predominantly proactive approach will likely give a better solution. However, such a model is difficult to solve and will likely have long runtimes. Given the timeline of this thesis project, it is better to model the tool using the predominantly reactive approach.

Bibliography

Federal Aviation Administration. *Pilot's handbook of aeronautical knowledge*. Federal Aviation Administration, 2016.

Jürgen Antes, Lenard Campen, Ulrich Derigs, Christoph Titze, and Gert Dieter Wolle. SYNOPSE: A model-based decision support system for the evaluation of flight schedules for cargo airlines. *Decision Support Systems, Elsevier*, 22(4):307–323, 1998.

Peter Belobaba, Amedeo Odoni, and Cynthia Barnhart. *The global airline industry*. John Wiley & Sons, 2009.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

Z. Berdowski, F.N. Van den Broek-Serlé, J.T. Jetten, Y. Kawabatta, J.T. Schoemaker, and R. Versteegh. Survey on standard weights of passengers and baggage. page 264, 2009.

John R Birge and Francois Louveaux. *Introduction to Stochastic Programming*. Springer Science & Business Media, 2nd edition, 2011.

Thijs Boonekamp, J Gromicho, W Dullaert, and B Radstaak. Air cargo revenue management. *Unpublished masters thesis, Vrije Universiteit, Amsterdam*, 2013.

Felix Brandt. *The air cargo load planning problem*. PhD thesis, KIT-Bibliothek, 2017.

Israel Brosh. Optimal cargo allocation on board a plane: a sequential linear programming approach. *European Journal of Operational Research*, 8(1):40–46, 1981.

Duarte Cachola. Performance and Efficiency Evaluation of Air Cargo Carriers Cargolux and Lufthansa Cargo Case Studies. 2017.

Yupo Chan and Ronny J. Ponder. The small package air freight industry in the United States: A review of the federal express experience. *Transportation Research Part A: General*, 13(4):221–229, 1979.

Chih Hsien Chen, Yao Hui Chang, and Shuo Yan Chou. Enhancing the design of air cargo transportation services via an integrated fuzzy approach. *Total Quality Management and Business Excellence*, 19 (6):661–680, 2008.

Larry Chestler. Overnight Air Express: Spatial Pattern, Competition and the Future in Small Package Delivery Service. *Transportation Quarterly*, 39(1), 1985.

Antonio J. Conejo, Enrique Castillo, Roberto Mínguez, and Raquel García-Bertrand. *Decomposition techniques in mathematical programming: Engineering and science applications*. Springer Science & Business Media, 2006.

Tom Crabtree, Tom Hoang, Gregg Gildemann, and Josh Collingwood. World Air Cargo Forecast 2020–2039. Technical report, The Boeing Company, 2020.

George B. Dantzig and Philip Wolfe. Decomposition Principle for Linear Programs. *Operations Research*, 8(1):101–111, 1960.

Welington de Oliveira, Juan Pablo Luna, and Claudia Sagastizábal. BAS: Basic Course on Stochastic Programming, 2016.

Felipe Delgado and Julio Mora. A matheuristic approach to the air-cargo recovery problem under demand disruption. *Journal of Air Transport Management*, 90(October 2020):101939, 2021.

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(April):292–307, 2019.

Felipe Delgado, Cristóbal Sirhan, Mathias Katscher, and Homero Larrain. Recovering from demand disruptions on an air cargo network. *Journal of Air Transport Management*, 85(March), 2020.

Ulrich Derigs and Stefan Friederichs. Air cargo scheduling: Integrated models and solution procedures. *OR Spectrum*, 35(2):325–362, 2013.

Ulrich Derigs, Stefan Friederichs, and Simon Schäfer. A new approach for air cargo network planning. *Transportation Science*, 43(3):370–380, 2009.

Bo Feng, Yanzhi Li, and Zuo Jun Max Shen. Air cargo operations: Literature review and comparison with practices. *Transportation Research Part C: Emerging Technologies*, 56:263–280, 2015.

Peter Gerber. How to become an air cargo expert? Peter Gerber of Lufthansa Cargo teaches you in simple steps. https://www.youtube.com/watch?v, 2017.

P. C. Gilmore and R. E. Gomory. A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, 9(6):849–859, 1961.

Mattias Grönkvist. *The Tail Assignment problem*. 2005.

Dong Ling Han, Loon Ching Tang, and Huei Chuen Huang. A Markov model for single-leg air cargo revenue management under a bid-price policy. *European Journal of Operational Research*, 200(3): 800–811, 2010.

Stefan Irnich and Guy Desaulniers. *Shortest path problems with resource constraints*. Springer Science & Business Media, 2005.

Michael Jünger, Thomas M Liebling, Denis Naddef, George L Nemhauser, William R Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A Wolsey. 50 years of integer programming 1958-2008: From the early years to the state-of-the-art. Springer Science & Business Media, 2009.

Robert S. Kaplan and David P. Norton. The balanced scorecard: Measures That drive performance. *Harvard Business Review*, 83(7-8), 2005.

Raja G Kasilingam. Air cargo revenue management: Characteristics and complexities. *European Journal of Operational Research*, 96(1):36–44, 1997.

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

Michael J. Kuby and Robert Gordon Gray. The hub network design problem with stopovers and feeders: The case of Federal Express. *Transportation Research Part A*, 27(1):1–12, 1993.

Anne Lange. Does cargo matter? The impact of air cargo operations on departure on-time performance for combination carriers. *Transportation Research Part A: Policy and Practice*, 119(December 2017): 214–223, 2019.

Ole Larsen and Gert Mikkelsen. An interactive system for the loading of cargo aircraft. *European Journal of Operational Research*, 4:367–373, 1980.

Sabine Limbourg, Michaël Schyns, and Gilbert Laporte. Automatic aircraft cargo load planning. *Journal of the Operational Research Society*, 63(9):1271–1283, 2012.

Yitao Liu, Zhenbo Yang, Junxiang Deng, and Junjie Zhu. Investigation of fuel savings for an aircraft due to optimization of the center of gravity. In *IOP Conference Series: Materials Science and Engineering*, number 7, page 072018. IOP Publishing, 2018.

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

Roy E Marsten and Michael R Muller. A mixed-integer programming approach to air cargo fleet planning. *Management Science*, 26(11):1096–1107, 1980.

Louis A Martin-Vega. Aircraft load planning and the computer description and review. *Computers & industrial engineering*, 9(4):357–369, 1985.

Mansoor Mirza. Economic impact of airplane turntimes. *Aero Quarterly*, 4:15–19, 2008.

Marcel Mongeau and Christian Bes. Optimization of aircraft container loading. *IEEE Transactions on aerospace and electronic systems*, 39(1):140–150, 2003.

David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016.

James Murphy. Benders, Nested Benders and Stochastic Programming: An Intuitive Introduction. http://arxiv.org/abs/1312.3158, 2013.

Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1):359–375, 1991.

Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

Vasco Reis and Joao Silva. Assessing the air cargo business models of combination airlines. *Journal of Air Transport Management*, 57:250–259, 2016.

Sabre Airline Solutions. Load Planning - A discussion of fuel cost containment relevant to load planning. https://www.sabreairlinesolutions.com/images/uploads/Load_Planning_WP_0408_finalLo.pdf, 2007.

Tohru Tamura and Sigeru Inano. Scheduling and routing models for regional airline systems. *INFRASTRUCTURE PLANNING REVIEW*, 5:155–162, 1987.

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.

Wim Vancroonenburg, Jannes Verstichel, Karel Tavernier, and Greet Vanden Berghe. Automatic air cargo selection and weight balancing: a mixed integer programming approach. *Transportation Research Part E: Logistics and Transportation Review*, 65:70–83, 2014.

John Versleijen. Lufthansa Cargo sluit 2016 af met rode cijfers. https://www.nt.nl/luchtvracht/2017/03/16/lufthansa-cargo-sluit-2016-af-met-rode-cijfers/, 2017.

VRR. Identifying a Unit Load Device (ULD). https://vrr.aero/knowledge-center/uld-info/uld-id-code/.

WestJet. Wide Body Fleet / ULD Types. https://www.westjet.com/en-ca/cargo/shipping/fleet-uld.

Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 1 edition, 1988.

DeWulf Wouter. The Strategy of Air Cargo airlines, 2017.

Shangyao Yan and Chia-Hung Chen. Optimal flight scheduling models for cargo airlines under alliances. *Journal of Scheduling*, 11(3):175–186, 2008.

Shangyao Yan, Yu-hsuan Chen, Ching-hui Tang, and Yuping Tu. A model for airline passenger and cargo flight scheduling.

Shangyao Yan, Shin-Chin Chen, and Chia-Hung Chen. Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E: Logistics and Transportation Review*, 42(5):409–430, 2006.

Markus Zils. *Modellbasierte und prozessorientierte Flugplanung*. PhD thesis, PhD thesis, Department of Information Systems and Operations Research, University of Cologne, 2000.

Jikai Zou, Shabbir Ahmed, Xu Andy Sun, and HM Stewart. Nested decomposition of multistage stochastic integer programs with binary state variables. *Optimization Online*, 5436, 2016.