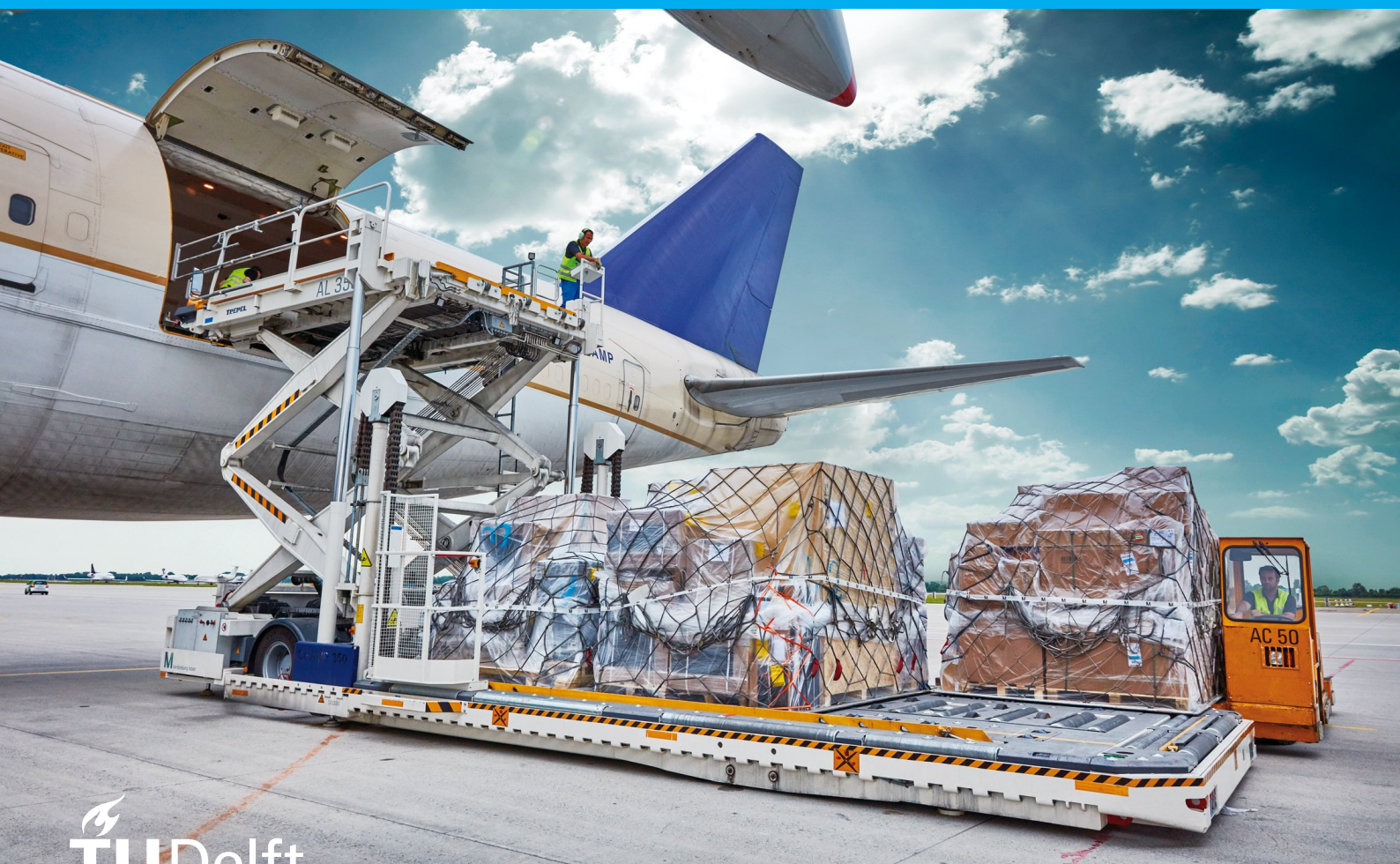


The Clustered Pickup and Delivery Problem with Time Windows and Multi Stack Side-Accessible Last-in First-out Loading

CPDPTW-SAL

Niels Maseland



The Clustered Pickup and Delivery Problem with Time Windows and Multi Stack Side-Accessible Last-in First-out Loading

CPDPTW-SAL

by

Niels Maseland

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on July 8th 2021.

Student number: 4540832
Project duration: October 2020 – July 2021
Thesis committee: Prof.dr.ir. J.M. Hoekstra
Dr. A. Bombelli
ir. P.C. Roling

Cover page is taken from: [1]

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

Acknowledgements

This document is the final result of nine months full time working on this thesis project. Apart from the challenging thesis topic, the conditions under which the research was performed were maybe even more challenging at some moments. The COVID-19 pandemic resulted in strict measures for physical on campus education for universities. This resulted in doing the thesis fully remotely and speaking to my supervisor(s) online only. I am very thankful for the help that I got and would like to spend a few words on this.

First of all, I would sincerely like to thank my daily supervisor, Alessandro Bombelli. Over the past nine months you have been encouraging me to keep improving the work with good ideas and suggestions. I remember that during our first meeting in the new calendar year, we expressed our hope that we would be able to physically meet once. Explaining the full set of linear equations and methods via Teams is not ideal, but you were always able to reflect on my work. Unfortunately, we were not able to physically meet for our weekly update and discussion, but I am looking forward to have the final thesis defence at the faculty. Hopefully my work can contribute to your future research in this field. Furthermore I would like to express my gratitude to ir. P.C. Roling for attending all official meetings and providing valuable feedback. Also special thanks to Prof. dr. ir. J.M. Hoekstra for chairing the thesis committee.

Apart from the TU Delft staff, I would like to thank my family and friends for supporting me. Working on a thesis subject for nine months remotely wasn't always easy. Discussing the work with family and friends was a welcome opportunity to motivate myself and keep going.

Niels Maseland
Gouda, July 2021

Contents

List of Abbreviations	vii
Introduction	ix
I Scientific Paper	1
II Literature Study	35
1 Introduction	37
2 Research Outline	39
2.1 Research Problem	39
2.2 Research Objective	39
2.3 Research Question(s)	40
2.4 Problem Example	41
2.5 Project Planning	41
3 Air Cargo Supply Chain	43
3.1 Current Procedures in the Air Cargo Supply Chain at Schiphol Airport	43
3.2 Relevant Efficiency Improvement Concepts	44
4 Key Performance Indicators	47
4.1 Travel Distance	47
4.2 Travel Time	47
4.3 Fuel Consumption	48
4.4 Financial Cost	48
4.5 Time Window Preference	48
4.6 Benchmark Datasets	48
5 Routing and Dock Allocation Model: an Exact Formulation	51
5.1 Vehicle Routing Problem Variants	51
5.2 Linear Vehicle Routing Problem Formulations	52
5.2.1 Two-index	52
5.2.2 Three-index	52
5.2.3 Set Partitioning	53
5.2.4 Commodity Flow	54
5.3 Linear Dock Allocation Model	55
6 Routing and Dock Allocation Model: an Heuristic Formulation	59
6.1 Heuristics of the Vehicle Routing Problem	59
6.1.1 Initial Solution	59
6.1.2 (Adaptive) Large Neighborhood Search	59
6.1.3 Simulated Annealing	61
6.1.4 Tabu Search	62
6.1.5 Genetic Algorithm	62
6.1.6 Dynamic Programming	63
6.2 Heuristics of the Dock Allocation Model	63
7 Truck Docking Priority Factors	65
7.1 Waiting Time	65
7.2 Commodity Types	65
7.3 Truck Load Factor	66
7.4 Multi-Objective Optimization	66

8	Model Extensions	69
8.1	Two-Dimensional Bin Packing Problem.	69
8.2	Stochastic Modelling	72
8.3	Site Dependent Model	73
8.4	Time Dependent Model.	73
9	Conclusion	75
III	Supporting work	79
1	Appendix 1	81
2	Appendix 2	83
	Bibliography	85

List of Abbreviations

2BPP	Two-Dimensional Bin Packing Problem
2L-CVRP	Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints
2SPP	Two-Dimensional Strip Packing Problem
3L-CVRP	Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints
ALNS	Adaptive Large Neighborhood search
CPDPTW-SAL	Clustered Pickup and Delivery Problem with Time Windows with Side-Accessible LIFO Loading
DP	Dynamic Programming
FF	Freight Forwarder
GA	Genetic Algorithm
GH	Ground Handler
KPI	Key Performance Indicator
LIFO	Last-in First-out
LNS	Large Neighborhood search
MILP	Mixed Integer Linear Programming
MTP	Maximum Touching Perimeter Method
NL	No-LIFO
PDP	Pickup and Delivery Problem
PDPTW	Pickup and Delivery Problem with Time Windows
PI	Physical Internet
SA	Side-Accessible LIFO
SHF	Shelf Heuristic Filling
SL	Strict-LIFO
TS	Tabu Search
TW	Time Window
ULD	Unit Load Device
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows

Introduction

The globalisation has a lot of positive effects on our daily life. The ease of accessibility to the Internet allows us to buy products from all over the world. The shipping times are often fairly limited if one considers the distance that has to be covered by a product. The existence and expansion of a global airline network significantly contributes to this fast delivery process. Although the (passenger) airline industry was severely impacted by the COVID-19 pandemic, it is not unreasonable to assume that the industry will restore to pre-pandemic volumes in a few years. For most industries, expansion to a certain extent is possible without loss of efficiency. Industries could simply scale up their buildings, assets and personnel. This could even lead to an increase in efficiency due to larger scale production. This principle also applies to the aviation industry to a certain extent. For example: the capacity of trucks and airplanes is increased and the efficiency is increased because a plane carries more cargo and/or passengers. Simply keep scaling up does however also lead to challenges that require another approach. This research will describe one of these challenges and provide a method to increase the efficiency.

In the current air cargo supply chain process, trucks pick up shipments from freight forwarders and these are shipped to ground handlers. The freight forwarding companies tend to pick up the shipments during a workday and deliver the shipments to the ground handlers at the end of a working day. The arrival of trucks at the ground handlers is uncoordinated which leads to peak hours. During these peak hours, the number of trucks arriving at the ground handlers is larger than the number of available docks. This leads to congestion and delay outside the ground handlers. For all involved parties, this leads to financial losses which is undesirable.

In the work that lies in front of you, this challenge is addressed and an optimization model is developed. The aim of this optimization method is to model the air cargo export supply chain. The objective function of this optimization function is to minimize the total cost of a solution. In addition to routing and docking, the model also introduces a novel Last-In First-Out model variant. In this variant, a multi-stack truck is introduced where shipments can also be Side-Accessible in the unloading route.

The thesis work is done to complete the Air Transport and Operations master track at the Faculty of Aerospace Engineering at the TU Delft. The work is performed under supervision of Dr. A. Bombelli. The thesis subject is in line with his work and expertise on air cargo operations, vehicle routing problems and methods to solve these. In his research, he has been working on this topic and asked me to think of an approach to model the cargo export supply chain. In addition to that, Dr. Bombelli was interested to see how a new loading formulation can be developed that allows for the Side-Accessible feature of this model.

The unique aspect of this thesis is that the provided Side-Accessible model variant can be introduced to a broad spectrum of applications. In this thesis it is applied to the context of the aviation export supply chain. However, the provided formulation can be implemented to many more practical applications. One could think for example to the supply of stock to supermarkets and all other applications of truck companies that make use of some sort of Unit Load Device.

The structure of this thesis report is as follows. First of all, [Part I](#) presents the scientific paper. In [Part II](#) the literature study is presented that was conducted at the start of the thesis period. In [Part III](#) the supporting work of this thesis is presented. This consists of the following parts. First of all, in [Appendix 1](#) a simple verification case for the heuristic model is presented and worked out. [Appendix 2](#) presents an overview of the data instance generation process.

I

Scientific Paper

The Clustered Pickup and Delivery Problem with Time Windows and Multi Stack Side-Accessible Last-in First-out Loading

Niels Maseland*

Delft University of Technology, Delft, The Netherlands

Abstract

This paper studies a variation of the pickup and delivery formulation with time windows which is applied to air cargo export operations. The formulation is extended using three factors. 1) Pickup nodes are positioned at freight forwarders. Delivery nodes are located at ground handlers. Trucks can only visit one freight forwarder. 2) Dock capacity of the ground handlers is implemented in the model. Each dock can only be occupied by one truck at a time. 3) A multi stack loading approach is introduced for the trucks. To be consistent with practice, a Last-in First-out (LIFO) approach is considered when delivering shipments. Three model variants are introduced in this paper with respect to the LIFO strategy with different nuances. i) All LIFO constraints are relaxed. This is also referred to as the *no-LIFO (NL)* model variant. ii) Only direct accessibility in the stacks is allowed. This is also referred to as the *strict-LIFO (SL)* model variant. iii) Side-accessible unloading from an adjacent stack is also allowed. This is also referred to as the *side-accessible (SA)* model variant. For each model variant, an exact model is presented and solved with the branch-and-bound approach. In addition to that, a meta-heuristic is developed that is based on a large neighborhood search to solve large data instances. Two objective functions are used. First of all, a cost-based objective function where a fixed penalty per truck is introduced. The second objective function is time-based, where only the total time duration of the routes of all trucks is minimized. It is concluded that the meta-heuristic model gives good results in terms of solution quality, computational time and stableness. It is also concluded that the SA model variant benefits over the SL model variant. The relative benefit depends on the data instance, capacity of a stack and the used objective function. For small data instances, a maximum benefit of 15.0% was observed for the SA model variant using the exact and meta-heuristic model. For larger data instances, a maximum benefit of 14.1% was observed using the meta-heuristic model. Given the current trends towards more shipment standardization in logistics and more collaboration among stakeholders, we believe many supply chains can potentially benefit from this scheduling model that incorporates both paradigms.

1 Introduction

The availability of a large (cargo) aviation network is an attractive opportunity to increase the amount of cargo that is shipped by aircraft as aviation provides a fast method for shipping freight over long distances. Flight schedules are often known in advance, which increases the predictability of arrival and departure times of flights. The predictability is highly desired for companies responsible for further transporting and processing of shipments. Boeing estimated in their *World Air Cargo Forecast 2018-2037* that the transported cargo volume by air will increase by 4.2% each year [Boeing, 2018]. Although this estimation was made before the coronavirus affected the aviation industry, it is not unlikely that this trend will keep on going after the pandemic. The increase in transported volume has its consequences throughout the entire supply chain and bottlenecks should be identified in order to allow for this increase in transported volume. One of the bottlenecks in the air cargo supply chain is identified in this study and an optimization model is proposed.

In the current air cargo supply chain, freight forwarders and ground handlers play an important role. Freight forwarders receive packages from various companies and are responsible to transport them by truck to the ground handlers at the airport. The ground handlers are on their turn responsible for processing the incoming shipments from the freight forwarders and ensure that the shipments are transported to their scheduled flight. The arrival of trucks at the ground handlers is currently an uncoordinated process because freight forwarders tend to optimize their own truck schedules. The ground handlers have a limited dock capacity and the uncoordinated arrival of trucks leads to the situation where the number of incoming trucks is larger than the number of available docks as concluded in [Verduijn et al., 2019]. This leads to delay and congestion outside the ground handlers that lead to financial losses for all involved parties.

Apart from the challenges in the aviation industry, the general logistic sector also faces inefficiencies. The environmental, economical and societal unsustainability of the current logistic sector have been thoroughly

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

review by [Montreuil, 2011]. For example: trucks are only partially loaded, trucks drive back empty, and shipments are potentially transported all over the world before reaching their destination. The Physical Internet (PI) is an innovative concept where shipments are not bounded to a specific shipper, but can make use of a worldwide network. One of the important success-factors in this concept is the design of standardized modular boxes [Montreuil, 2011]. The call for generalized box modularization is researched by [Landschützer et al., 2015]. In this work the requirements and design of containers for the physical internet concept are presented. The outcome is an Unit Load Device (ULD) where multiple smaller individual packages are combined. All ULDs should have the same outer dimensions which allows for efficient handling and design of other systems in the sector. In this work, the shipments are represented by ULDs to prepare the implementation of this work in the concept of the physical internet.

The purpose of this paper is to present an optimization method for the air cargo supply chain that includes the limited dock capacity for ground handlers and different LIFO loading variants. This paper presents an exact and meta-heuristic method to model the problem at hand. This work also aims at showing the benefit of the side-accessible feature in the provided context of the air cargo supply chain. The model includes three important aspects. 1) The model performs routing from the freight forwarders to the ground handlers. This can be incorporated in the model as a variant of the *Pickup and Delivery Formulation with Time Windows*. The formulation should be changed so that pickup nodes represent the freight forwarders. The delivery nodes represent the ground handlers. Due to practical objections that were found against horizontal collaboration by [Basso et al., 2019], it was decided that horizontal collaboration is not allowed. This means that each truck can only visit a single freight forwarder. This work is limited to export operations, meaning that import is not considered. 2) All trucks arriving at a ground handler are assigned to a dock. It is never allowed to have more than one truck occupy a dock at the same time. Similar problems in literature are the parking spot assignment or gate allocation problem. 3) The pickup and delivery route should be visited in such order that no unloading of other items is required if visiting a delivery node. This is referred to as the Last-In-First-Out (LIFO) constraint. In this study, a multi stack system is introduced in trucks as is done before by [Côté et al., 2012] and [Iori and Riera-Ledesma, 2015]. The innovative feature of this model is that side-accessibility between adjacent stacks is allowed. This new feature increases the number of feasible pickup and delivery route combinations. In this work, three model variants are considered. 1) *no-LIFO* model variant. This model variant does not have LIFO loading constraints. Only the capacity of a truck is taken as a constraint. 2) *strict-LIFO* model variant. In this model variant, only the ULD positioned closest to the (un)loading door in each stack can be accessed. This prevents certain ULDs to be temporarily unloaded, so that another ULD can be accessed. 3) *side-accessible* model variant. In this model variant, an ULD can also be accessed if an adjacent stack has a free spot.

In Figure 1 an example is presented of a possible truck loading pattern. This truck has a layout of two stacks, where each stack has a capacity of three. Each location in the stack is referred to as (c,r) where c indicates the stack number and r represents the location in the stack. In the current situation, the truck is loaded with four ULDs, represented as Pickup Node 1, 2, 3 and 4. The locations that are not occupied by any ULD are represented as *Free*. In this example, these are locations (2,2) and (2,3). If the no-LIFO model variant is used, the unloading sequence is not relevant, so all nodes can be unloaded in this layout. For the strict-LIFO model, only the last ULD in a stack can be accessed. This means that only Node 3 and 4 can be delivered if the truck is loaded as represented in the figure. If the side-accessible model variant is used, a location can also be accessed via an adjacent stack with a free spot. This implies that Node 2 can also be accessed in this situation. This implies that Node 2, 3 and 4 can be accessed if the side-accessible model variant is used. In terms of solution quality representation, it can thus be concluded that the no-LIFO model always find the best solution, followed by the side-accessible model variant. The strict-LIFO model variant should perform the worst. It is also possible that one (or more) solution find the same solution quality. Visiting multiple freight forwarders is not allowed, leading to a clustering of pickup nodes at the freight forwarders. The model is referred to as *The Clustered PDPTW Formulation with Multi Stack Side-Accessible LIFO Loading (CPDPTW-SAL)*.

The build up of this paper will be as follows. In section 2 relevant literature is presented. In section 3 the MILP formulation is described. In section 4 the meta-heuristic method is explained which is based on an a large neighborhood search. In section 5 the results of the CPDPTW-SAL are presented.

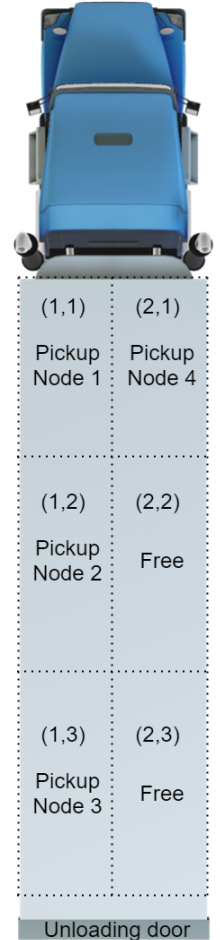


Figure 1: Example truck loading representation

2 Literature Review

Increasing the efficiency of individual parts of the air cargo supply chain process can be implemented at many different levels and locations in the process. Collaboration between the involved parties in the air cargo supply chain can significantly increase the efficiency of the process. For this work, the two most relevant parties involved are the freight forwarders and ground handlers. Collaboration between these parties can be *horizontal* or *vertical*. Horizontal collaboration is explained in [Prakash and Deshmukh, 2010] and implies that companies collaborate that are operating at the same level in the supply chain. In contrast to that, vertical collaboration is between parties that operate at different levels in the supply chain. Horizontal collaboration in the air cargo supply chain could decrease the transportation cost up to 40%, as concluded by [Ankersmit et al., 2014]. Horizontal collaboration in this specific context would imply that a neutral fleet of trucks can pickup shipments from different freight forwarders and transport these to the ground handlers. This method thus seems very promising but is however not yet widely implemented. As found by [Basso et al., 2019], there are several practical issues that prevent the implementation of horizontal collaboration. The four categories which are identified are: 1) Design, 2) Planning and Operations, 3) Business/market, 4) Behaviors. Each of these four categories is split up in more detail and it is concluded that there is a total of 16 important practical issues against horizontal collaboration. Information sharing between companies is also something that competitors are not very open to. This is however required to efficiently implement horizontal collaboration. Although there is a potential benefit when including full horizontal collaboration, it is decided that this is not done in this work due to the practical difficulties.

Another efficiency improving concept is the method of *cross-docking* which is discussed in [Boysen et al., 2013]. In cross-docking methods, all trucks travel to a central cross-docking station. At this station, trucks are unloaded and the shipments from different trucks are combined in outgoing trucks. This procedure is done to increase the load factor of the departing trucks and thus reducing the number of outgoing trucks, compared to the number of incoming trucks.

The *milk run* principle as described in [Brar and Saini, 2011] can reduce the number of trucks used and decrease the total distance which is travelled by all trucks. In the milk run principle, a fleet of neutral trucks visits the pickup locations where the pickup quantity is known in advance which allows to optimally load the trucks. The milk run principle has been introduced to Schiphol Airport with various ground handlers and trucking companies since the 1st of May 2015. One year after the introduction of the project, a reduction of 40% of the truck movements was observed. In addition to that, the emission of carbon dioxide was decreased up to 30% [Air Cargo Netherlands, 2016].

The research projects that are mentioned above aim at a specific part in the entire supply chain and their main goal is to reduce the total number of trucks and to increase the efficiency of the trucks which are used. Another method to increase the efficiency of the air cargo supply chain is the implementation of a full mathematical optimization model. The most intuitive method would be the implementation of a vehicle routing problem and add any required extensions to this type of problem. The first Vehicle Routing Problem (VRP) was introduced in 1959 as the *Truck Dispatching Problem* [Dantzig and Ramser, 1959]. A fleet of trucks needs to meet the demand of a set of customers. The aim of the optimization model is to minimize the total distance travelled by all trucks. The *Capacitated Vehicle Routing Problem* is an extension of this model where a truck has a physical capacity which cannot be exceeded. The model of [Laporte, 1992] incorporates this feature by constraining the load at each node to be smaller or equal than the maximum load of the vehicle. Time Windows can be added to nodes as is done in the research of [El-Sherbeny, 2010]. The VRP can be extended to a situation in which there are multiple depots, as is done in [Wang et al., 2015]. The disadvantage of this method is that each depot is assumed to have the same commodity type. The Pickup-and-Delivery (PDP) formulation is a more specific extension which assigns a pickup and delivery node for each shipment as presented in [Ropke and Pisinger, 2006] and [Rais et al., 2014]. The Pickup-and-Delivery formulation can also be implemented with time windows, leading to the PDPTW. The implementation of the export air cargo supply chain seems to have a lot in common with the PDPTW.

If a (meta)-heuristic method is used, the accepting criteria of a solution is relevant for the final outcome of the total algorithm. If one would only accept a new solution if it is better than the best known solution up to that point, one might end up with a local optimum instead of the global optimum. In the work of [Ropke and Pisinger, 2006] and [Laporte et al., 2014] the introduction of the Simulated-Annealing accepting criteria is introduced for heuristic method. With this method, also solution that are worse than the best known solution can be accepted to explore the full solution space.

Apart from routing the trucks, literature for the docking feature of the model is also researched. Assignment of trucks or airplanes to docks and gates respectively has been the topic of studies for a long time. In the research of [Roca-Riu et al., 2015] the topic has been addressed for parking spot assignment for truck (un)loading operations in a city center. The research is based on a VRPTW where each route represents one parking spot. The number of routes in this VRPTW cannot exceed the number of available parking spots. This approach has also been taken by [Boysen et al., 2013] where a cross-docking station is modelled with limited number of

docks available for incoming trucks. The work of [Rieck and Zimmermann, 2010] presents a similar method that is applied to a vehicle routing problem where the depots have a limited dock capacity. Another dock scheduling approach is introduced by [Miao et al., 2009] which is not based on a VRPTW model. In their formulation a new set of decision variables, n_{ij} , is introduced which is binary and equal to 1 if the departure time of truck i is smaller or equal than the arrival time of truck j and zero otherwise. Other sets of constraints are introduced which state that if truck i and j are allocated to the same dock, then this implies that $n_{ij} + n_{ji}$ needs to be larger or equal to one. If this constraint is satisfied, this means that the trucks do not overlap and thus can be assigned to the same dock. The research of [Mangoubi and Mathaisel, 1985] already addressed the gate assignment problem where only one plane can be assigned to a dock at a time. In the work of [Xu and Bailey, 2001] a binary decision variable is introduced, z_{ijk} , which is equal to one if and only if flight i and j are assigned to gate k and flight i should immediately precede flight j .

Incorporating loading of trucks in the vehicle routing problem implies that the physical dimensions of the shipment fit within the truck and that shipment dimensions do not overlap with each other. In [Iori and Martello, 2010] an overview of vehicle routing problems with loading constraints is presented. For two dimensional loading, a distinction is made between the *Two-Dimensional Bin Packing Problem (2BPP)* and *Two-Dimensional Strip Packing Problem (2SPP)*. In the 2BPP all shipments are placed in different bins and the number of bins is minimized. In contrast, in the 2SPP all items are placed in the same bin. In that case, the total length of the bin is minimized. For this research a 2BPP or a variant seems to be most promising. The literature on exact methods which implement two dimensional loading constraints in the vehicle routing problem is fairly limited as is concluded by [Pollaris et al., 2014]. Only the work of [Iori et al., 2007] and [Martínez and Amaya, 2013] are mentioned. The first work describes the exact location of items by x and y coordinates in the truck and these are implemented as decision variables. The second work does the same, but considers circular items which complicates the work. The literature overview of [Pollaris et al., 2014] only refers to the work of [Junqueira et al., 2013] where three dimensional loading constraints are included. This research differs from the two dimensional models in terms of modelling technique as binary decision variables are used to determine the location of a shipment. This implies that for all possible x , y , z combinations, a decision variable should be created. Instead of assigning each item to x , y (and z for three-dimensional loading) coordinates, the introducing of a (*multi*) *stack* system can simplify the constraints. Reality also requires that the loading and unloading sequence of the trucks is taken into account. When unloading, it is desired to have the shipment which should be unloaded immediately accessible without temporarily unloading other shipments. This constraint is referred to as the Last-In-First-Out (LIFO) constraint. In the work of [Cordeau et al., 2010] a PDP problem is addressed with LIFO-loading constraints. This work can also be seen as a single-stack system where the LIFO constraint is satisfied. The working principle behind the LIFO constraint is that the load of a vehicle when visiting the delivery node should be exactly equal to the load at the pickup node after picking up the shipment. If a system is introduced with multiple stacks, the number of possible pickup and delivery routes is significantly increased. When the LIFO constraint is introduced to a multi-stack system, the number of feasible routes is thus also significantly higher compared to a single-stack system. In the work of [Côté et al., 2012] a multi-stack system is introduced where the LIFO constraint is implemented. The working principle in this work is the same as described in the work of [Cordeau et al., 2010].

The research of [Ou et al., 2010] is specific for the aviation industry and schedules the arrival of trucks at an air cargo terminal. This goal of this work is to minimize the total cost of the overall solution. The model also incorporates a limited number of docks for (un)loading activities. This model does however not include routing from freight forwarders to ground handlers and only one ground handler is modelled. In literature there is a general lack of models that include the routing between freight forwarders and ground handlers. Also the combination between routing, docking and loading was not found. Especially with the addition of the side-accessible loading feature, it is believed that this research can contribute to the available literature in this field.

3 Exact Methodology Approach

This section will present the exact model methodology of the proposed model. First of all, in section 3.1 the general model formulation will be discussed. The process of generating arcs is described in section 3.2. In section 3.3 the decision variables of the model are presented and explained. The full MILP model is presented in section 3.4. Finally, section 3.5 presents some additional information about the stack-routing method that is used in the MILP formulation.

3.1 General Model Explanation

The linear model is closely related to the Pickup and Delivery Formulation with Time Windows. The set of pickup nodes, \mathbf{P} , represents the shipments to be transported. Each shipment is represented as an unit load device (ULD). Each pickup node is located at a specific freight forwarder. The set of freight forwarders is represented by the set \mathbf{FF} . The total number of pickup nodes is represented as n . This implies that $j = i + n$ where i is the pickup node and j the delivery node. The delivery nodes are located at the ground handlers and represented in the set \mathbf{D} . The set of ground handlers is represented by \mathbf{GH} . The set \mathbf{N} contains the pickup and delivery nodes. The set \mathbf{V} contains the pickup, delivery, start depot and end depot nodes. Each truck from the set \mathbf{K} has to depart from the *start depot node* (s) and arrive at *end depot node* (e). If a route is not used, this simply implies that the route is from the start depot node to the end depot node. The set of arcs, \mathbf{A} , is selected with underlying principles which are explained in section 3.2. All nodes have a time window which is represented by $[A_i, B_i]$ where A_i is the earliest start service time and B_i the latest start service time of a node. The parameter C_i represent the processing time of node i . The load of a node is introduced by the parameter D_i . All nodes are represented by an ULD. The load of a pickup node is thus equal to 1 and the load of a delivery node is equal to -1 . The set of docks is represented by \mathbf{M} . The multi-stack loading feature of the model introduces for each truck the set of stacks, \mathbf{C} . Each stack has a capacity Q and each stack has a set of locations where a pickup node can be positioned and is represented by the set \mathbf{R} . For cases where more than two stacks are used, some stacks are not adjacent. The binary variable $Z_{c,c'}$ is set equal to zero if c and c' are not adjacent stacks. Finally, the fixed buffer time T_b is introduced if a truck leaves a ground handler. It is assumed that the truck fleet is homogeneous, so the truck specific parameters are the same for all trucks. An overview of the parameters and sets is presented in Table 1 and Table 2 respectively.

Table 1: Parameters used in the linear model

Parameters	
s	Start depot node
e	End depot node
n	Total number of pickup nodes
T_{ij}	Travel time from node i to node j
A_i	Start of time window of node i
B_i	End of time window of node i
C_i	Processing time of node i
D_i	Load of a node: 1 if i is pickup node, -1 if i is delivery node
$Z_{c,c'}$	Binary variable, equals 1 if c and c' are adjacent stacks, zero otherwise
Q	Capacity of a stack
T_b	Fixed buffer time at a dock

Table 2: Sets used in the linear model

Sets	
P	Set of pickup nodes
D	Set of delivery nodes
N	$\mathbf{P} \cup \mathbf{D}$
V	$\mathbf{N} \cup s \cup e$
K	Set of vehicles
A	Set of arcs
GH	Set of ground handlers
FF	Set of freight forwarders
M	Set of docks
C	Set of stacks in a truck
R	Set of locations in a stack

3.2 Arc Generation Process

As mentioned before, the pickup nodes are located at freight forwarders and delivery nodes at the ground handlers. The formulated model does not allow for horizontal collaboration between freight forwarders. There are two methods to prevent a truck from visiting multiple freight forwarders. First of all, the set of arcs could be generated between all nodes in the set **V** and additional constraints are introduced which do not allow a truck to visit multiple freight forwarders. This method has the disadvantage that the number of arcs is high and additional constraints should be introduced to prevent trucks from visiting multiple freight forwarders.

Another approach is to create the set of arcs accordingly to the nature of the problem. Arcs between freight forwarders are not included in the set of arcs and thus no additional constraints are needed. In total, seven type of arcs should be generated. 1) Arcs from pickup nodes to other pickup nodes within the same freight forwarder. 2) Arcs from delivery nodes to other delivery nodes within the same ground handler. 3) Arcs from delivery nodes to other delivery nodes within another ground handler. 4) Arcs between pickup nodes in the freight forwarder to delivery nodes in the ground handlers. 5) Arcs from the start depot node to the pickup nodes. 6) Arcs from the delivery nodes to the end depot node. 7) Arcs from the start depot node to the end depot node. This approach is used in the MILP formulation presented later. Therefore no additional constraints are introduced that prevent a truck from visiting multiple freight forwarders.

In addition to that, non feasible arcs with respect to time windows are excluded from the set of arcs. If $A_i + C_i + T_{ij} > B_j$, the combination between node i and j is not feasible and the arc is not included in the set **A**.

3.3 Decision Variable Overview

In Table 3 the decision variables of the proposed model are presented. The binary decision variable is x_{ij}^k represents if vehicle k travels from node i to j. At time t_i , the service of node i will start. The time at which servicing node i is finished, if thus equal to: $t_i + C_i$. If a truck arrives at the ground handler before a dock is available or before the start of a time window, the truck has to wait at the ground handler before service can start. This waiting time of truck k at ground handler gh is represented by w_{gh}^k . The arrival time and departure time of truck k at ground handler gh is represented as a_{gh}^k and d_{gh}^k respectively. The start of docking at a ground handler is thus equal to $a_{gh}^k + w_{gh}^k$. If the docking time of truck k' is larger or equal than the departure time plus the fixed buffer time of truck k from the same ground handler gh, this implies that these trucks can be served at the same dock consecutively. This is represented by the binary variable $h_{gh}^{k,k'}$. If a truck has to wait at a freight forwarder or ground handler between servicing two nodes, inter-node time is introduced and represented as v_i . Truck assignment to docks is done using the variable $q_{k,m}$ which implies that truck k is assigned to dock m. The variable $z_{m,m'}^{k,k'}$ is equal to 1 if truck k is assigned to dock m and truck k' to dock m'. The decision variable $y_{i,c}^k$ is equal to 1 if pickup node i is assigned to stack c in truck k. The parameter $\hat{l}_{i,c}^k$ and $l_{i,c}^k$ represent the load of stack c in truck k before and after servicing node i respectively. The binary variable $p_{i,j,c}^k$ equals 1 if i and j are both assigned to stack c in vehicle k and i directly precedes j in the route. The binary $\alpha_{i,c,r}^k$ variable defines if pickup node i is assigned to location (c,r) in truck k. The variable $\beta_{c,r,n}^k$ describes if location (c,r) in truck k is directly accessible from the (un)loading door at delivery node n. The binary variable $\gamma_{c,r,n}^k$ represents if location (c,r) in truck k is free (not used) when visiting delivery node n. Finally, $\delta_{c',c,r,n}^k$ presents if location (c,r) is side-accessible via stack c' in truck k.

Table 3: Decision variables used in the linear model

Variables	Type	Explanation
x_{ij}^k	Binary	Equals 1 if vehicle k travels from node i to node j
t_i	Integer	Start time of servicing node i
w_{gh}^k	Integer	Waiting time of truck k at ground handler gh
a_{gh}^k	Integer	Arrival time of truck k at ground handler gh
d_{gh}^k	Integer	Departure time of truck k from ground handler gh
$h_{gh}^{k,k'}$	Binary	Equals 1 if arrival time + waiting time of truck k' is larger or equal than the departure time + fixed buffer time of truck k from ground handler gh
v_i	Integer	Inter-node time of node i. Time before servicing another node
$q_{k,m}$	Binary	Equals 1 if truck k is assigned to dock m
$z_{m,m'}^{k,k'}$	Binary	Equals 1 if truck k is assigned to dock m and truck k' to dock m'
$y_{i,c}^k$	Binary	Equals 1 if p-node i is assigned to stack c in truck k
$l_{i,c}^k$	Integer	Load of stack c in truck k <i>after</i> servicing node i
$\hat{l}_{i,c}^k$	Integer	Load of stack c in truck k <i>before</i> servicing node i
$p_{i,j,c}^k$	Binary	Equals 1 if p-node i and j are assigned to stack c in truck k and i precedes j
$\alpha_{i,c,r}^k$	Binary	Equals 1 if p-node i is assigned to truck k at location (c,r)
$\beta_{c,r,n}^k$	Binary	Equals 1 if location (c,r) in truck k is accessible directly from the loading door at d-node n
$\gamma_{c,r,n}^k$	Binary	Equals 1 if location (c,r) in truck k is free (not used) at d-node n
$\delta_{c',c,r,n}^k$	Binary	Equals 1 if location (c,r) in truck k can be side-accessed from stack c'

3.4 CPDPTW-SAL MILP Formulation

$$\min \quad \tau_1 \left(\sum_{k \in \mathbf{K}} \sum_{ij \in \mathbf{A}} T_{ij} x_{ij}^k + \sum_{i \in \mathbf{N}} C_i + \sum_{k \in \mathbf{K}} \sum_{gh \in \mathbf{GH}} w_{gh}^k + \sum_{i \in \mathbf{V}} v_i \right) + \tau_2 \left(\sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{P}} x_{s,j}^k \right) \quad (1a)$$

s.t.

$$\sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{N}} x_{ij}^k = 1 \quad \forall i \in \mathbf{P}, \quad (1b)$$

$$\sum_{j \in \mathbf{V}} x_{ij}^k = \sum_{j \in \mathbf{V}} x_{j,n+i}^k, \quad \forall i \in \mathbf{P} \quad \forall k \in \mathbf{K}, \quad (1c)$$

$$\sum_{i \in \mathbf{V}} x_{ij}^k = \sum_{i \in \mathbf{V}} x_{ji}^k, \quad \forall j \in \mathbf{N} \quad \forall k \in \mathbf{K}, \quad (1d)$$

$$\sum_{j \in \mathbf{P} \cup \mathbf{e}} x_{s,j}^k = 1 \quad \forall k \in \mathbf{K}, \quad (1e)$$

$$\sum_{i \in \mathbf{D} \cup \mathbf{s}} x_{i,e}^k = 1 \quad \forall k \in \mathbf{K}, \quad (1f)$$

$$t_j \geq t_i + C_i + T_{ij} - M(1 - x_{ij}^k) \quad \forall k \in \mathbf{K} \quad \forall ij \in \mathbf{A}_{\mathbf{gh}} \quad \forall gh \in \mathbf{GH}, \quad (1g)$$

$$t_j \geq t_i + C_i + T_{ij} - M(1 - x_{ij}^k) \quad \forall k \in \mathbf{K} \quad \forall ij \in \mathbf{A}_{\mathbf{ff}} \quad \forall ff \in \mathbf{FF}, \quad (1h)$$

$$t_j \geq t_i + C_i + T_{ij} - M(1 - x_{ij}^k) + w_{gh}^k \quad \forall j \in \mathbf{D}_{\mathbf{gh}} \quad \forall i \in \mathbf{N} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1i)$$

$$t_j \leq t_i + C_i + T_{ij} + M(1 - x_{ij}^k) + w_{gh}^k \quad \forall j \in \mathbf{D}_{\mathbf{gh}} \quad \forall i \in \mathbf{N} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1j)$$

$$t_i \leq A_i \quad \forall i \in \mathbf{N}, \quad (1k)$$

$$t_i \geq B_i \quad \forall i \in \mathbf{N}, \quad (1l)$$

$$a_{gh}^k \geq t_i + C_i + T_{ij} - M(1 - x_{ij}^k) \quad \forall j \in \mathbf{D}_{\mathbf{gh}} \quad \forall i \in \mathbf{N} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1m)$$

$$a_{gh}^k \leq t_i + C_i + T_{ij} + M(1 - x_{ij}^k) \quad \forall j \in \mathbf{D}_{\mathbf{gh}} \quad \forall i \in \mathbf{N} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1n)$$

$$d_{gh}^k \geq t_i + C_i - M(1 - x_{ij}^k) \quad \forall i \in \mathbf{D}_{\mathbf{gh}} \quad \forall j \in \mathbf{V} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1o)$$

$$d_{gh}^k \leq t_i + C_i + M(1 - x_{ij}^k) \quad \forall i \in \mathbf{D}_{\mathbf{gh}} \quad \forall j \in \mathbf{V} \setminus \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, \quad (1p)$$

$$a_{gh}^{k'} + w_{gh}^{k'} \geq d_{gh}^k + T_b - M(1 - h_{gh}^{k,k'}) \quad \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall gh \in \mathbf{GH}, \quad (1q)$$

$$a_{gh}^{k'} + w_{gh}^{k'} \leq d_{gh}^k + T_b + M(h_{gh}^{k,k'}) \quad \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall gh \in \mathbf{GH}, \quad (1r)$$

$$\begin{aligned}
v_i &\geq t_j - (t_i + C_i) - M(1 - x_{ij}^k) & \forall i, j \in \mathbf{D}_{\mathbf{gh}} \quad \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, & (1s) \\
v_i &\geq t_j - (t_i + C_i) - M(1 - x_{ij}^k) & \forall i, j \in \mathbf{P}_{\mathbf{ff}} \quad \forall k \in \mathbf{K} \quad \forall ff \in \mathbf{FF}, & (1t) \\
\sum_{m \in \mathbf{M}_{\mathbf{gh}}} q_{k,m} &= \sum_{i \in \mathbf{V} \setminus \mathbf{D}_{\mathbf{gh}}} \sum_{j \in \mathbf{D}_{\mathbf{gh}}} x_{ij}^k & \forall k \in \mathbf{K} \quad \forall gh \in \mathbf{GH}, & (1u) \\
z_{m,m'}^{k,k'} &\leq q_{k,m} & \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall m, m' \in \mathbf{M}_{\mathbf{gh}} \quad \forall gh \in \mathbf{GH}, & (1v) \\
z_{m,m'}^{k,k'} &\leq q_{k',m'} & \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall m, m' \in \mathbf{M}_{\mathbf{gh}} \quad \forall gh \in \mathbf{GH}, & (1w) \\
q_{k,m} + q_{k',m'} - 1 &\leq z_{m,m'}^{k,k'} & \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall m, m' \in \mathbf{M}_{\mathbf{gh}} \quad \forall gh \in \mathbf{GH}, & (1x) \\
h_{gh}^{k,k'} + h_{gh}^{k',k} &\geq z_{m,m'}^{k,k'} & \forall k, k' \in \mathbf{K} \mid k \neq k' \quad \forall m \in \mathbf{M}_{\mathbf{gh}} \quad \forall gh \in \mathbf{GH}, & (1y) \\
\sum_{c \in \mathbf{C}} y_{i,c}^k &= \sum_{j \in \mathbf{V}} x_{ji}^k & \forall i \in \mathbf{P} \quad \forall k \in \mathbf{K}, & (1z) \\
x_{s,j}^k &\leq y_{j,1}^k & \forall j \in \mathbf{P} \quad \forall k \in \mathbf{K}, & (1aa) \\
l_{j,c}^k &\geq l_{i,c}^k + D_j \cdot y_{j,c}^k - M(1 - x_{ij}^k) & \forall j \in \mathbf{P} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ab) \\
l_{j,c}^k &\leq l_{i,c}^k + D_j \cdot y_{j,c}^k + M(1 - x_{ij}^k) & \forall j \in \mathbf{P} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ac) \\
l_{j,c}^k &\geq l_{i,c}^k + D_j \cdot y_{j-n,c}^k - M(1 - x_{ij}^k) & \forall j \in \mathbf{D} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ad) \\
l_{j,c}^k &\leq l_{i,c}^k + D_j \cdot y_{j-n,c}^k + M(1 - x_{ij}^k) & \forall j \in \mathbf{D} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ae) \\
l_{i,c}^k &\leq Q & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1af) \\
\hat{l}_{j,c}^k &\geq l_{i,c}^k - M(1 - x_{ij}^k) & \forall j \in \mathbf{D} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ag) \\
\hat{l}_{j,c}^k &\leq l_{i,c}^k + M(1 - x_{ij}^k) & \forall j \in \mathbf{D} \quad \forall i \in \mathbf{V} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ah) \\
\sum_{i \in \mathbf{PUs}} p_{i,e,c}^k &= 1 & \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ai) \\
\sum_{j \in \mathbf{PUe}} p_{s,j,c}^k &= 1 & \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1aj) \\
\sum_{j \in \mathbf{PUe}} p_{i,j,c}^k &= y_{i,c}^k & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ak) \\
\sum_{j \in \mathbf{PUe}} p_{i,j,c}^k &= \sum_{j \in \mathbf{PUs}} p_{j,i,c}^k & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1al) \\
t_j &\geq t_i - M(1 - p_{i,j,c}^k) & \forall i, j \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1am) \\
\alpha_{i,c,1}^k &= p_{0,i,c}^k & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1an) \\
\sum_{i \in \mathbf{P}} \alpha_{i,c,r}^k &\leq 1 & \forall r \in \mathbf{R} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ao) \\
\sum_{r \in \mathbf{R}} \alpha_{i,c,r}^k &= y_{i,c}^k & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1ap) \\
\alpha_{i,c,r-1}^k - \alpha_{j,c,r}^k &\leq 1 - p_{i,j,c}^k & \forall i, j \in \mathbf{P} \quad \forall r \in \mathbf{R} \mid r \neq 1 \quad \forall c \in \mathbf{C} \quad \forall k \in \mathbf{K}, & (1aq) \\
\hat{l}_{n,c}^k &\geq r + 0.1 - M(\beta_{c,r,n}) & \forall n \in \mathbf{D} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1ar) \\
\hat{l}_{n,c}^k &\leq r + M(1 - \beta_{c,r,n}) & \forall n \in \mathbf{D} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1as) \\
\hat{l}_{n,c}^k &\geq r - 1 + 0.1 - M(\gamma_{c,r,n}) & \forall n \in \mathbf{D} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1at) \\
\hat{l}_{n,c}^k &\leq r - 1 + M(1 - \gamma_{c,r,n}) & \forall n \in \mathbf{D} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1au) \\
\delta_{c',c,r,n}^k &\leq \gamma_{c',r,n}^k & \forall n \in \mathbf{D} \quad \forall c, c' \in \mathbf{C} \mid c \neq c' \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1av) \\
\delta_{c',c,r,n}^k &\leq Z_{c',c}^k & \forall c, c' \in \mathbf{C} \mid c \neq c' \quad \forall k \in \mathbf{K}, & (1aw) \\
\alpha_{i,c,r}^k &\leq \beta_{c,r,i+n}^k + \sum_{c' \in \mathbf{C}} \delta_{c',c,r,i+n}^k & \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K}, & (1ax) \\
x_{ij}^k, h_{gh}^{k,k'}, q_{k,m}, z_{m,m'}^{k,k'}, y_{i,c}^k, p_{i,j,c}^k &\in \{0, 1\}, & & (1ay) \\
\alpha_{i,c,r}^k, \beta_{c,r,n}^k, \gamma_{c,r,n}^k, \delta_{c',c,r,n}^k &\in \{0, 1\}, & & (1az) \\
t_i, w_{gh}^k, a_{gh}^k, d_{gh}^k, v_i, l_{i,c}, \hat{l}_{i,c} &\in \mathbb{N} & & (1ba)
\end{aligned}$$

It should be noted that although not specified in the formulation, the set of arcs has been generated as advised in section 3.2. Therefore, if an x_{ij}^k variable is introduced in a constrain, one should always check if the arc (i,j) exists in the set of arcs. When implementing the formulation in a programming environment, this should always be done but has been left out in the formulation above for clarity.

Equation 1a is the cost function consisting of 5 terms. The four terms which are multiplied by τ_1 are time related. The first term is the sum of all travel times. The second term is the process time of all nodes. The third term is the sum of all waiting times at the ground handlers. The fourth term is the sum of all inter-node times. These four terms are the total time duration of the solution. The term multiplied by τ_2 represents the active arcs from the start depot to a pickup node, which is equivalent to the total number of trucks used.

Equation 1b assigns each pickup node to a truck. Equation 1c ensures that if a pickup node is visited by vehicle k, the same vehicle visits the delivery node. In Equation 1d the flow conservation over a node is ensured. Equation 1e and Equation 1f ensure that each vehicle starts and ends at the depot respectively. Equation 1g defines the time variable within ground handler arcs. Equation 1h defines the time variable within freight forwarder arcs. Equation 1i and Equation 1j define the time variable into a ground handler from a freight forwarder or another ground handler and introduce the waiting time variable. The time window constraints are introduced by Equation 1k and Equation 1l. In Equation 1m and Equation 1n the arrival time of a truck at a ground handler is introduced. The departure time of a truck from a ground handler is introduced by Equation 1o and Equation 1p. In Equation 1q and Equation 1r the binary overlap variable is turned on or off. If the arrival time plus the waiting time of truck k' is larger or equal than the departure time plus the fixed buffer time of truck k, this variable is turned on. The inter-node time at ground handlers and freight forwarders is introduced in Equation 1s and Equation 1t respectively. In Equation 1u the dock assignment parameter is assigned if the truck travels to a ground handler. Equation 1v, Equation 1w and Equation 1x enforce the z variable to have the right value. Equation 1y ensures that trucks assigned to the same dock do not overlap.

In Equation 1z the pickup node is assigned to one stack if the node is picked up by a vehicle. The symmetry is broken by introducing Equation 1aa which ensures that the first node that is visited is always assigned to the first stack. Equation 1ab and Equation 1ac ensure that the load parameter is updated for the pickup nodes. Equation 1ad and Equation 1ae do the same for the delivery nodes. This extra set of constraints is needed as $y_{i,c}^k$ is only defined for the pickup nodes. Equation 1af ensures the capacity of the stacks is never exceeded. Equation 1ag and Equation 1ah assign the variable which represent the load of a node *before* servicing the node.

The following constraints are relevant for the LIFO loading constraints in the pickup route. Equation 1ai and Equation 1aj ensure that each stack-route starts and ends at the depot. In Equation 1ak all nodes are placed in the right stack-route, based upon the assigned stack. Equation 1al ensures the flow conservation in the stack-route parameter. The order of the nodes in the stack-route parameter is ensured by introducing the time parameter as is done in Equation 1am. The first node which is visited by a stack in the pickup route has to be assigned closest to the driver cabin, which is ensured in Equation 1an. In Equation 1ao it is ensured that each location in the stack can be occupied by at most one node. In Equation 1ap it is ensured that a pickup node is assigned to the right stack as previously introduced. Finally, the right order within the stacks is ensured by Equation 1aq. If the right hand side of this equation is equal to zero, this implies that i and j are assigned to stack c and that i is the predecessor of node j . This also implies the left hand side of the equation needs to be zero. Therefore, the first and second term should both be zero or both be one. If i is assigned to location (c,r-1), this thus means that j has to be allocated to location (c,r). If i is not allocated to (c,r-1), j cannot be assigned to location (c,r).

The following constraints are relevant for the LIFO loading constraints in the delivery route. If the load of a stack *before* servicing is less or equal than r , this means that the location is directly accessible. This is ensured in Equation 1ar and Equation 1as. The $+ 0.1$ value in Equation 1ar is introduced to break ties where $\hat{l} = r$. If the load of a stack *before* servicing is less or equal than $r-1$, this means that the location is free. This is ensured in Equation 1at and Equation 1au. Again, the $+ 0.1$ value in Equation 1at is introduced to break ties where $\hat{l} = r - 1$. The side-accessible parameter is turned on if an adjacent stack has a free spot at the same r -location. This is ensured in Equation 1av. Side access is only allowed for adjacent stacks and is ensured in Equation 1aw. Finally, in Equation 1ax the side-accessible LIFO constraint is satisfied. This implies that each delivery node should be accessible directly from the unloading door and/or it should be side-accessible. Finally, Equation 1ay, Equation 1az and Equation 1ba define the nature of the decision variables.

When using the *no-LIFO* model variant, Equation 1ax is simply relaxed. If the *strict-LIFO* model variant is considered, this means that Equation 1ax is changed to Equation 2. In this equation, the side-accessible component is not included.

$$\alpha_{i,c,r}^k \leq \beta_{c,r,i+n}^k \quad \forall i \in \mathbf{P} \quad \forall c \in \mathbf{C} \quad \forall r \in \mathbf{R} \quad \forall k \in \mathbf{K} \quad (2)$$

3.5 Stack-Routing Formulation

In the previous formulation, the $p_{i,j,c}^k$ variable indicates if pickup node i and j are assigned to stack c in truck k and i directly precedes j . This formulation is introduced to allow for the right allocation of the $\alpha_{i,c,r}^k$ variable as is done in Equation 1a_q. What is done in the $p_{i,j,c}^k$ variable, is represented in the Figure 2

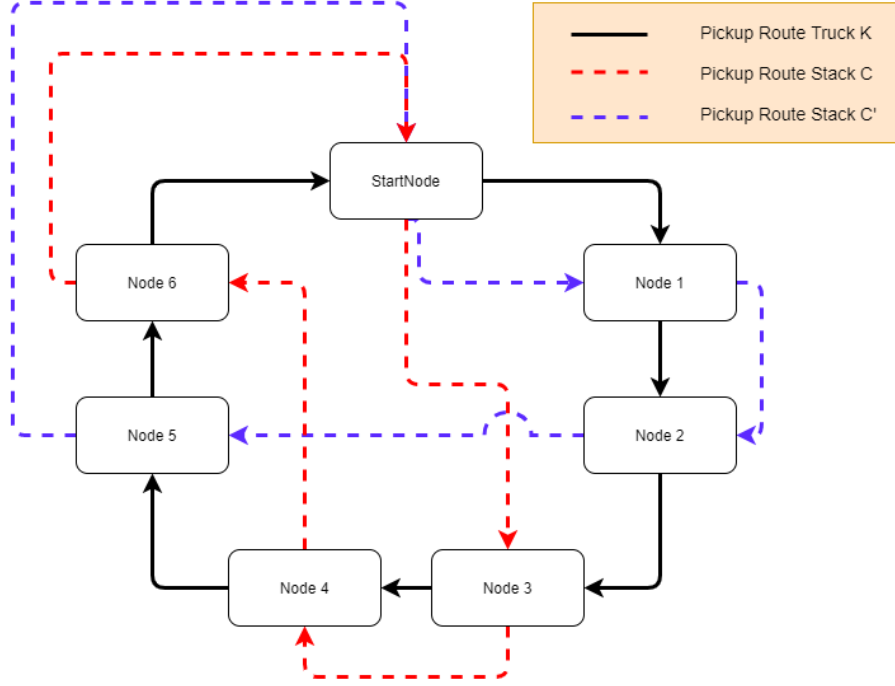


Figure 2: Stack-route representation for a two stack system

First of all, *Truck K* travelled the pickup route: $[1, 2, 3, 4, 5, 6]$ and the truck is based on a two stack system. For each stack, an individual route is created. The truck route and the stack-routes starts and ends at a depot node (represented in the figure as 'StartNode'). In this case, this means that two stack-routes are created. The route of *Stack C*, is represented as: $[3, 4, 6]$. For *Stack C'*, this route is: $[1, 2, 5]$. Due to this additional variable, it is known over what order the nodes are visited in the pickup route for each individual stack. With the introduction of Equation 1a_q, the $\alpha_{i,c,r}^k$ is correctly assigned. If this constraint would not have been implemented, the ordering of shipments for each individual stack cannot be done in the right way.

4 A Large Neighborhood Search Approach

A meta-heuristic method is set up that is based on of a large neighborhood search. First of all, in section 4.1 the general outline of the heuristic is presented. After that, in section 4.2 an overview of all heuristic moves is presented. Finally, in section 4.3 the side-accessible feature in the heuristic model is presented. It should be noted that when generally stated that 'a node is inserted or removed', this implies that the pickup and associated delivery node are both inserted or removed. If specifically the pickup or delivery node is meant, this is explicitly stated.

4.1 General LNSA Approach

The heuristic algorithm consist of three phases. 1) Start Phase, 2) Repair Phase, 3) Improvement Phase. The purpose of the *Start Phase* is to create routes where only the time window constraint of the pickup and delivery nodes is considered. If a node cannot be inserted into an existing route, a new truck is initialized for the node. At the end of the start phase, the dock capacity constraint and loading constraints are checked. If a route is not feasible with respect to dock capacity, the entire route is removed and the nodes from the route are inserted in the requestbank. If a violation is found for a route in the loading constraint, one random node from the route is removed. The loading feasibility is rechecked and nodes are removed until a load-feasible route was found. The removed nodes are inserted in the requestbank.

The requestbank and solution from the start phase are forwarder to the *Repair Phase*. The goal of this phase is to create a feasible solution where all nodes are implemented and no dock or loading constraint violations are present. This outcome is the initial solution and used for improving the solution in the next phase. If the start phase algorithm was already dock-, and load-feasible for all routes, the repair phase is thus not needed. For all requests in the requestbank it is tried to implement it in the current solution. This time, the dock capacity and loading constraint are considered immediately. If a feasible implementation position is found, the request is implemented in the solution and removed from the requestbank. If implementing a node from the requestbank in the current truck routes is not possible, it is tried to create a new truck for this node. At the end of the repair phase, a feasible solution is created that satisfies all constraints and has implemented all requests. The solution produced in the repair phase is thus the initial solution.

The *Improvement Phase* aims at improving the solution from the Repair Phase. This is done by a series of heuristic moves which are described in more detail in the next section. First of all, S_b is the best known solution at that moment in the algorithm, with associated best known cost, C_b . Before an iteration, the start solution is referred to as S_o , with cost C_o . The solution after one heuristic move (one iteration) is referred to as S_n , with cost C_n . After an iteration is performed, the start solution for the next iteration is updated using the simulated annealing accepting criteria. If $C_n \leq C_b$, S_b is set to S_n . If $C_n \leq C_o$, S_n is used as the start solution in the next iteration. If $C_n > C_b$, the simulated accepting criteria determines the start solution for the next iteration. To prevent the solution from iterating for large number of iterations on a solution which is not optimal, it is enforced that the algorithm takes the best known solution as the start solution for the next iteration if an improvement was not found in the last ten iterations.

4.2 Heuristic Moves

This section will present the heuristic moves that are used in the large neighborhood search algorithm.

- **Remove and Implement Move**

- Removal Moves

- * *Worse Removal*

In the worse removal heuristic move, the node with the highest cost in the solution is selected and removed from the route. The node is implemented in another position in a route at the position where the cost is the lowest. The worse node is selected as follows. First of all, the cost of the current route containing the node, f_r , is computed. The node is then removed from the solution and the cost of the new route is equal to f_n . The difference between f_r and f_n is the benefit of removing *node n* from the solution, and thus the cost of the node and is represented by c_n . This is done for all nodes and the node where c_n is the largest, is labelled as the worse node in the solution and is removed. The removed node is inserted in the requestbank.

- * *Random Removal*

In the random removal heuristic move, a random node is selected and removed from the solution. The node is inserted in the requestbank.

- Implementation Move

No horizontal collaboration is allowed between freight forwarders, so the node can only be implemented in trucks that are already departing from the same freight forwarder where the node is

located. For each node in the requestbank a set of vehicles is constructed that depart from the corresponding freight forwarder. For these vehicles, it is tried to implement the pickup and delivery node at all feasible positions. For each combination, the loading constraints are checked immediately. If the loading constraint is satisfied, the dock capacity constraint is checked. If a feasible implementation location is found, the implementation option is stored with the associated solution cost. After setting up all feasible implementation options, the option is selected where the total solution cost is the lowest. If no feasible position was found in the current vehicles, a new truck is initialized and it is tried to implement the node in this new truck. If the node was implemented in the existing set of vehicles, or in a new truck, the node is removed from the requestbank. If no feasible implementation position was found, the node is kept in the requestbank for a later iteration. The new found solution will then probably not be accepted since a missed delivery introduces a significant penalty.

- **Merge Trucks**

Two trucks are selected that pickup nodes from the same freight forwarder. It is tried to merge these routes into a single truck. For data instances where strict time windows are included, all feasible combinations with respect to the time windows are tried. If a feasible combination is found, the algorithm is stopped and the combination is accepted. If the data instance does not have strict time windows, not all feasible combinations are tried, because there would be too many to compute in a reasonable time span. For that reason, the pickup nodes are sorted according to their time windows. The delivery nodes are also ordered according to their time windows within a ground handler. The possible combinations to visited the ground handlers are then determined and it is tried to find a feasible combination.

- **Destroy route and rebuild**

A random vehicle is selected and the entire route is removed. This implies that all nodes visited by the vehicle are inserted in the requestbank. To implement the requestbank, the previously explained *Implementation Move* is used.

- **Change two nodes in one freight forwarder**

A freight forwarder is selected where at least two vehicles are departing from. From each vehicle, a node is randomly selected. The selected nodes are changed between the two trucks. In Figure 3(a) an example is presented of two trucks departing from the same freight forwarder. It should be noted that S represents the start depot node and E the end depot node. The 1^+ node represents the pickup node of ULD item 1 and 1^- represents the delivery node. In Figure 3(b) the move is visualized where node 1 and 3 are changed between the two trucks.

- **Change two nodes in one route**

A random vehicle is selected that has at least two pickup nodes in the route. Two pickup nodes are randomly selected from the route. The location of the two pickup nodes is then switched. A visual representation of this heuristic move is presented in Figure 3(c). In this specific case *truck 1* is selected to undergo the 'change two nodes in one route' move.

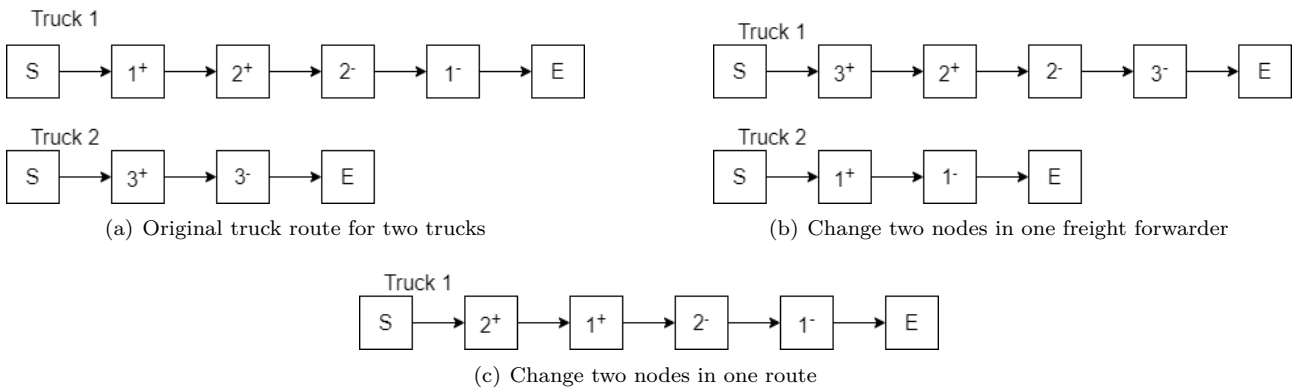


Figure 3: Change two nodes in one route and in one freight forwarder heuristic visualization

- **Change ground handler dock**

To prevent situations where the assignment of docks to trucks is not optimal, this heuristic move is introduced. In this move, it is checked if another dock at the same ground handler is available for a truck. If this is the case, the assigned dock is changed. This heuristic move allows another truck to be assigned to the dock which got available. In Figure 4(a) a situation is presented where truck 4 has to be assigned

to one of the docks but a feasible position cannot be found. If the '*change ground handler dock*' heuristic is applied, truck 2 is moved to dock 2. Truck 4 can then be implemented in dock 1 as is visualized in Figure 4(b).

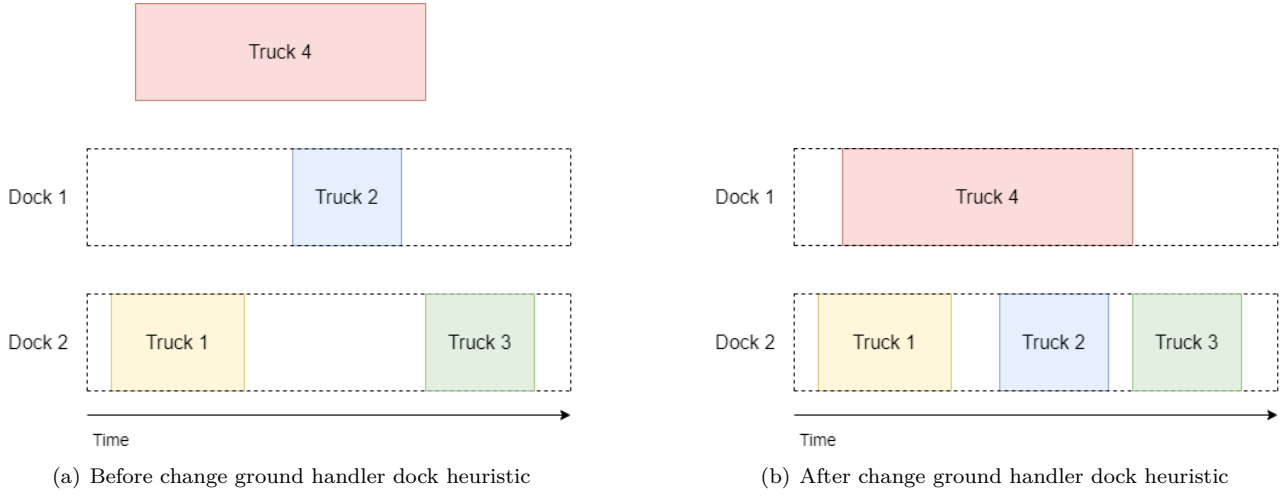


Figure 4: Change ground handler dock heuristic visualization

- **Split Truck**

This heuristic is used to diversify the search and create better solutions for specific data instances. This heuristic move is especially relevant for models where the objective function is solely time based, and no additional fixed cost per truck is charged. A random truck is selected and in its route a random node is selected. This node is removed from the current route and a new vehicle is initialized for the node.

4.3 Side-Accessible LIFO constraint

The side-accessible LIFO constraint is introduced in the model by checking the pickup and delivery route of a vehicle. In Figure 9 (presented in Appendix A), a structured method to generate all feasible loading patterns is presented. This tree is based on a truck with two stacks where each stack has a capacity of three. The pickup route of the truck is simply: [1, 2, 3, 4, 5, 6]. If a location is represented by N, this means that the location is not used. It can be observed that a total of ten unique loading patterns are feasible for this pickup route.

The next step is the delivery route of the vehicle. For the delivery route, each unique loading pattern is checked. If a feasible loading pattern for the delivery route is found, the iteration is stopped and the route is thus load-feasible. For example, the delivery route is: [6, 5, 2, 1, 4, 3]. If the model variant is the *strict-LIFO*, this implies that only pattern 7 and 8 are load feasible. If *Side-Accessibility* is also allowed, this implies that pattern 4 is also load feasible. If the delivery route [2, 1, 6, 5, 4, 3] is considered, no feasible loading pattern can be matched to this route (for the strict-LIFO and side-accessible LIFO model variant). This implies that the combination between this pickup and delivery route is not feasible and the route is not accepted. The no-LIFO model does not consider any LIFO loading constraints, meaning that all pickup and delivery route combinations - that do not exceed the truck capacity - will be accepted.

5 Results

This section presents the results of the CPDPTW-SAL formulation. First of all in section 5.1 the study area, Schiphol Airport, is presented. In parameters for the objective function are determined in section 5.2. The comparison between the MILP formulation and meta-heuristic is presented in section 5.3. In section 5.4 the benefit of the side-accessible model variant is presented. Finally, section 5.5 presents the sensitivity analysis of the results.

5.1 Study Area

The study area that is used for this work is the Schiphol Airport area. In this area, five freight forwarders and five ground handlers are considered, presented in Table 4. The locations of these freight forwarders and ground handlers are presented in Figure 5. In this figure, freight forwarders and ground handlers are represented in blue and red respectively. In the remainder of this work, data instances will be referred to as $n\text{-}ff\text{-}gh$. In this formulation, n represents the number of shipments in that data instance. In addition, ff and gh refer to the number of freight forwarders and ground handlers respectively. It is assumed that each ground handler has two docks available. The locations of the freight forwarders and ground handlers are based on their longitudinal and latitudinal coordinates. The travel time between two locations is computed using the *OpenStreetMap* database.

Table 4: Freight Forwarder and Ground Handler location

Freight Forwarder	Ground Handler
DHL Global Forwarding (DHL)	KLM Ground Handling (KLM)
Expeditors International Forwarding (EX)	Dnata (DNT)
Kuehne Nagel (KN)	Menzies (MNZ)
DB Schenker (SCH)	Worldwide Freight Services (WFS)
UPS (UPS)	Swissport (SCS)



Figure 5: Location of freight forwarders (blue) and ground handlers (red)

5.2 Objective Function Parameter Determination

Two type of objective functions are introduced. First of all a time-based objective function. This is the summation of the duration of all individual routes. In the cost-based objective function a fixed cost is introduced for each additional truck that is used.

In the research of [Engholm et al., 2020] an in depth economic analysis is made for driverless operations of trucks. It starts by stating the cost of current trucks which are driven by humans and the information below is deducted from that. The following cost are incorporated in this work. For this model, only the driver wage is included as a time dependent cost and is equal to \$27.04/hour, which is \$0.45/min. The cost of dispatching an additional truck is based on the acquisition cost of the truck (\$170,000) with a ownership period of 8 years. The residual value of the truck can be derived from the values in [Engholm et al., 2020] and was estimated to be approximately \$12,500. It is assumed that a year has on average 260 working days. From Equation 3 is can be concluded that the cost per working day is thus equal to \$75.72/day.

$$\frac{170000 - 12500}{8 \cdot 260} = \$75.72/day \quad (3)$$

In addition to that, [Engholm et al., 2020] mention other fixed cost such as insurance and taxes. These costs are equal to approximately \$9500/year, which is \$36.54/day. In total, the cost of dispatching truck is thus equal to \$112.26/day. It is realistic to assume that each dispatched truck services one route per day, and thus each truck has a fixed cost of \$112.26. When referring back to the objective function, Equation 1a, τ_1 is thus equal to \$0.45 and τ_2 is equal to \$112.26.

The cost of dispatching an additional truck is not included in the time-based objective function. This means that only the time duration of a route is included. The cost of the solution can then also simply be stated as minutes.

5.3 Comparison MILP formulation with Meta-Heuristic Model

This section will present the comparison of the MILP formulation with the meta-heuristic model. This will be done for both the cost- and time-based objective function. The MILP formulation of the cost-based objective function has acceptable optimality gaps for small datasets and the optimality gap increases rapidly with increasing dataset size. The MILP formulation of the time-based objective function produces acceptable optimality gaps for medium size instances. For that reason, different datasets are presented for the different objective functions. In the columns of the tables of this section for the MILP formulation, *Cost* represents the cost of the solution. This is represented in dollars, or minutes, depending on the objective function. *Gap* represents the optimality gap as a percentage. *Time* is the computational time of the model in seconds that were needed to achieve the optimal solution. If the optimal solution is not found, this parameter represents the time when the optimization was cutoff. The results for the meta-heuristic model are based on a total of five runs and 200 iterations. The *Cost* represents the mean cost of the solution for the five runs, in dollar or minutes. *Time* represents the mean computational time that the algorithm needed to complete one run. In section 5.3.1 the comparison between the MILP formulation and the meta-heuristic model for the cost-based objective function is presented. The comparison for the time-based objective function is presented in section 5.3.2. The evaluation of the meta-heuristic model is based on three evaluation criteria. First of all, the solution quality of the meta-heuristic model with respect to the MILP formulation. The solution quality of the meta-heuristic method should be in an acceptable range of the MILP formulation, or even better. The second aspect is the computational time of the meta-heuristic model. Finally, the stability of the meta-heuristic model can be expressed as the difference between the best and worse run. The stability should also be in an acceptable range to have a reliable outcome of the meta-heuristic model.

5.3.1 Cost Based Objective Function

This section presents the results for the cost-based objective function of small data instances. A comparison is made between the MILP formulation and the meta-heuristic method to validate the meta-heuristic model performance. The results are presented in Table 5.

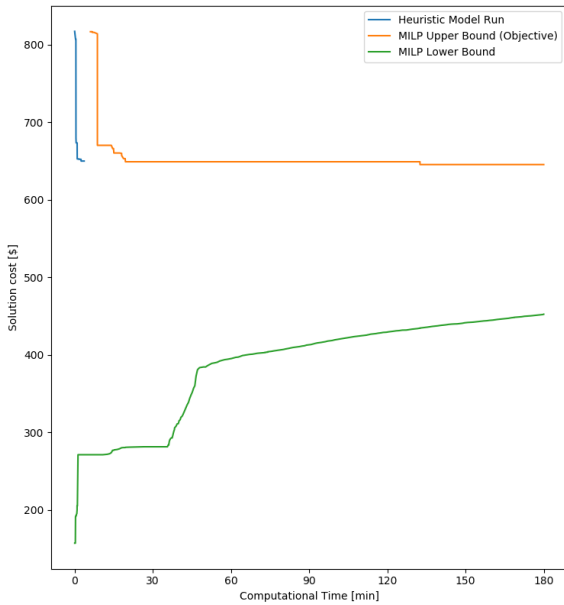
Table 5: Results for small size datasets on a cost-based objective function. Trucks have a two stack system with 3 locations in each stack.

Dataset	No-LIFO			Side-accessible LIFO			Strict-LIFO			Strict-LIFO			Strict-LIFO		
	MILP			Meta-heuristic			MILP			Meta-heuristic			MILP		
	Cost	Gap	Time	Cost	Time		Cost	Gap	Time	Cost	Time		Cost	Gap	Time
10-3-3	497.2	0	115	497.4	23		497.2	0	109	497.2	24		497.2	0	146
11-3-3	518.0	0	235	518.9	38		518.0	0	298	518.2	42		518.0	0	233
12-3-3	500.4	3.1	3600	500.7	91		500.4	0	308	500.4	85		520.2	0	963
13-3-3	663.1	0	758	663.1	55		663.6	0	532	663.6	54		780.8	9.2	3600
14-3-3	531.0	0.6	7200	532.6	73		531.0	1.8	7200	532.9	67		531.0	0	6641
15-3-3	645.5	21.9	10800	657.0	124		645.5	29.9	10800	654.6	146		655.9	32.7	10800
16-3-3	686.5	23.0	10800	691.0	133		685.6	22.5	10800	687.7	131		685.6	22.5	10800

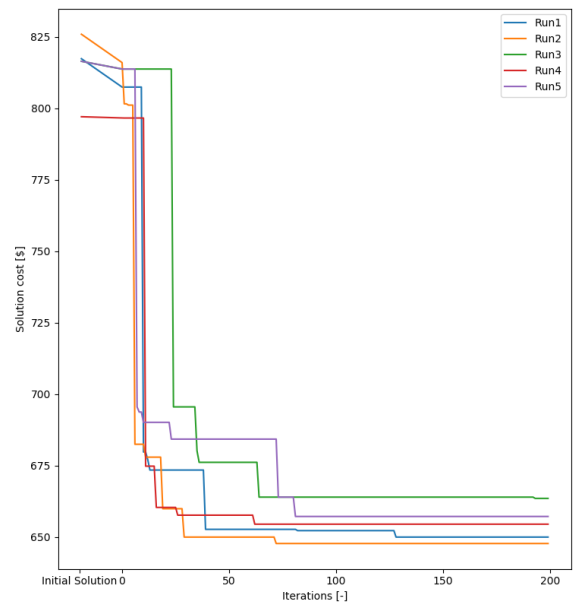
It can be observed from Table 5 that the solution quality of the meta-heuristic is in range of the MILP formulation. If the meta-heuristic model solution is not equal to the solution of the MILP formulation, the maximum difference between the two models is approximately 1.8%, which is assumed to be very good. In terms of computational time, it can be observed that for all datasets and model variants the meta-heuristic model outperforms the MILP formulation. The stability of the five individual runs is not presented in this table due to limited table size, but the difference between the runs for this small data instance is on average 0.74%, which is assumed to be acceptable.

In data instance 12-3-3 it becomes clear that the benefit of the side-accessible model variant is about 3.8% with respect to the strict-LIFO model variant. No extra truck is required in this strict-LIFO model variant, but the solution still benefits from allowing Side-Accessibility. Especially interesting is to observe data instance 13-3-3 where the benefit of the side-accessible LIFO loading is about 15%. The strict-LIFO model requires an additional truck in the solution, which comes at the cost of approximately \$110, as calculated in section 5.2. For the other data instances, no significant benefit of the Side-Accessibility was observed.

It can also be observed that the optimality gap of the model increases rapidly with data instance size. Although it is expected that the best solution found in the exact model is close to the optimal solution, this cannot be proofed because the optimality gap is unacceptable large. In Figure 6(a) the development of the upper and lower bound of the exact model are plotted. In addition to that, one of the meta-heuristic model runs is presented. In Figure 6(b) the development of the five individual runs is presented with respect to the number of iterations.



(a) Computational results of MILP formulation with respect to meta-heuristic model



(b) Development of different meta-heuristic runs

Figure 6: Results for the side-accessible model variant for the 15-3-3 data instance with a cost based objective function

From Figure 6(a) it can be observed that the MILP lower bound is increasing very slowly over time. It is expected that this slow increase is mostly due to the fact that a fixed cost is introduced for each additional truck that is used. Increasing the size of future data instances will thus not give reliable results. It is concluded that the meta-heuristic model performs well for cost based objective function in terms of solution quality,

computational time and stability. For larger data instances, the meta-heuristic model can thus be used. The next section will show that the meta-heuristic model also provides reliable results for the time-based objective function. In Figure 6(b) it can be seen that after 200 iterations the improvement of the solution has stabilized and thus no more iterations are needed.

5.3.2 Time Based Objective Function

From the previous section it was concluded that the meta-heuristic model performs well for the cost-based objective function on small data instances. To validate the meta-heuristic model for larger data instances, the time-based objective function is introduced. In Table 6 the performance of the exact and meta-heuristic model is compared for medium sized data instances. Again, the results are based on the average results of five runs. For data instances with 25 ULDs or more, it was decided to increase the number of iterations to 300. In addition to that, from now on, end time windows at the freight forwarders and start time windows at the ground handlers are relaxed. This simulates the most realistic scenario where a pickup node only has a start time window, describing the time at which it is ready for picking up. The delivery nodes only have an end time window, describing the time at which it should be at the ground handler to be on time for its scheduled flight. For the medium-to-large data instances with these specific time windows, new data instances were generated. The 15-3-3 data in this section is therefore slightly different than the one presented in the previous section.

Table 6: Results for medium size datasets on a time-based objective function. Trucks have a two stack system with 5 locations in each stack.

Dataset	No-LIFO					Side-Accessible LIFO					Strict-LIFO				
	MILP			Meta-heuristic		MILP			Meta-heuristic		MILP			Meta-heuristic	
	Cost	Gap	Time	Cost	Time	Cost	Gap	Time	Cost	Time	Cost	Gap	Time	Cost	Time
15-3-3	429	12.3	7200	431	126	429	12.1	7200	430	178	429	12.4	7200	431	440
17-3-3	492	10.2	7200	493	188	496	10.9	7200	498	433	502	11.8	7200	506	436
20-3-3	585	9.1	18000	575	498	592	10.1	18000	575	521	596	10.7	18000	587	1198
25-3-3	693	12.0	43200	678	2373	698	12.6	43200	686	2779	2779	15.4	43200	694	3605
30-3-3	-	-	43200	813	2128	-	-	43200	813	1825	-	-	43200	820	2267
30-4-4	-	-	43200	867	1934	-	-	43200	871	2267	-	-	43200	870	3371

From Table 6 it can be concluded that the solution quality of the meta-heuristic model is very close to the MILP formulation for all data instances. If the MILP formulation performs better, the difference is very small and acceptable. For some other data instance, the meta-heuristic even outperforms the MILP formulation. For data instances with 30 ULDs or more, the MILP formulation could not find a feasible solution in the allowed iteration time of 12 hours. The meta-heuristic method was however capable of finding a solution in an acceptable computational time duration. The computational time of the meta-heuristic is always a fraction of the MILP formulation. Again, the stability of the model is not presented in the table due to table size. In section 5.5.1 the full table for these results are presented and it is shown that the difference between the *Min* and *Max* of the 5 individual runs is limited to 2.0%, which is assumed to be acceptable. The benefit of the side-accessible model variant is present in most data instances, but not as significant as in the cost-based objective function. This will be elaborated in the sensitivity analysis of this section, presented in section 5.5.

From this section it is concluded that the meta-heuristic model performs well on the three identified criteria with respect to the MILP formulation. The solution quality is close to the MILP formulation and for larger data instances better than the MILP formulation. In terms of computational time, the meta-heuristic model also outperforms the MILP formulation. The stableness of the model is also within an acceptable range. The meta-heuristic model is thus assumed to be working reliable for both the cost- and time-based objective function.

5.4 Side-Accessible LIFO Model benefit

This section will present a more detailed analysis of the benefit of the side-accessible model with respect to the strict-LIFO model variant. In Table 7 an overview of medium to large size data instances is presented on a cost-based objective function. The results from now on are based on the meta-heuristic model because larger data instances cannot be solved by the MILP. In this table, the full range of results are presented which is based on 5 individual runs. In this table, *Min* represents the solution with the minimum costs which was found by the meta-heuristic. *Mean* represents the mean solution cost of all 5 runs and *Max* presents the solution cost of the worse solution. Finally, *Time* represents the mean computational time for the 5 runs in seconds. The most right column, *Benefit SA*, represents the benefit of the side-accessible model variant with respect to the strict-LIFO variant. The benefit is calculated based on the best solution which was found by the meta-heuristic for each of the model variants. This is done to prevent assigning a too large benefit to the side-accessible model variant, which could be the case if the mean value is taken. It is believed that this approach represents the benefit of the side-accessible model variant as fair and realistic as possible.

Table 7: Results for medium to large size datasets on a cost-based objective function for the meta-heuristic model. Trucks have a two stack system with 5 locations in each stack.

Dataset	No-LIFO				Side-Accessible LIFO				Strict-LIFO				Benefit SA
	Min	Mean	Max	Time	Min	Mean	Max	Time	Min	Mean	Max	Time	%
15-3-3	642.8	643.1	643.3	71	642.8	643.6	645.5	100	643.7	756.3	870.5	343	0.1
17-3-3	559.4	582.6	674.8	144	674.4	675.1	675.7	301	675.7	743.7	790.7	469	0.2
20-3-3	708.6	710.0	714.9	405	708.6	708.6	708.6	375	824.5	872.6	941.7	572	14.1
25-3-3	981.8	1007.6	1098.6	2165	875.4	984.2	1096.3	2026	985.9	1098.2	1215.3	2676	11.2
30-3-3	1148.6	1152.8	1155.8	1595	1150.9	1154.3	1159.4	1755	1266.3	1318.7	1499.4	2432	9.1
30-4-4	1176.5	1207.5	1309.1	1616	1173.8	1180.6	1188.7	1704	1287.0	1369.2	1435.3	2357	8.8
35-3-3	1208.1	1351.1	1450.2	2083	1326.2	1397.7	1446.1	2378	1330.3	1396.6	1440.7	2739	0.3
35-4-4	1456.5	1502.8	1572.8	2233	1459.2	1482.3	1567.0	2211	1459.7	1639.7	1801.4	2983	0.0
40-4-4	1614.7	1686.3	1739.2	2452	1614.3	1687.0	1736.5	2569	1842.9	1849.0	1856.8	3171	12.4
50-5-5	2102.5	2236.4	2330.2	6034	2098.9	2234.8	2335.1	4791	2433.9	2532.1	2680.5	6315	13.8

First of all, the stableness of the model is looked into. The average difference between the minimum and maximum run is 10.6% which is assumed to be acceptable. It should be noted that the stableness of the model might seem to be performing worse than presented in section 5.3. It is expected that this is due to two reasons. First, the size of the data instances is increased. Intuitively, it is expected that the variance of the solution quality increases if the size of the instance is increased. The second reason is the fixed cost per truck which is introduced in the cost-based objective function. Only if all runs per data instance and model variant come to the exact same number of trucks, the difference in solution quality could seem to be acceptable. Especially for larger data instances, this behaviour is difficult to implement in the meta-heuristic. It is however concluded that the average difference between the minimum and maximum of the runs is in an acceptable range for reliable results. Especially since the benefit of the side-accessible model variant is based on the best run.

If the *Benefit SA* column is studied, it should be noted that for all data instances the benefits of the side-accessible model variant is larger or equal than zero. This is in line with the expectations as the side-accessible model variant should always perform the same or better than the strict-LIFO model variant. It is also visible that the SA-benefit fluctuates per data instance and no average percentage can be determined. The benefit of the side-accessible model variant for most data instances is mostly due to an extra truck which is required in the strict-LIFO model variant. This fixed cost per truck introduces significant benefit of the side-accessible model variant in some data instances. If the SA-benefit is close to zero, this indicates that no additional truck is required between the two models variants. If an additional truck is used in the strict-LIFO model variant, the benefit of the side-accessible model variant is usually in the order of 10%.

It is however difficult to predict what data instances will benefit from the side-accessible model variant. The combination of time windows, freight forwarder location, ground handler location and number of ULDs per freight forwarder or ground handler seem to have an impact on the benefits of the side-accessible model variant.

In Appendix B the development of the solution cost is plotted for the 20-3-3 data instance. In Figure 10 this is done for the cost-based objective function. It can be seen that there are clear jumps down when the iterations are increased. If a jump down decreases the solution cost about \$120, this means that a new solution was found where the number of trucks is reduced by one. In Figure 11 the development of the solution cost is plotted for a time based objective function. It should be noted that the scale on the y-axis of this plot is zoomed in a lot further than the cost-based objective function. Also in this data instance, a step down approach can be observed. But for this objective function there are no clear jumps of removed trucks visible. For both situations it can be seen that the five individual runs have found a stable plateau after the 200 iterations. It is thus assumed that for data instances with this size, 200 iterations is enough. For data instances with 25 ULDs or more, it was decided to increase the number of iterations from 200 to 300. This was done after analyzing the results and it was concluded that in some cases 200 iterations is not enough. In Figure 12 the development of the 40-4-4 data instance is presented that is on a cost-based objective function. Again, each run shows clear plateau's and jumps down if a truck is removed from the solution. In Figure 13 the plot is presented for the time-based objective function. Due to the nature of this objective function, no clear plateau's and jumps are present. This is in line with the expectations. After 300 iterations the runs have stabilized and if a better solution was found recently, the increase is only very limited. It is concluded that 300 iterations in a good trade-off between computational time and solution quality.

In Figure 14 (presented in Appendix C) the solution for the 35-3-3 data instance is presented. This figure is presented to give an idea on the final routing of a medium sized data instance. In this figure freight forwarders are represented by the grey horizontal bars and each ground handler is given an individual colored bar where the different docks are separated by the dotted line in the middle of the bar. Each line represents an individual truck route. The numbers above the line represent the ULD node ID. The small vertical lines represent the start and end time at which the node is serviced. At the side of the ground handlers, it should be noted that no trucks overlap at a dock. In addition to that, after each truck leaves the ground handler, a fixed buffer time (15 minutes) is introduced before another truck can arrive. At the side of the freight forwarders, it can

be seen that there is some overlap between different trucks. This is however acceptable as freight forwarders are responsible for their own truck schedule and no docking constraints are introduced to these locations. The solution from Figure 14 is also (partly) presented in Figure 15 as a plot in the Schiphol Airport area. It should be noted that not all routes are plotted, due to visual purposes. Only the routes that can be combined best for visual purposes are presented in this graph.

5.5 Sensitivity Analysis

This section presents the sensitivity analysis of the results and the impact on the benefit of the side-accessible model with respect to the strict-LIFO model variant. First of all, in section 5.5.1 the sensitivity analysis with respect to the chosen objective function is presented. After that, in section 5.5.2 the sensitivity analysis with respect to truck capacity is presented.

5.5.1 Objective Function

This section presents the results for the same data instances as in Table 7, but this time the objective function is time-based and not cost-based. The results are presented in Table 8. It should be noted that the *Mean* results presented in Table 8 correspond to the values for the meta-heuristic presented in Table 6. An overview of the *Benefit SA* column for the two objective functions is plotted in Figure 7.

Table 8: Results for medium to large size datasets on a time-based objective function for the meta-heuristic model. Trucks have a two stack system with 5 locations in each stack.

Dataset	No-LIFO				Side-Accessible LIFO				Strict-LIFO				Benefit SA
	Min	Mean	Max	Time	Min	Mean	Max	Time	Min	Mean	Max	Time	%
15-3-3	430.0	430.6	432.0	126	429.0	430.4	431.0	178	430.0	431.2	432.0	440	0.2
17-3-3	492.0	493.2	494.0	188	492.0	498.4	504.0	433	501.0	506.0	510.0	436	1.8
20-3-3	575.0	575.0	575.0	498	575.0	575.2	576.0	521	584.0	586.6	589.0	1198	1.5
25-3-3	672.0	677.8	688.0	2373	678.0	685.8	698.0	2779	680.0	694.0	704.0	3605	0.3
30-3-3	805.0	813.4	822.0	2128	805.0	812.8	821.0	1825	813.0	819.8	826.0	2267	1.0
30-4-4	857.0	866.8	890.0	1934	859.0	871.0	887.0	2267	862.0	870.4	883.0	3371	0.3
35-3-3	944.0	951.6	959.0	2922	937.0	949.6	966.0	2895	945.0	960.0	978.0	4918	0.8
35-4-4	980.0	991.0	999.0	3588	984.0	990.2	995.0	2806	991.0	1004.6	1013.0	6367	0.7
40-4-4	1075.0	1092.0	1109.0	2956	1077.0	1086.8	1101.0	3140	1089.0	1100.8	1117.0	3637	1.1
50-5-5	1401.0	1412.2	1423.0	4448	1404.0	1426.4	1449.0	5004	1414.0	1432.4	1442.0	4404	0.7

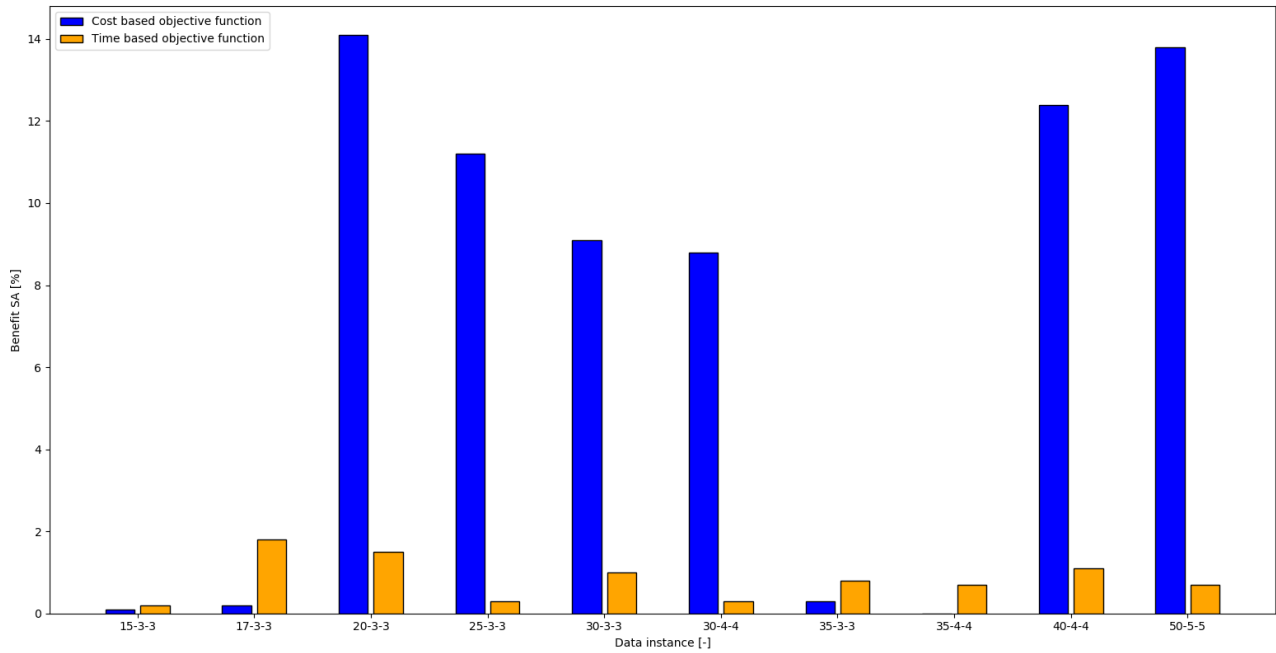


Figure 7: Sensitivity analysis of objective function on the benefit of the side-accessible model variant

The average difference between the *Min* and *Max* is equal to 2.0% for this objective function. This is significantly smaller than the value presented in section 5.4. Because the time-based objective function does not

introduce an additional fixed cost per truck, the model can have similar solution qualities although the number of trucks is not the same. Using the time-based objective function thus seems to benefit the stableness of the solution.

When the results of the two objective functions are compared in Figure 7, it is observed that the time-based objective function is generally below the cost-based objective function. All data instances have a benefit, but it is only limited up to 2%. The most intuitive reason for the reduced benefit of the side-accessible model variant for the time-based objective function is that cost of additional trucks are not incorporated. A new truck can be initialized without any additional cost. In the cost-based objective function, the solution cost significantly increases if a new truck is initialized. The cost-based objective function thus tries to create trucks that have a very high load factor. The time-based objective function does not necessarily do this. Instinctively, the side-accessible model variant has most benefit if trucks are filled up close to capacity. For trucks that are only partially filled, the side-accessible model variant is in most cases not beneficial. This behaviour is reflected in the results of Table 8 and Figure 7 and explains why the benefit of the side-accessible model variant is generally higher for the cost-based objective function compared to the time-based objective function.

5.5.2 Truck Capacity

This section presents determines the performance of the three model variants with respect to the truck capacity. Instead of a separate full table with the results for each truck capacity, this time the solution cost is plotted for the different model variants as this is the most insightful. The results are on a cost-based objective function. If the runs are considered for all different truck capacities, the average difference between the *Min* and *Max* is limited to 9.1%. It is concluded that the model output is reliable to perform this sensitivity analysis. In Figure 8 the solution cost for six different data instances is presented for the different model variants, based on the truck capacity. The results are based on the best run of the 5 runs for each model variant. The decision to plot the best run is made based on the same argumentation as presented in section 5.4. The trucks are based on a two-stack system, implying that the capacity of a stack is 2, 3, 4 and 5 respectively.

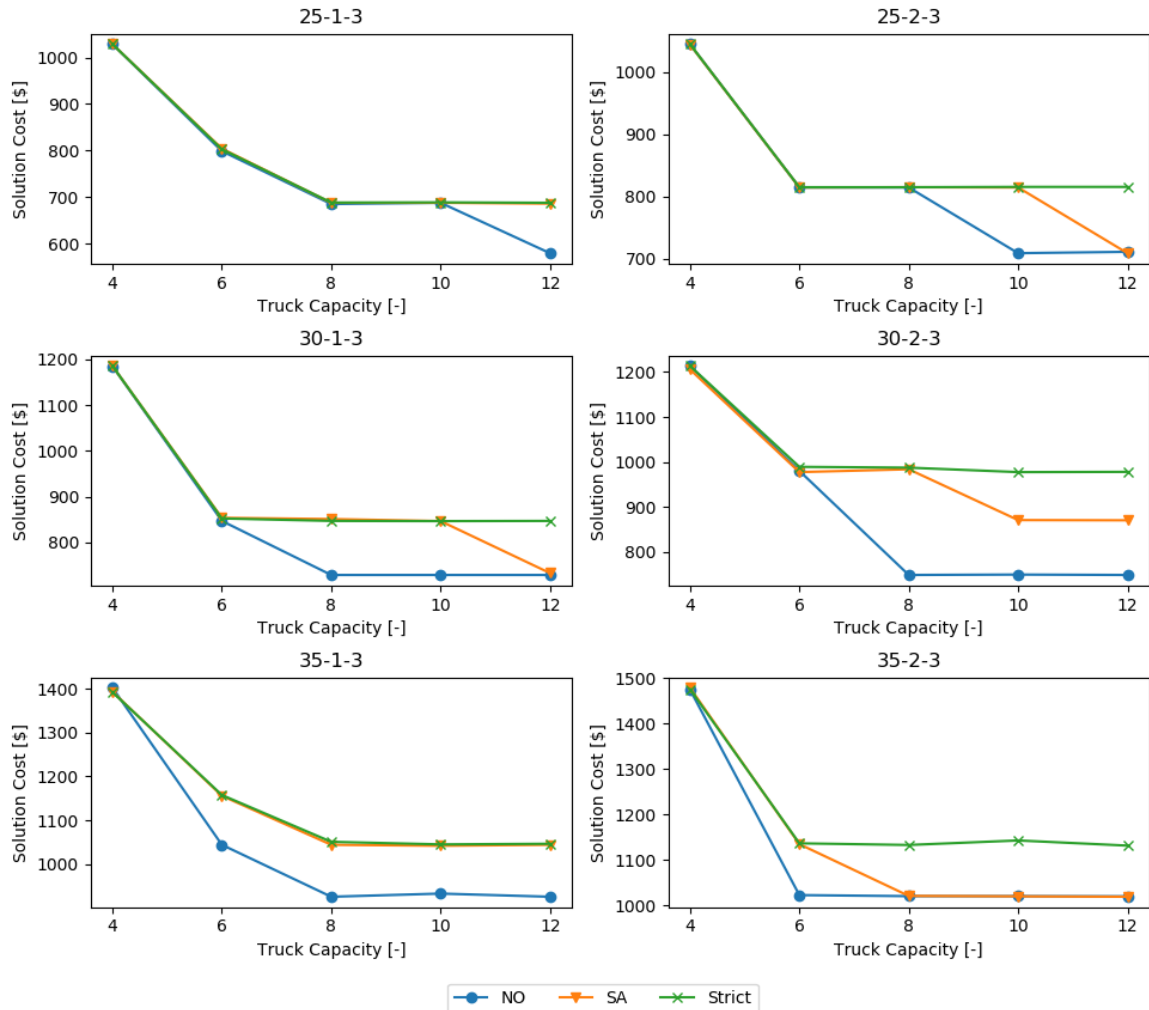


Figure 8: Sensitivity analysis based on truck capacity for different data instances

It is expected that the solution cost decreases (or stays the same) if the truck capacity is increased. A descending trend in all lines is thus expected. In addition to that, it is expected that the no-LIFO line will be the line with the lowest cost and the strict-LIFO line has the highest cost. The SA-LIFO model variant line should be in between them. It is also possible that one or more lines coincide with each other. In general, it can be observed that increasing the capacity from 4 to 6 or from 6 to 8 results in the best improvement. In some cases, increasing the capacity up to 10 or 12 gives an additional improvement. For all data instances, it can be observed that the no-LIFO model variant performs best, followed by the side-accessible model variant. The strict-LIFO model variant always has the highest cost, or coincides with the side-accessible and strict-LIFO model variant. This behaviour is in line with the expectations. For data instance 25-1-3 and 35-1-3, no clear benefit of the side-accessible model can be observed. The lines of the strict-LIFO and side-accessible model variant are very close to each other. For the other data instance, a clear benefit of the side-accessible model variant can be observed for one (or more) of the truck capacities. For the 25-2-3 and 30-1-3 data instances, the benefit of the side-accessible model is clear for a truck capacity of 12. The benefit of the side-accessible model for the 30-2-3 data instance is at a truck capacity of 10 and 12. The 35-2-3 data instance has a clear SA-benefit for a truck capacity of 8, 10 and 12.

From this section it is concluded that the truck capacity significantly influences the side-accessible model variant benefit. If the truck capacity is 4 or 6, the benefit is relatively small. If the capacity is increased, the benefit of the side-accessible model variant is more apparent. This is in line with the expectation of the model. One can imagine that if the truck capacity is increased, the possibilities to introduce side-accessible access to shipments is increased significantly. It is however very difficult to predict what the benefit of the side-accessible model will be for a specific data instance and truck capacity. The pickup and delivery location and time windows of the data instance nodes determine the possible benefit of the model.

6 Conclusion

The last few decades witnessed a rise in the volumes of air-transported cargo. This rise was not always matched by an increase in size of the supply chain, which has fostered the creation of several bottlenecks and inefficiencies along the air cargo supply chain. In this paper, we have presented different mathematical models that incorporate different strategies to mitigate some of those issues that affect cargo export ground operations. In particular, we developed a model, called *The Clustered Pickup and Delivery Problem with Time Windows and Multi Stack Side-Accessible Last-in First-out Loading*, that considers a set of freight forwarders and optimizes the scheduling of their deliveries to ground handlers. Although they do not collaborate explicitly, their schedules are optimized by a central planner in such a way that delays due to dock unavailability on the ground handler side are minimized. In addition to that, a novel formulation is presented for the Last-in First-out constraint that includes a side-accessible feature in the delivery route.

In our study, we considered three different variations of the Last-in First-out delivery strategy. First, no constraints are imposed on the unloading sequence in the delivery route. Second, only the last item in a stack is accessible. Third, side-access between stacks is allowed. Two different cost functions were considered, one is solely based on the time duration of the total solution. The second objective function is cost-based and introduces a fixed penalty for each truck that is used. We solved instances of different sizes with a branch-and-bound method, and with a meta-heuristic. Data instances up to 14 ULDs could be solved to optimality by the MILP formulation in two hours. Increasing the size of the data instance for the cost-based objective function results in a drastic increase in the optimality gap percentage. The time-based objective function is used for data instances up to 25 ULDs and a feasible solution was found within 12 hours. The meta-heuristic is based on a large neighborhood search and proved to be performing well for both objective functions in terms of solution quality, computational time, and solution stableness for multiple runs.

From a practical perspective, it was proven how side-accessibility can provide a significant reduction in cost, especially when the cost-based objective function is considered. In fact, providing extra-flexibility in the way deliveries are carried out might increase the load factor per truck and the routing options, hence reducing the fleet size. A cost reduction up to 15.0% with respect to the current standard in multi-stack formulations in literature is found. When the time-based objective function is used, the benefit is limited to 1.8%. This is easily explained by the fact that no additional cost per truck is charged, leading to low load factors of the trucks. The benefit of the side-accessible feature however becomes more significant if the load factor of a truck is high.

Note that, in this work, the way positions inside a trailer are defined implies some form of container standardization. This is of course true if unit load devices are considered, but it is also in line with current trends in logistics, such as the Physical Internet, that advocate for a higher standardization of boxes and containers when transporting freight of any kind. Hence, we believe our model to be useful in real operations and to be applicable to paradigms such as the Physical Internet, once fully implemented. We also want to point out that, when side-accessibility is considered, we assumed that a shipment that is side-accessible can be correctly offloaded. This assumption should be verified in real-world operations and might pose some challenges if large

containers are considered.

Many interesting research directions naturally stem from this research. First, from an algorithmic perspective, different solution methods could be tested to improve the solution quality of large-sized instances. One should focus especially on designing an algorithm that is stable with respect to the number of trucks that is used in a solution for different runs. Second, it could be interesting to add horizontal collaboration to the model. Although it was argued this is not easily implementable in current supply chains, due to the selfishness of stakeholders, the additional benefit of such an approach could be investigated. Third, the unique feature of this model is that the side-accessible model variant can be introduced to a broad spectrum of applications. One could think of the supply of stock to supermarkets for example. Researching the other practical applications of the side-accessible feature is therefore something that could be done. Finally, while here all parameters were considered deterministic, in real-world operations traveling times, processing times, etc., are stochastic in nature. Although a fixed buffer time was introduced for trucks leaving the ground handlers to account for this deterministic behaviour, it would be advisable to define a framework that can account for such variability.

References

- [Air Cargo Netherlands, 2016] Air Cargo Netherlands (2016). Milkrun results after 1 year. <https://acn.nl/milkrun/milkrun-resultaten/?lang=en>. Retrieved on: 15-10-2020.
- [Ankersmit et al., 2014] Ankersmit, S., Rezaei, J., and Tavasszy, L. (2014). The potential of horizontal collaboration in airport ground freight services. *Journal of Air Transport Management*, 40:169–181.
- [Basso et al., 2019] Basso, F., D’Amours, S., Rönnqvist, M., and Weintraub, A. (2019). A survey on obstacles and difficulties of practical implementation of horizontal collaboration in logistics. *International Transactions in Operational Research*, 26(3):775–793.
- [Boeing, 2018] Boeing (2018). World air cargo forecast 2018 - 2037. https://www.boeing.com/resources/boeingdotcom/commercial/about-our-market/cargo-market-detail-wacf/download-report/assets/pdfs/2018_WACF.pdf. Retrieved on: 25-09-2020.
- [Boysen et al., 2013] Boysen, N., Briskorn, D., and Tschöke, M. (2013). Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum*, 35(2):479–504.
- [Brar and Saini, 2011] Brar, G. S. and Saini, G. (2011). Milk run logistics: literature review and directions. In *Proceedings of the world congress on engineering*, volume 1, pages 6–8. WCE.
- [Cordeau et al., 2010] Cordeau, J., Iori, M., Laporte, G., and Salazar González, J. J. (2010). A branch and cut algorithm for the pickup and delivery traveling salesman problem with lifo loading. *Networks*, 55(1):46–59.
- [Côté et al., 2012] Côté, J.-F., Gendreau, M., and Potvin, J.-Y. (2012). Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks. *Networks*, 60(1):19–30.
- [Dantzig and Ramser, 1959] Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- [El-Sherbeny, 2010] El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, 22(3):123–131.
- [Engholm et al., 2020] Engholm, A., Pernestål, A., and Kristoffersson, I. (2020). Cost analysis of driverless truck operations. *Transportation Research Record*, 2674(9):511–524.
- [Iori and Martello, 2010] Iori, M. and Martello, S. (2010). Routing problems with loading constraints. *TOP*, 18(1):4–27.
- [Iori and Riera-Ledesma, 2015] Iori, M. and Riera-Ledesma, J. (2015). Exact algorithms for the double vehicle routing problem with multiple stacks. *Computers & Operations Research*, 63:83–101.
- [Iori et al., 2007] Iori, M., Salazar-González, J.-J., and Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2):253–264.
- [Junqueira et al., 2013] Junqueira, L., Oliveira, J. F., Carravilla, M. A., and Morabito, R. (2013). An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research*, 20(5):645–666.
- [Landschützer et al., 2015] Landschützer, C., Ehrentraut, F., and Jodin, D. (2015). Containers for the physical internet: requirements and engineering design related to fmcg logistics. *Logistics Research*, 8(1):8.
- [Laporte, 1992] Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358.
- [Laporte et al., 2014] Laporte, G., Ropke, S., and Vidal, T. (2014). *Vehicle Routing: Problems, Methods, and Applications*, book section Chapter 4: Heuristics for the Vehicle Routing Problem, pages 87–116. SIAM.
- [Mangoubi and Mathaisel, 1985] Mangoubi, R. S. and Mathaisel, D. F. X. (1985). Optimizing gate assignments at airport terminals. *Transportation Science*, 19(2):173–188.
- [Martínez and Amaya, 2013] Martínez, L. and Amaya, C. A. (2013). A vehicle routing problem with multi-trips and time windows for circular items. *The Journal of the Operational Research Society*, 64(11):1630–1643.
- [Miao et al., 2009] Miao, Z., Lim, A., and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research*, 192(1):105–115.

- [Montreuil, 2011] Montreuil, B. (2011). Toward a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2):71–87.
- [Ou et al., 2010] Ou, J., Hsu, V. N., and Li, C.-L. (2010). Scheduling truck arrivals at an air cargo terminal. *Production and Operations Management*, 19(1):83–97.
- [Pollaris et al., 2014] Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., and Limbourg, S. (2014). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330.
- [Prakash and Deshmukh, 2010] Prakash, A. and Deshmukh, S. (2010). Horizontal collaboration in flexible supply chains: a simulation study. *Journal of Studies on Manufacturing*, 1(1):54–58.
- [Rais et al., 2014] Rais, A., Alvelos, F., and Carvalho, M. S. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.
- [Rieck and Zimmermann, 2010] Rieck, J. and Zimmermann, J. (2010). A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181(1):337–358.
- [Roca-Riu et al., 2015] Roca-Riu, M., Fernández, E., and Estrada, M. (2015). Parking slot assignment for urban distribution: Models and formulations. *Omega*, 57:157–175.
- [Ropke and Pisinger, 2006] Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.
- [Verduijn et al., 2019] Verduijn, T., Vunderink, S., Bombelli, A., Tavasszy, L., Damme, D. v., Romero-Silva, R., and Mujica Mota, M. (2019). *Smartest Connected Cargo Airport Schiphol: Onderzoek naar innovaties die de logistieke prestaties van luchtvrachtketens op Schiphol versterken*. Amsterdam: Hogeschool van Amsterdam, Urban Technology.
- [Wang et al., 2015] Wang, S., Zeyuan, L., Wei, L., Ji, G., and Yang, J. (2015). Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem. *SIMULATION*, 91.
- [Xu and Bailey, 2001] Xu, J. and Bailey, G. (2001). The airport gate assignment problem: mathematical model and a tabu search algorithm. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, page 10 pp.

Appendices

A Appendix 1

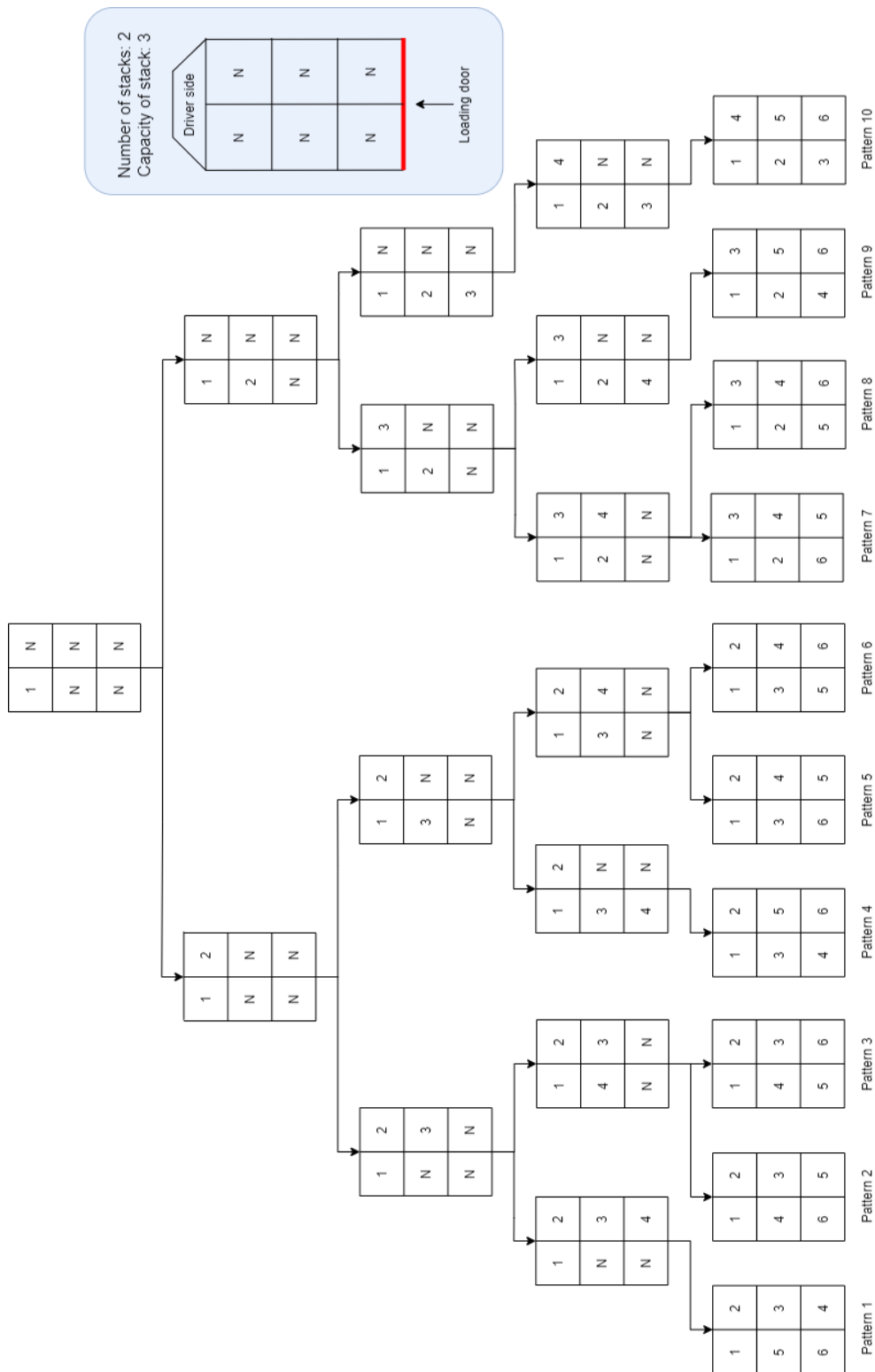


Figure 9: Truck loading pattern generation process based on pickup route: [1, 2, 3, 4, 5, 6]

B Appendix 2

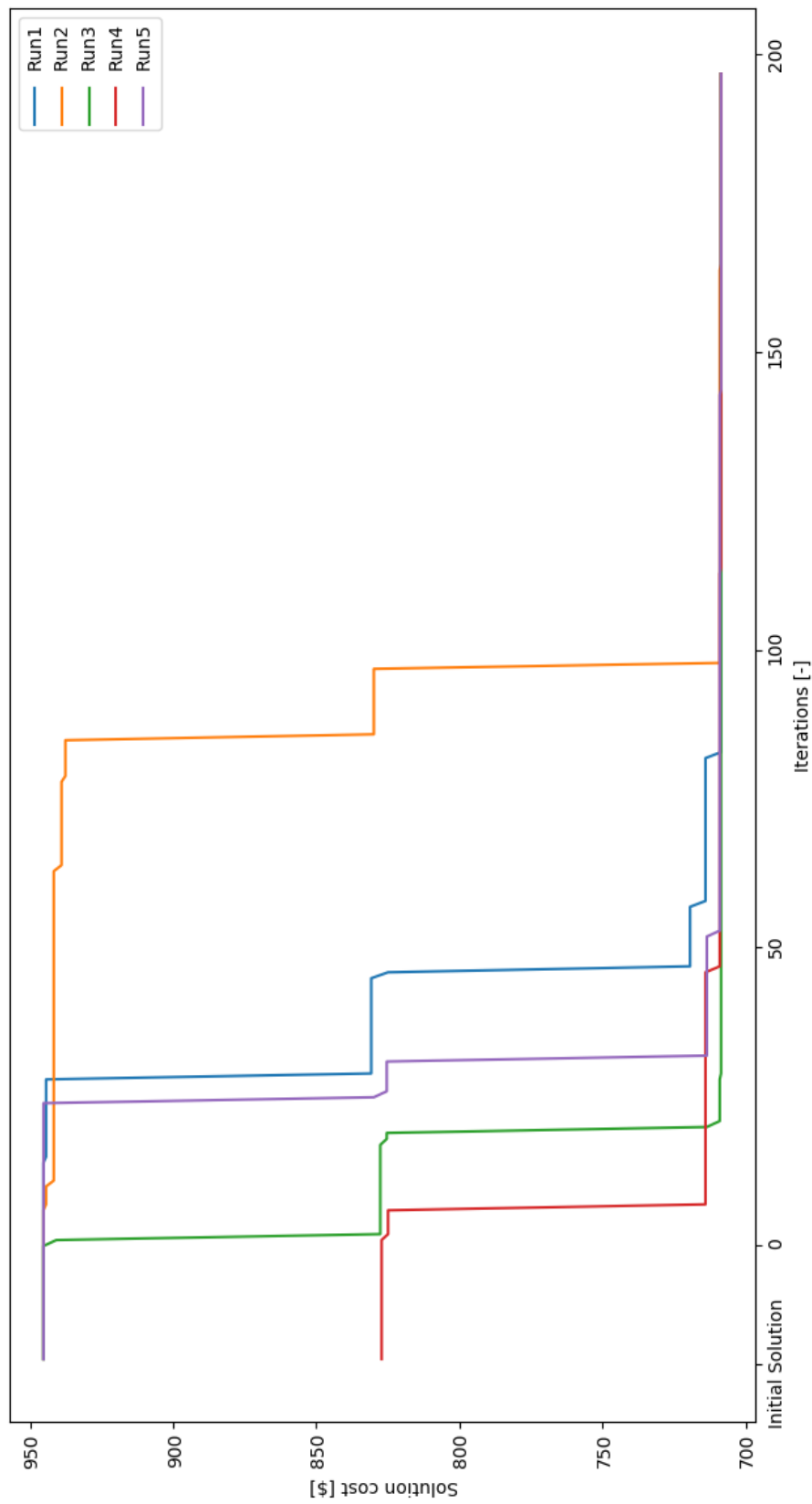


Figure 10: Development of different heuristic runs for 20-3-3 data instance for the side-accessible model variant on a cost based objective function.

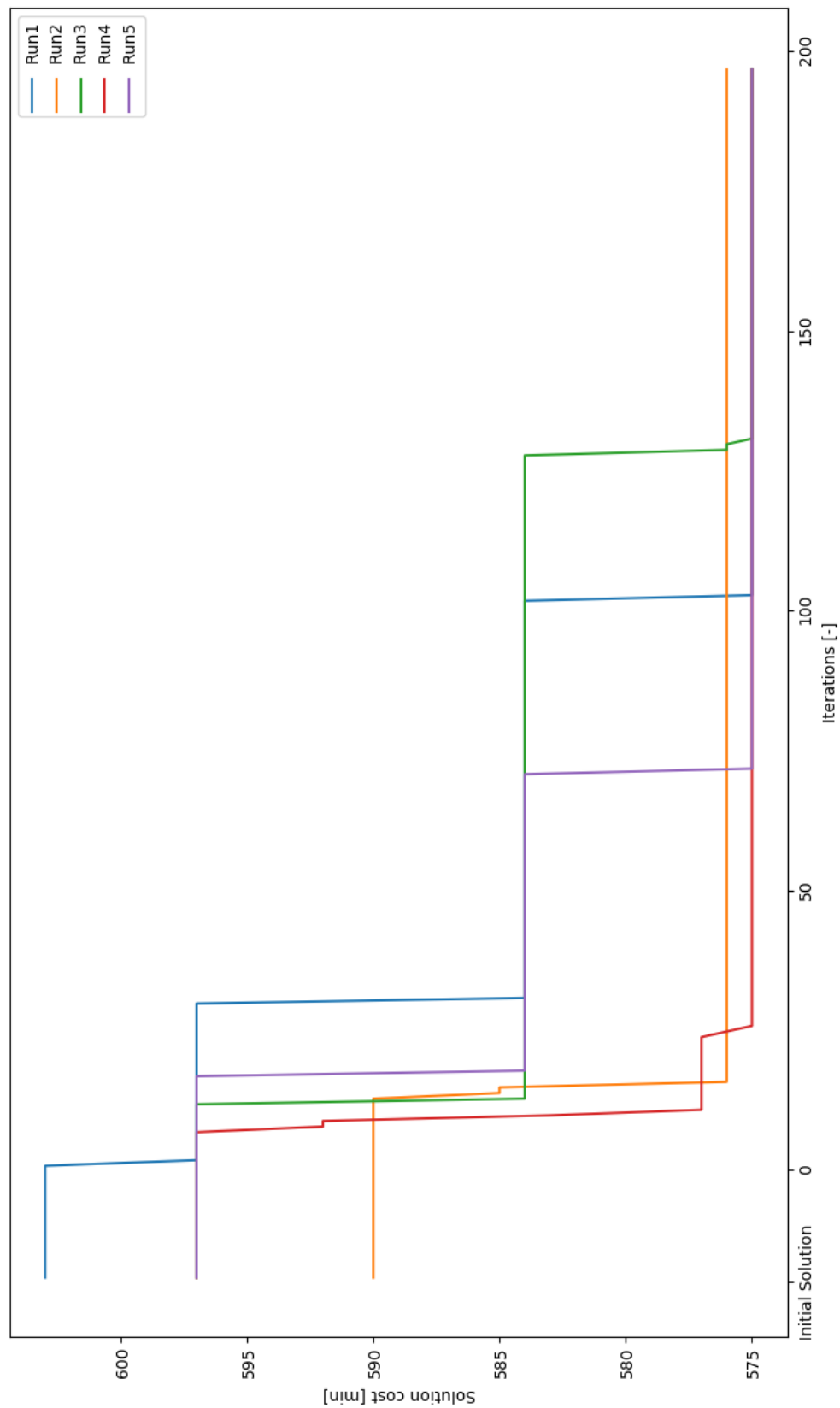


Figure 11: Development of different heuristic runs for 20-3-3 data instance for the side-accessible model variant on a time based objective function.

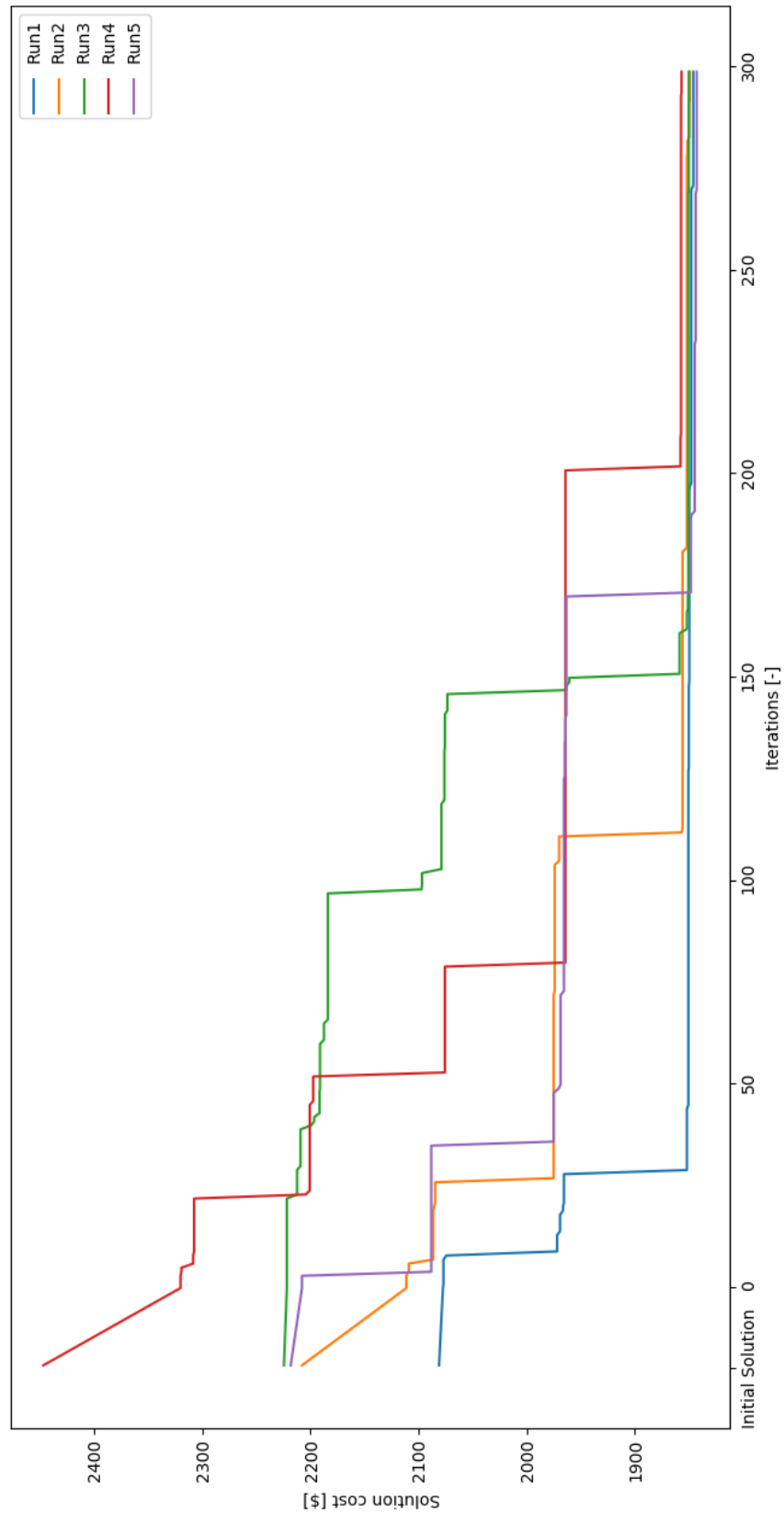


Figure 12: Development of different heuristic runs for 40-4-4 data instance for the strict-LIFO model variant on a cost based objective function.

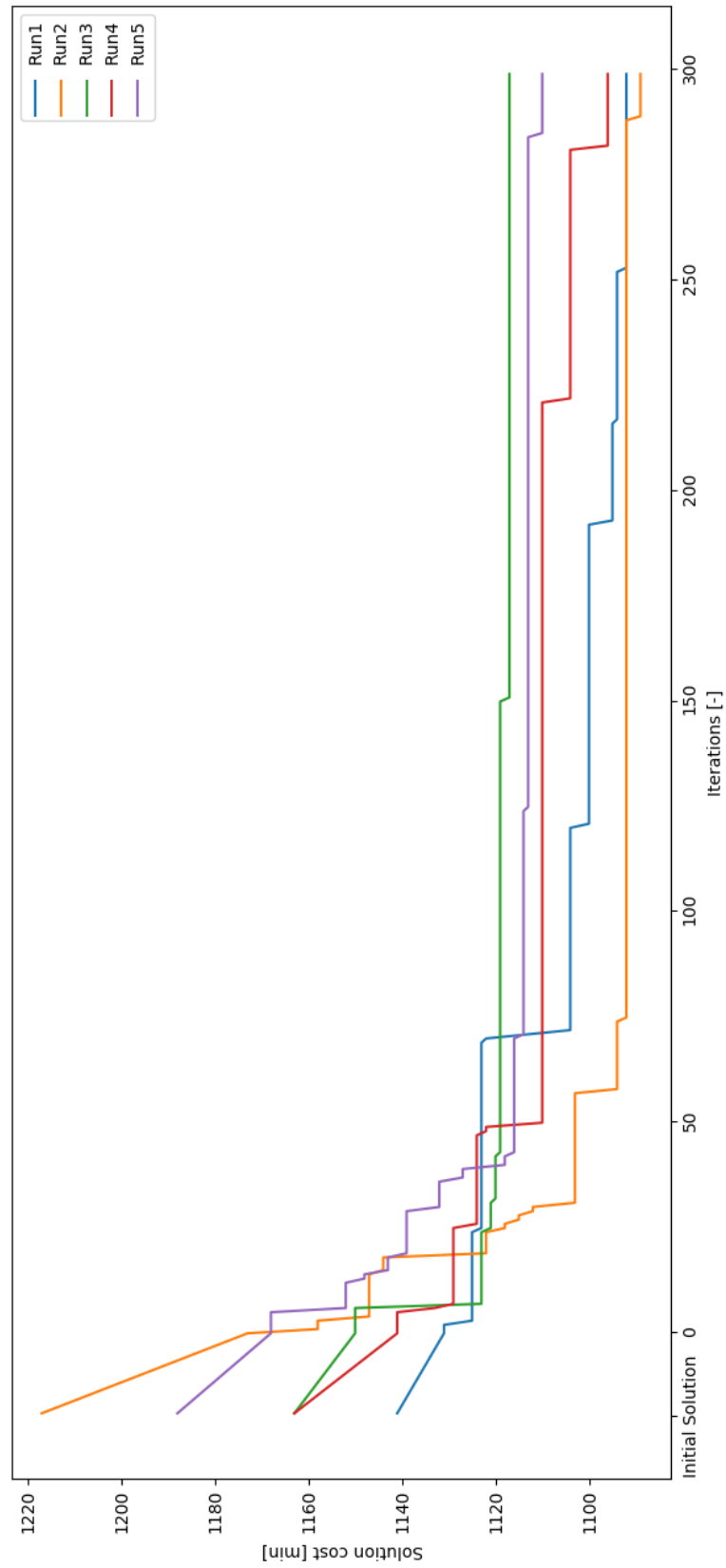


Figure 13: Development of different heuristic runs for 40-4-4 data instance for the strict-LIFO model variant on a time based objective function.

C Appendix 3

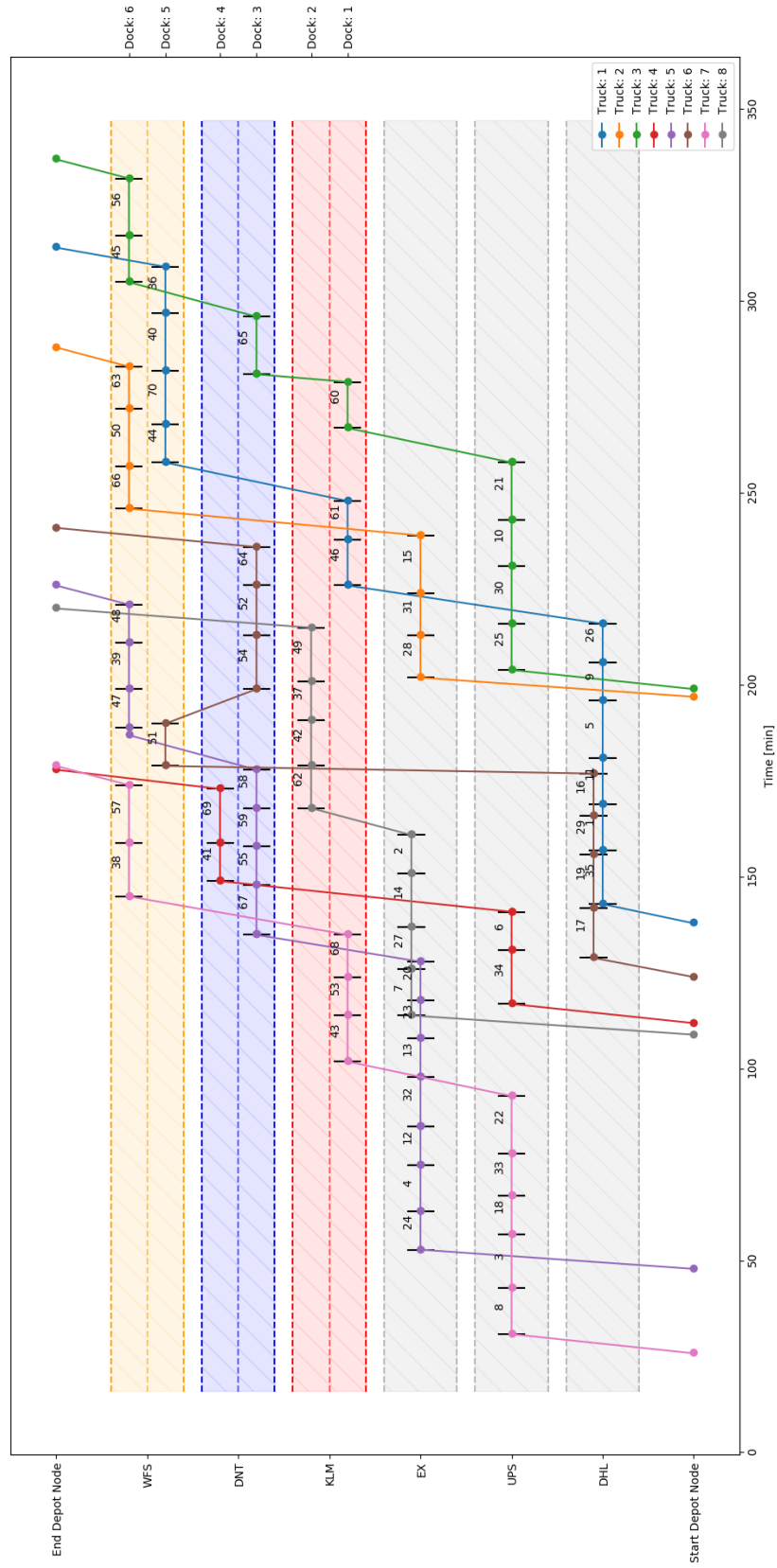


Figure 14: Time-space network for the 35-3-3 data instance for the side-accessible LIFO model variant

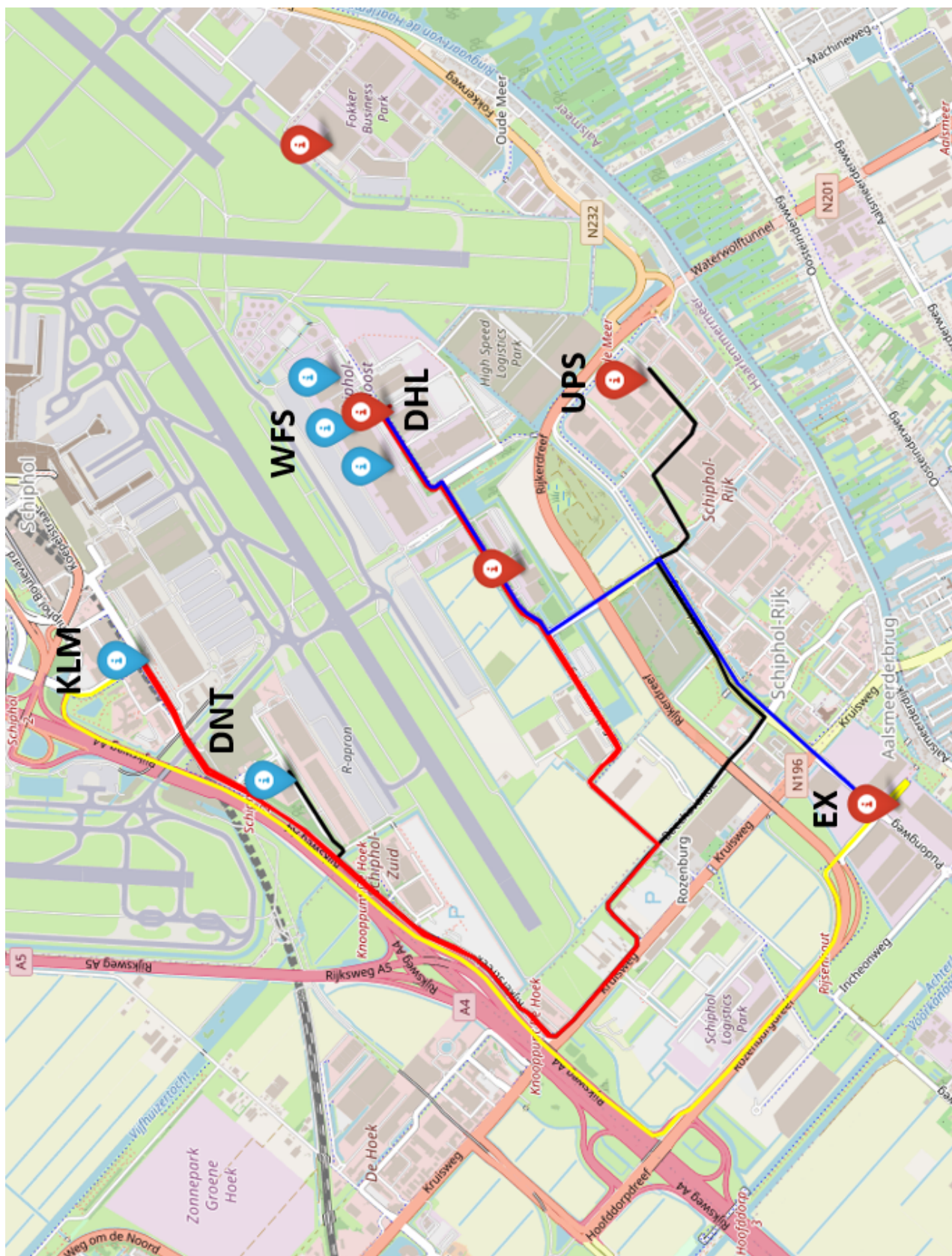


Figure 15: Routing of 35-3-3 data instance plotted in Schiphol Airport area. The red line represents truck 1, blue represents truck 2, black represents truck 4, yellow represents truck 8.

II

Literature Study

Introduction

With the availability of a large network covered by aviation, the possibility of transporting cargo by air is also very interesting. Transporting cargo by air is a fast method for shipping long distances. In addition to that, the time at which a shipment is departing or arriving is often known in advance because of preexisting flight schedules. Boeing estimated that the transported cargo volume by air will increase by 4.2% each year (before the coronavirus affected the aviation industry) [2]. Cargo can be transported either via dedicated full-cargo freighters or in the belly of passenger aircraft. Because of international travel bans and dropped demand due to the coronavirus, passenger aircraft movements have dropped significantly. The drop in cargo demand has however not seen such an extreme drop as is observed for passenger transport. In the early stages of the pandemic, medical equipment was flown all over the world. When the 'Air Transport Movements Full Freighter Services' of May 2020 are compared to May 2019, the movements have more than doubled [3]. The air cargo supply chain thus needs to scale up as the pressure on it is also increasing.

In the current air cargo supply chain at Schiphol Airport, freight forwarders and ground handlers play an important role. The freight forwarders receive packages from various companies and are responsible to transport them to the airport. The ground handlers receive and process the shipments from the freight forwarders which arrive by truck. From the ground handlers, the shipments are transported to the airside of the airport onto their flight. The arrival and departure of cargo trucks at the ground handlers is thus a daily business and a smooth operation is crucial for an efficient supply chain. The arrival of cargo trucks is however uncoordinated and happens with peaks. Peaks happen at the end of a workday, just before the weekend or after the weekend. During these peaks, the number of trucks arriving at the ground handlers exceeds the number of available docks [4]. The problem is defined by Verduijn as follows.

“Each day, trucks are picking up and delivering cargo shipments at ground handling stations, and during certain periods it is very busy, leading to high congestion and trucking companies having to wait in line.” [5]

Congestion or delay is undesirable and increases the cost for all parties and inconvenience for customers. The process for delivering shipments to the ground handlers should thus be more coordinated. This can be done by the development of a routing tool which incorporated dock scheduling to prevent the situation described above. In this thesis, such a model is researched, developed, and evaluated. The research question is therefore defined as follows.

What is the efficiency improvement of introducing an optimization model which incorporates dock scheduling within the pickup-and-delivery model for the landside air cargo supply chain?

Initially, the model will be set up as a linear model which is solved to optimality. The disadvantage of this approach is that the problem is NP-hard and that the computational time increases drastically with dataset size. For larger datasets, a heuristic model should be developed which produces a solution within reasonable computational time with good solution quality. The solution quality of the heuristic can be compared for smaller datasets with the linear model.

The purpose of this report is to present a literature review on the proposed thesis subject. The most relevant and recent literature on this topic is found. This is done based on the research sub-question which have been established. This literature review is a follow up on the project plan for this thesis. In this project plan, the research sub-questions have been defined, the research methods have been explained and the project planning has been extensively discussed. At the end of this literature review, the sub-questions should be answered and the reader is aware of the state-of-the-art literature. The knowledge gathered in this literature review can be used for the development of the model which is done after the literature review has been completed.

The build-up of this literature review will be as follows. First of all, in [chapter 2](#) the research problem, research objective, and research sub-questions are presented. In addition to that, a problem example is presented and the project planning is briefly described. In [chapter 3](#) the current procedures in the air cargo supply chain are presented together with efficiency improving concepts. In [chapter 4](#) the Key Performance Indicators are presented which can be used to evaluate a solution from the model. After that, [chapter 5](#) presents the linear model of the vehicle routing problem and the dock allocation model. The heuristic approach of the model is presented in [chapter 6](#). In [chapter 7](#) truck dock priority factors are discussed which can be used in the objective function of the model. Finally, in [chapter 8](#) the model extensions are discussed which can be implemented to make the model more realistic and reliable.

2

Research Outline

This chapter presents the research outline of this thesis. In [section 2.1](#) the research problem is introduced for this thesis. In [section 2.2](#) the research objective is presented. After that, in [section 2.3](#) the research question with the corresponding sub-questions are presented. In [section 2.4](#) an example of the problem is given. Finally, a brief overview of the project planning is described in [section 2.5](#).

2.1. Research Problem

The increase in cargo being transported by the cargo department of Schiphol Airport introduces challenges in the operational process of processing incoming trucks efficiently. The arrival of cargo trucks to unload trucks at the ground handlers is currently one of the bottlenecks in this process. Verduijn et al. [4] describe the process of the air cargo supply chain. Verduijn defines the problem as follows.

“Each day, trucks are picking up and delivering cargo shipments at ground handling stations, and during certain periods it is very busy, leading to high congestion and trucking companies having to wait in line” [5]

The situation above causes delay because the number of trucks which arrive at the ground handler is larger than the available number of docks. Because coordination between the freight forwarding companies is not done, ground handlers are not able to schedule docks to specific freight forwarders. In addition to that, most freight forwarders have the tendency to pickup items throughout the day and deliver a full truckload at the end of the day. A peak of arriving trucks is also observed on Friday afternoons. The scarcity of docks has the consequence that trucks have to wait outside the ground handlers, which leads to congestion and delays. This situation is undesirable as it will lead to financial losses for the freight forwarding companies. First of all, the trucks have to wait for a dock, meaning that the trucks cannot be used for other operations. Second, it might lead to a situation in which an item is delivered too late at the ground handlers and misses its intended flight.

Currently, there are models available that are inspired by the Vehicle Routing Problem which model depot and shop nodes to be visited. An extension of this is the Pickup-and-Delivery model which specifies pickup and delivery nodes. There is plenty of research available on this model. The integration of dock capacity is however not commonly practiced. In addition to that, this problem is specifically for freight forwarders and ground handlers which needs specific modifications to the model. To the best of our knowledge, a model that combines the pickup-and-delivery formulation for the integration between ground handlers and freight forwarders with limited dock capacity is not developed yet.

2.2. Research Objective

In the paragraph above the research gap has been introduced. Formally, the research objective of this thesis is defined as follows.

To find out if the delivery process of cargo trucks at ground handlers can be improved by developing a truck routing and dock scheduling tool.

It should be noted that we specifically consider the *delivery* process. In other words, the trucks arriving at the ground handler for export. Importing goods is considered out of scope for this thesis. The problem definition is based on the situation of Amsterdam Schiphol Airport. The data for the model is therefore based on data from this airport. The model should however be set up that it also allows being implemented at other airports.

2.3. Research Question(s)

The research question for this thesis is defined as follows.

What is the efficiency improvement of introducing an optimization model which incorporates dock scheduling within the pickup-and-delivery model for the landside air cargo supply chain?

To answer this research question, a set of sub-questions is developed. In the overview below, the sub-questions are presented with a short explanation. These sub-questions serve as a basis for the following chapters in this literature review. Each chapter aims at answering one of these sub-questions. With these sub-questions, the theoretical basis for an optimization model is presented and the model can be developed in the next stage of the thesis. If the model has been developed, the main research question can be answered.

1. *What is the current procedure in the air cargo delivery process?*

The current procedure in the delivery process should be studied to be familiar with this process. If this is known, the bottlenecks can be identified. Knowing these bottlenecks can help in the formulation of the algorithm and identify possible weak spots.

2. *What indicators are relevant to determine the efficiency of the ground handling process?*

To evaluate the efficiency and quality of a schedule, the key performance indicators need to be known. If these are known and it is clear how they can be formulated, different schedules can be compared to each other in terms of efficiency.

3. *What linear programming methods are suitable for the truck routing combined with the scheduling problem?*

The trucks should be routed from the freight forwarders to the ground handlers. Different linear models should be compared to each other to determine which method is most suitable and efficient to solve this problem. Also, the methods that perform dock scheduling should be studied. Ideally, a model should be found which has the potential of combining the two already. If this is not possible, the interaction between the two models should be studied.

4. *What heuristic methods are suitable for the truck routing combined with the scheduling problem?*

For larger data sets, the computational time to find the solution from a linear programming problem might take very long. To reduce the computational time, the possibilities of using a heuristic should be studied. This sub-question is set up to find out which heuristics can be used to solve the truck routing problem. In addition to that, the dock allocation heuristics should be studied. Ideally, a model that combines the truck routing and dock scheduling should be found. Alternatively, the interaction between the two models should be found.

5. *Which factors influence the priorities to allocate trucks to a dock?*

To allocate trucks to docks, there are several factors that are of influence to determine the priority of the trucks allocated to a dock. This could be for example the type of cargo that the trucks contain, how much cargo is in the truck, and the time that the cargo needs to be delivered at the ground handling station at latest to ensure it is on time for the intended flight. These factors are relevant to set up the objective function of the problem.

6. *What are possible extensions of the model to be implemented later to make the model more complete?*

The incorporation of truck routing and dock scheduling is the basis of the model. The model can however be extended even more to make it more realistic. This sub-question aims at finding possible improvements for the model to incorporate at a later stage if time allows for this.

2.4. Problem Example

In [Figure 2.1](#) a schematic overview of the problem is presented. The three square boxes represent the freight forwarders. Each of the freight forwarders has several shipments to be sent to a specific ground handler. Each shipment has specific properties which are presented below the freight forwarder boxes. The color of the layer of the freight forwarder represents to which ground handler the shipment should be transported (e.g. shipment 1, 2 and 3 need to be transported to ground handler 1). In this specific case, shipments 1, 2 and 4 can

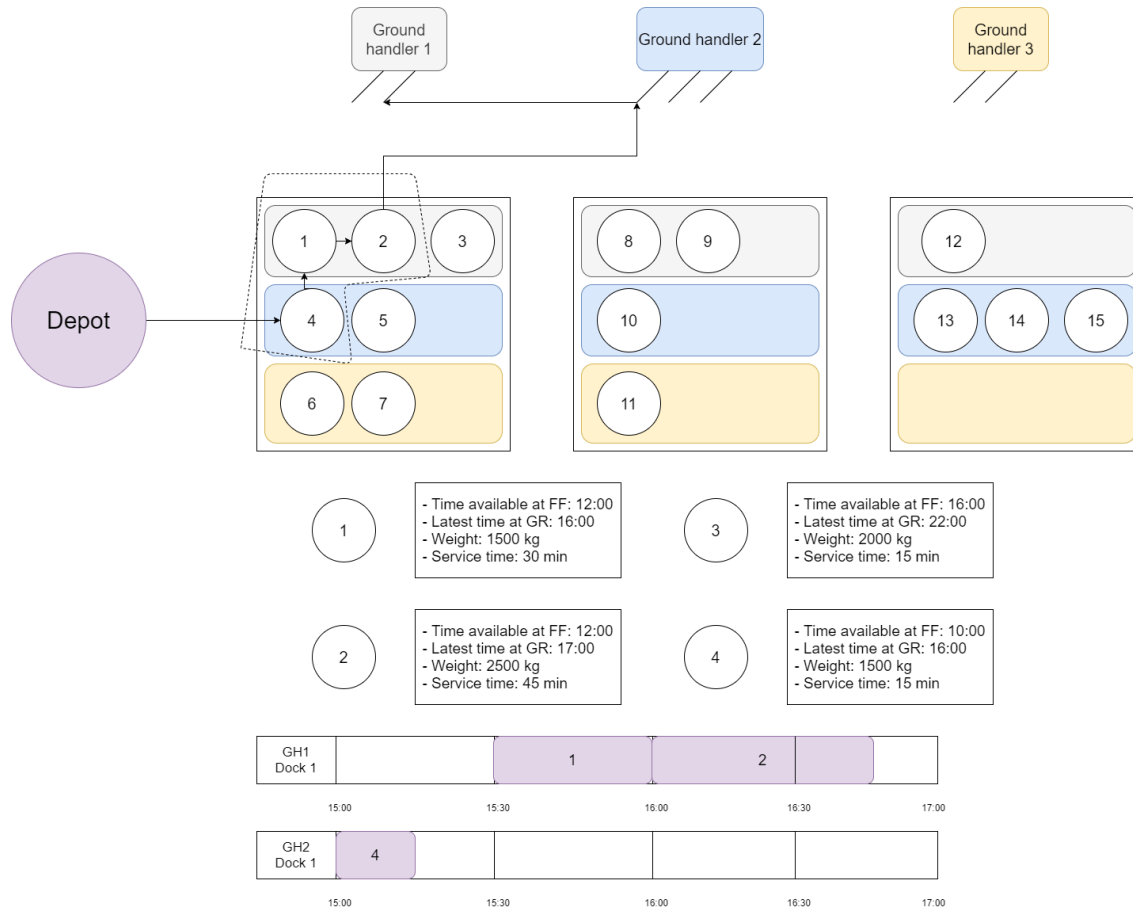


Figure 2.1: Example of problem definition

be picked up and delivered in a single truck (given the capacity of the truck is enough). The time available at the freight forwarder of shipment 3 is at 16:00. It can thus not be picked up by this truck as shipment 1 needs to be at the ground handler by 16:00 already. Finally, the model should also produce a schedule of the docks at the ground handlers. An example is presented at the bottom of [Figure 2.1](#).

2.5. Project Planning

The project planning for this thesis is described in this section. The thesis is split up in different phases. Each phase is separated from another phase by a review point. In total, five phases are identified. An overview of the phases and their respective review points is presented in [Figure 2.2](#).

From this figure it can be seen that the project plan and literature study phase run in parallel. These two phases are also highly interacting with each other. For each of the phases a workflow diagram has been set up. In addition to that, a work breakdown structure was generated. If the literature study phase is finished, the initial phase is started. In this phase, the basic model is developed and tested. In the final phase of the project, the model is extended, if possible, with extensions mentioned in this literature review.

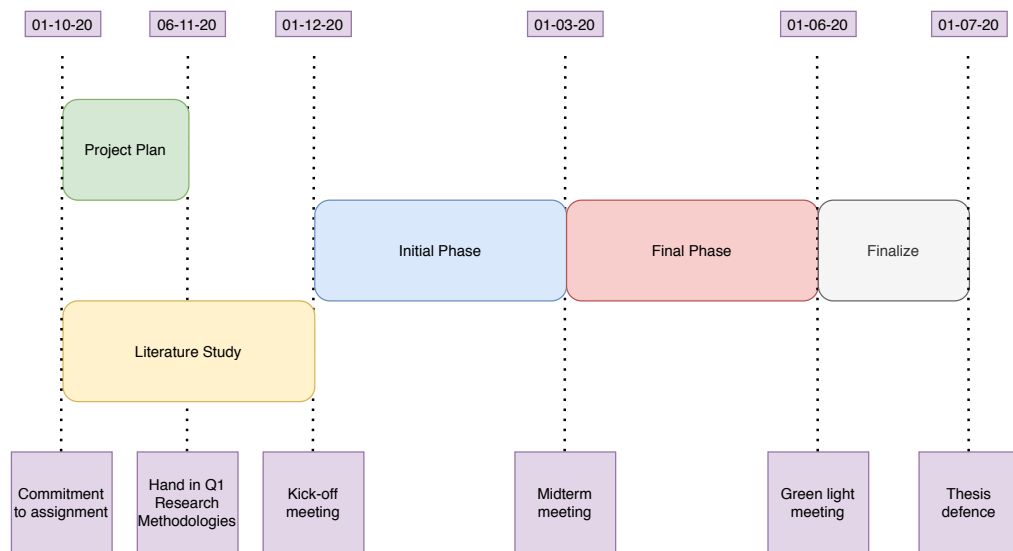


Figure 2.2: Project Phases related to review points

3

Air Cargo Supply Chain

This chapter presents the current situation of the air cargo supply chain. In addition to that, this chapter will present the research which has been executed to increase the efficiency of the current situation. The build-up of this chapter will be as follows. First, in [section 3.1](#) the current procedures in the air cargo supply chain are presented. After that, [section 3.2](#) presents the most recent research efficiency improving concepts.

3.1. Current Procedures in the Air Cargo Supply Chain at Schiphol Airport

The urge to find innovation in the supply chain of air cargo is natural and has always been a topic of research. In December 2017 the 'Smartest Connected Cargo Airport Schiphol' project started [5]. The project aims at optimizing the delivery processes and integrating collaboration between the involved parties in the research. The project is a collaboration between KLM, Schiphol Airport, TU Delft, Cargonaut, Amsterdam University of Applied Sciences and other companies involved in the supply chain. The project was also subsidized by the Dutch Government and had a total budget of 6.2 million euros [6]. The project is concluded by the publication of a report by Verduijn et al. [4] in which the findings are reported. The report provides a detailed explanation of the current procedures related to the ground handling procedure at Schiphol Airport. This is in the range from required documentation for freight handling to the optimal routing of trucks at the airport. For this thesis, the focus will be on the procedures related to the landside pickup and delivery. The documentation process is not considered as this is considered out of scope.

At first, the relevant parties are discussed which will be used in the rest of this literature review. The air cargo supply chain starts with many individual customers. For this example, the customers are based in countries outside The Netherlands and place orders from companies in The Netherlands. The orders have to be shipped via Amsterdam Schiphol Airport as transporting via plane is assumed to be the most efficient method. The companies themselves are often not involved in the sending procedure of their orders to the airport. They outsource this to the freight forwarders of Schiphol Airport. Each freight forwarder is responsible to ship the orders to the airport. On the side of the airport, ground handlers are positioned. These ground handlers receive the shipments from the freight forwarders and prepare them for air travel. After the required paperwork has been completed, the freight forwarders send the shipments to the flight it is intended for. This can be as belly freight for a passenger airline or a dedicated freight aircraft. Each ground handler is dedicated to a specific airline or has special agreements with a specific freight forwarder. The freight forwarders receive orders from multiple companies with multiple destinations. Each freight forwarder thus has to travel to different ground handlers in most cases.

The collaboration between freight forwarders could increase the efficiency of this pickup and delivery process. In Verduijn et al. [4] the current pickup and delivery process is described and three consolidation methods have been identified which are currently implemented at Schiphol Airport. At first, in *dedicated trucking*, one of the freight forwarders is responsible to pickup all shipments and to transport them to the ground handler. The truck can visit multiple freight forwarders to have the highest possible load factor. In some situations, the immediate delivery of the shipments is more efficient and the truck drives with a suboptimal filled truck.

However, in situations in which multiple freight forwarders are not able to fill a full truck on their own to the ground handler, the freight forwarders can cooperate to fill the trucks. This is referred to as *consolida-*

tion between freight forwarders by Verduijn et al. [4]. In this principle, freight forwarders forward their goods to a cross-dock station. At the cross-docking station, the items from different freight forwarders are combined. More information about cross-docking and related literature is presented in the following section. Finally, Verduijn et al. [4] define the third type of collaboration as *consolidation between freight forwarders and ground handlers*. The freight forwarders cooperate with other freight forwarders and ground handlers. One of the freight forwarders determines if it can pickup a load from another freight forwarder that has to travel to the same ground handler. If that is the case, the truck picks up another load from the ground handler.

From the three methods mentioned above, it can be concluded that the freight forwarders have realized that to increase the efficiency of the supply chain process, interaction between the links is needed. The next section will present other possibilities to improve the efficiency of the air cargo supply chain which are studied in literature.

3.2. Relevant Efficiency Improvement Concepts

This section will present methods to increase the efficiency of the air cargo supply chain. This section will present methods which are not only applicable to Amsterdam Schiphol, but airports in general. As mentioned in the section above, some aspects are already implemented at Amsterdam.

In the research published by Verduijn et al. [4], another possibility is proposed to improve the pickup and delivery process at Amsterdam Schiphol Airport. It is described as the *time slot booking* system. Each freight forwarder can book a time slot. During this time slot, the truck of the freight forwarder can unload the shipments at the reserved dock. With this method, it is clear for the freight forwarders at what time they can unload at the ground handler. For that reason, it is expected that the delay will be decreased. There is however a problem if the truck needs to visit more than one ground handler in a single route. The reserved time slots might not be optimal related to each other. The time slot booking system has been introduced at Brussels Airport and showed a potential efficiency increase of 5-10% [7].

Ideally, full collaboration between the freight forwarders and ground handlers should be implemented. This also implies that all parties have to share information and resources. It would imply that there is a *central and neutral control tower* which performs scheduling [4]. This type of collaboration implies *horizontal collaboration* as described by Prakash and Deshmukh [8]. Horizontal collaboration is the collaboration between companies that are operating at the same level in the supply chain. This is in contrast to *vertical collaboration* where the collaboration is only between companies that are operating at different levels in the supply chain. Prakash and Deshmukh [8] concluded that horizontal collaboration can increase the overall efficiency of the supply chain, but individual members can face increased costs. If all members have to benefit from the collaboration, it might thus imply that the profit should be shared.

The implementation of full collaboration to the air cargo supply chain has been studied by Ankersmit, Rezaei and Tavasszy [9]. It is concluded that full horizontal collaboration in the air cargo supply chain could result in reduced transport costs up to 40%. The report from Verduijn et al. [4] includes *A Pickup and Delivery Formulation for Lanside Air Cargo Supply Chain* which is developed by Bombelli and Tavasszy [10]. In this model, full horizontal collaboration is assumed where a central planner is responsible for optimizing the problem. A fleet of neutral trucks can pickup shipments from all freight forwarders and transport these to the ground handlers. The theoretical basis for this model will be explained later as it has similarities to this thesis subject. Bombelli and Tavasszy [10] have concluded that the transportation costs can be reduced by 10% and fleet size by 25% if horizontal collaboration is allowed compared to individual optimization. The method thus seems very promising but is however not widely implemented. Basso et al. [11] found out that there are several practical issues that prevent the full implementation of horizontal collaboration. In total, four categories are identified that contribute to this: Design, Planning and operations, Business/market and Behaviors. Each of these categories can be split up and Basso et al. [11] conclude that there are 16 important practical issues. In section 3.1 the topic of cross-docking has already been touched upon briefly. Cross-docking has been the topic of numerous papers. In cross-docking multiple trucks arrive at an inbound station. The trucks are unloaded into the docking station where the load is distributed to other departing trucks. The purpose of cross-docking is to increase the efficiency by rearranging the shipments from the inbound to the outbound trucks. Boysen, Briskorn and Tschöke [12] consider a model in which a limited number of inbound docks is available. The outbound docks and trucks are not modeled in this research. Each shipment is given a departure time at which it should be at the outbound truck. If this is not possible, it is seen as a missed delivery.

Another innovation in the air cargo supply chain is the *milk run* principle. In the milk run process, a truck visits multiple suppliers in a route. Each of the trucks loads the goods in the truck and this was the truck is efficiently loaded to a facility. Brar and Saini [13] concluded that the milk run process can reduce the number of trucks and travel distance of trucks. The routes for the milk run process can be set up on forehand if the shipments to be picked up are known for all companies. It is thus important that information is shared with all parties involved. The milk run principle has been introduced at Schiphol Airport with various ground handlers and trucking companies since the 1st of May 2015. One year after the introduction of the project, it was shown that up to 40% of truck movements at the ground handlers can be reduced. In addition to that, the emission of carbon dioxide has decreased up to 30% [14].

Concluding Remarks

From this section it can be concluded that there are already several efficiency improving methods implemented at Amsterdam Schiphol Airport. Full horizontal collaboration would be the optimal situation, given that the profit is shared as some individual companies might not benefit from it. However as discussed by Basso et al. [11], there are several practical problems to this method. Most methods which are now implemented aim to reduce the number of trucks which arrive at the ground handlers. This is done by optimizing the loading of the trucks which arrive at the ground handler. The scheduling of trucks to docks and considering that only a limited number of docks are available, is something which is not implemented yet. The time slot booking seems promising to help solve this situation but unfortunately this can lead to situations in which the booked time slots at different ground handlers is not optimal or feasible with respect to the route the truck needs to follow.

4

Key Performance Indicators

To evaluate the quality of a solution, it is important to consider the key performance indicators (KPI's) on forehand. By doing this, it allows to compare different solutions to each other. Moreover, it is of importance to compare the new model to the old, uncoordinated situation. Research on vehicle routing problems or scheduling problems always includes some KPI's to compare their outcome to other models. This section presents an overview of literature on what KPI's have been used. Soysal, Bloemhof-Ruwaard and Bektas [15] have developed a model for the time dependent vehicle routing problem with special focus on the environmental impacts by using a two-row echelon approach. In this study, four KPI's are identified. These are (i) total distance, (ii) total time, (iii) total fuel consumption and (iv) total cost [15]. The research of Soysal et al. [15] is especially relevant and interesting as an overview is presented of relevant other studies on each of the four determined KPI's. The four mentioned KPI's sound natural and will be elaborated on in the following sections how these can be applied to this specific thesis subject. In addition to these four indicators, other relevant indicators are discussed. In [section 4.1](#) the total distance KPI will be discussed. After that, the total time is discussed in [section 4.2](#). The fuel consumption is discussed in [section 4.3](#) which is followed by a discussion on the cost parameter in [section 4.4](#). In [section 4.5](#) the time windows preferences are discussed. Finally, in [section 4.6](#) the benchmark datasets are presented to evaluate the solution.

4.1. Travel Distance

In most vehicle routing problems the travel distance is to be minimized and is a vital part of the problem definition. To simplify a vehicle routing problem, one could argue to combine other performance indicators into the travel distance parameter. This is also suggested by Laporte [16]. Laporte defines a cost matrix (c_{ij}) which represents the cost from node i to j . For simplicity, Laporte states that distances and travel times are equivalent. Incorporating travel distance in the vehicle routing problem sounds natural as it ensures that trucks minimize the total distance which is travelled. The travel distance indicator is not the only relevant parameter for this thesis subject as the observed problem is the waiting time of trucks. It might even be that to solve this, trucks will have to increase their distance travelled to have an optimal solution. The travel distance should however be incorporated in the model and can be used to compare different solution outcomes to see if the dock optimized solution does not increase the travel distance significantly.

4.2. Travel Time

As mentioned in the previous section, the optimization problem can be simplified by incorporating the travel time into the travel distance. This can be done by assuming that the travel distance is directly scaled to the distance travelled, so including the travel time is not needed. If there are certain arcs which however have a very high travel time, but low distance (routes city centers), this does not seem a realistic modelling technique. The objective function can then be changed by including the travel times and allocating a weighting factor of the travel times relative to the distance travelled. Incorporating travel time can go even further as done by Franceschetti, Honhon, Van Woensel, Bektas and Laporte [17]. In the time-dependent vehicle routing problem, the travel time is dependent on the time the route is travelled. Three time windows are identified and the speed of the vehicle is dependent on the time window in which it is assigned. The three phases differ from *congestion*, the *transient* and *free-flow* with increasing truck speed respectively.

Especially interesting would be the feature of including the waiting time of trucks. Since the observed problem at Schiphol Airport is the waiting time for trucks outside the ground handlers at this moment in situations at which the dock capacity is exceeded. If the waiting time could be compared to the situation as it was before incorporating the dock scheduling tool, it is expected that the waiting time of trucks will be decreased.

4.3. Fuel Consumption

The research of Soyal et al. [15] incorporates an extensive environmental impact study. This model incorporates the fuel used per distance unit with a specific speed and load in the truck. The equations which are used in this study are based on the emissions model of Barth, Younglove and Scora [18]. This emissions model is very elaborate and complicated and has many dependencies. The implementation of this feature to the model seems to be interesting to reduce the emission of greenhouse gasses. The aim of this thesis is however not to optimize for fuel-efficiency. For this thesis subject, it could be argued that the fuel consumption is scaled with the distance travelled by the trucks. Combined with the fact that the model of Barth et al. [18] is very complex, this will not be discussed in more detail.

4.4. Financial Cost

Apart from time and distance travelled which have been discussed before, the financial aspect is also of importance for this problem. The shipments should be delivered at the ground handler before a specific time to be on time for that flight. If this time is violated, the shipment is not able to catch the intended flight. Bombelli and Tavasszy [10] have modelled the pickup and delivery problem of the air cargo supply chain and approached this as follows. If a shipment is picked up, a dedicated variable is set to 0. If a shipment is not picked up, this binary variable is equal to 1. In the objective function, among others, this variable is minimized. In addition this approach, it is natural to imagine that each additional truck that is used increases the cost of the solution as well. The cost of each additional truck can also be included in the minimization function.

4.5. Time Window Preference

When working with a vehicle routing problem with time windows model, most models take the time window as a hard constraint (the vehicle routing problem with time windows will be explained in more detail in [chapter 5](#)). The constraint cannot be exceeded and the vehicle must visit the node in the time window. Instead of setting this time constraint as a hard constraint, it can also be implemented as a soft constraint. This is done in the model of Ioannou, Kritikos and Prastacos [19]. In this model, each node is given a preferred time window. As there is no hard constraint for this time window, the violation of the time window is incorporated in the objective function of the model. In case a node is visited before or after the time window, a penalty function is created. The absolute difference in earliness or lateness is multiplied by a scaling factor. Increasing this scaling factors increases the penalty of early or late arrival. If the scaling factors are set to infinity, the soft time window is removed and it becomes a hard constraint. This method can be implemented to this thesis if each of the shipments is given a preferred time window. It can however also be argued that this problem should have hard constraint on the time windows. A shipment cannot be pickup before it is available at the freight forwarder and late arrival to the ground handler will simply result in missing the flight. For that purpose it does not matter if a shipment is late by 5 minutes or 5 hours, it has missed the flight in both cases. The soft time window can however be implemented on the side of the ground handler. The ground handlers only have a limited storage capacity and thus the ground handler could give time at which the truck can arrive at the ground handler, related to its plane departure time. If the truck arrives before this time, the ground handler has to arrange additional storage space and this storage cost are reflected in this soft time window violation penalty. The research of Roca-Riu, Fernández and Estrada [20], which is specifically towards parking slot assignment and will be discussed later, also incorporates the possibilities of soft time windows with penalty functions for violations.

4.6. Benchmark Datasets

For the evaluation of the model, a dataset is needed as input. The dataset can be created in two ways. First of all, a data set can be generated manually. This is especially interesting as it allows to alter the dataset slightly. These modifications can be done smartly to test specific features of the model. Apart from controlling this

dataset manually, the dataset can also be created randomly. This allows to evaluate the performance of the model to different datasets. In addition to that, the size of the dataset can be increased gradually. This allows to check the runtime of the model with respect to the size of the dataset. This method implies that the user of the model creates a dataset by himself to evaluate the model. The advantage of this is that the implementation of the model can be easily checked. The disadvantage of this approach that verifying and validating the result is nearly impossible as there is no comparison to other existing models.

For that reason, the model can also be evaluated using a benchmark dataset from literature. For these datasets, models have found the best or optimal solution. These best-known solutions are stored and can be used for evaluation of the model. In the research of Kachitvichyanukul, Sombuntham and Kunnapapdeelert [21] the benchmark datasets for the general vehicle routing problem are presented. For this thesis subject, the datasets on the Pickup and Delivery Problem with Time Windows (PDPTW) are especially relevant. This dataset partially resembles the actual problem, but does not include the ground handlers, freight forwarders and dock capacity constraint. If these three aspects are relaxed from the model, the dataset can be used to evaluate the solution quality. Kachitvichyanukul et al. [21] only mention the dataset of Li and Lim [22] for the PDPTW. This dataset is also found online with the corresponding best-known solutions via <https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/>. Another dataset that often comes back in literature is the dataset of Solomon [23] from 1987. This dataset is however specific for the Vehicle Routing Problem with Time Windows (VRPTW) instead of the PDPTW. Li and Lim [22] present in their research a way to work around this. For example: two customers are connected to each other where one is the pickup and the other the delivery node. This can be done randomly or based on position or clusters. However, if this would be done randomly, the comparison of model outcomes is not valid anymore as the random behaviour can influence the results significantly. For that reason, the dataset provided by Li and Lim [22] is currently the best benchmark dataset to compare the model to.

Concluding Remarks

From this chapter it can be concluded that there are multiple key performance indicators available for determining the efficiency of a schedule. All vehicle models include the minimization of the distance to be travelled, as this is the theoretical ground of the problem. In some cases the travel time can be scaled to the distance travelled. In that case, separate inclusion of this parameter is not needed. In case this does not reflect reality, the travel time can be added as additional minimization parameter. It is concluded that the fuel consumption is out of scope for this problem and is reflected in the distance travelled. The financial cost is relevant for shipments missing their flights and for financial costs of additional trucks to be used. The time window preference can be implemented in the model as a soft constraint to incorporate the limited storage capacity of the ground handlers. Finally, a benchmark dataset has been introduced by Li and Lim suitable for the pickup and delivery problem with time windows. This dataset can be used to evaluate the simplified linear model and heuristic.

5

Routing and Dock Allocation Model: an Exact Formulation

This chapter will present the linear vehicle routing problems which are currently available in literature. Apart from the linear routing model, the dock scheduling method should also be incorporated in the model. This chapter will also present the linear models for this problem. These two models should then be integrated into each other to have the overall problem optimized. As there is currently no model available that incorporates the two, this chapter will also present methods to do this. The build-up of this chapter is as follows. First of all, in [section 5.1](#) the variants of the vehicle routing problem will be discussed. After that, in [section 5.2](#) the linear vehicle routing formulations are discussed. Finally, in [section 5.3](#) the linear scheduling models for dock allocation will be discussed and how this can be integrated into the vehicle routing problem.

5.1. Vehicle Routing Problem Variants

This section will present the linear models of the truck routing problem. This problem can be seen as a type of the vehicle routing problem (VRP). The vehicle routing problem is introduced first by Dantzig and Ramser in 1959 [24]. In the vehicle routing problem, a set of nodes need to be visited. The objective of the vehicle routing problem is to minimize the total distance which is travelled. Each truck departs from a depot-node and travels to the shop-nodes and returns to the depot-node afterwards. For this specific thesis subject, the freight forwarders can be seen as the depots and the ground handlers are the nodes to be visited. The vehicle routing problem has been the topic for many research papers. The solution methods are linear models as well as heuristics. An overview of research solutions is given by Laporte [16]. This section will present the relevant extensions of the vehicle routing problem.

The basic extension of the vehicle routing problem is the *Capacitated Vehicle Routing Problem*. This model is widely implemented as it includes the capacity of a vehicle on its route. It is thus not possible to load trucks as full as possible, but their capacity is limited. Each node that is visited has specific demand. It is generally assumed that the demand of a node cannot be higher than the capacity of the truck. Therefore, each node is only visited by a truck once. Laporte presents a method on how to incorporate the capacity constraint to the model in [25].

The reality of the model can be increased by adding the feature of shops to present a time window in which the truck can arrive at the node to unload. This is incorporated in the *Vehicle Routing Problem with Time Windows*. The time windows can be applied as hard constraint or soft-constraint. If it is implemented as a hard constraint, the truck has to arrive in the provided time window. If it is implemented as a soft constraint, violation of the time window is allowed. Applying a penalty to time violations can then be implemented. El-Sherbeny [26] presents an exact formulation of the VRPTW.

In the model of El-Sherbeny [26], only a single depot (often referred to as node 0) is available. To extend the model further, the *Multi Depot Vehicle Routing Problem* can be applied. The extension of the model is explained by Wang, Zeyuan, Wei, Ji and Yang [27]. It is important to note that the Multi Depot Vehicle Routing Problem only considers the same type of good from all depot nodes. It optimizes the demand of each shop node based upon travel distance. It does not allow for orders from a ground handler to a specific ground handler.

This specific feature is however essential for the modelling for this specific subject. Each depot (freight handler) has specific orders for the nodes (ground handlers). The shipments for a specific ground handler cannot be supplied by an arbitrary depot. To incorporate this feature, the *Pickup-and-Delivery Problem* is developed. This extension of the general Vehicle Routing Problem model is explained by Rais et al. [28]. In this approach, another set of decision variables is added to the model that consider the specific shipments orders and deliveries.. The y_{ij}^{kr} variable is introduced which is one if vehicle k carries the request r on the arc (i,j) and is zero otherwise. The model allows that more than one vehicle visits a specific node. With this assumption, the model also introduces the possibility of transferring freight from one vehicle to another at a node (taking into consideration the capacity constraint of the vehicle). This means that the model basic which is presented above is slightly changed in the pickup-and-delivery problem. Each node may be visited more than once, only by different vehicles. The new introduced variable, y_{ij}^{kr} brings additional constraints as explained by Rais, Alvelos and Carvalho [28]. As explained before, this model allows to change load from a truck to another at a node. In this specific thesis subject, this is however not desired. This procedure will take time and requires additional equipment and dock capacity. In addition to this, the proposed model by Rais et al. [28] does include time windows which are critical for the to be developed model.

The model developed by Bombelli and Tavasszy [10] resembles the desired situation in more reality. The model represents in detail the Pickup-and-Delivery for freight forwarders to ground handlers. The model incorporates import and export deliveries with a time window. In contrast to the model of Rais et al. [28], import and export blocks are created for each freight forwarder. These important and export blocks are incorporated to visit import an export nodes where deliveries are placed. The model also incorporates time windows which are modelled as a hard constraint. In case a delivery is not picked up, a binary variable is set equal to the one, else it is zero. The binary variable is also incorporated in the objective function of the model to minimize the number of shipments which are not delivered. The model does imply full horizontal collaboration and a neutral fleet which is assigned by a planner. To the best of our knowledge, the capacity of docks is not included in this model.

5.2. Linear Vehicle Routing Problem Formulations

This section presents the different formulations which can be used to express the vehicle routing problem. In the previous section, this is already briefly addressed.

5.2.1. Two-index

The two-index formulation is the most classical representation. In the two-index formulation, x_{ij} is defined which is a binary variable. It is equal to 1 if the route from i to j is travelled and zero otherwise. The model assumes a homogeneous fleet as there are no vehicle specific decision variable. The fleet capacity is introduced by limiting the number of allowable departures from the depot. The number of routes which are allowed to depart from the depot should be less or equal than the number of vehicles. Each node can only be visited once and flow conservation between the nodes has to be ensured.

5.2.2. Three-index

In contrast to the two-index formulation, the three-index formulation allows for a heterogeneous fleet. The decision variable is defined as: x_{ij}^k . This decision variable is binary and has unitary value if node i to j is travelled with k and zero otherwise. The advantage of this is that the vehicle parameters can be adjusted. For example, the vehicle capacity (Q) or vehicle speed (V) can be implemented as variable (Q^k, V^k) depending on the vehicle. Rieck and Zimmermann [29] present a discussion on the implementation of the two- or three-index vehicle routing problem. As mentioned, the three-index is more flexible when a non-homogenous fleet is implemented. It is however shown in the paper that the three-index approach is more time consuming when using larger data instances and thus their model make use of a two-index formulation. An example of the three-index formulation is presented below where the Pickup-and-Delivery Problem with Time Windows formulation (most suitable for the model) is presented. The model below is based on the formulation as

presented by Ropke and Pisinger [30].

$$\text{minimize } \sum_{k \in K} \sum_{i,j \in A} d_{ij} X_{ij}^k \quad (5.1a)$$

subject to

$$\sum_{k \in K} \sum_{j \in N} X_{ij}^k = 1 \quad \forall i \in P, \quad (5.1b)$$

$$\sum_{j \in V} X_{ij}^k - \sum_{j \in V} X_{j,n+i}^k = 0 \quad \forall i \in P \quad \forall k \in K, \quad (5.1c)$$

$$\sum_{j \in P \cup Des} X_{0,j}^k = 1 \quad \forall k \in K, \quad (5.1d)$$

$$\sum_{i \in D \cup O} X_{i,2n+1}^k = 1 \quad \forall k \in K, \quad (5.1e)$$

$$\sum_{i \in V} X_{ij}^k - \sum_{i \in V} X_{ji}^k = 0 \quad \forall j \in N \quad \forall k \in K, \quad (5.1f)$$

$$S_{ik} + t_{ij} + c_i - M(1 - X_{ij}^k) \leq S_{jk} \quad \forall i,j \in A \quad \forall k \in K, \quad (5.1g)$$

$$a_i \leq S_{ik} \leq b_i \quad \forall i \in V \quad \forall k \in K, \quad (5.1h)$$

$$L_{ik} + l_i - M(1 - X_{ij}^k) \leq L_{jk} \quad \forall i,j \in A \quad \forall k \in K, \quad (5.1i)$$

$$L_{ik} \leq C^k \quad \forall i \in V \quad \forall k \in K, \quad (5.1j)$$

$$X_{ij}^k \in \{0, 1\} \quad \forall i,j \in A \quad \forall k \in K, \quad (5.1k)$$

$$S_{ik} \geq 0 \quad \forall i \in V \quad \forall k \in K, \quad (5.1l)$$

$$L_{ik} \geq 0 \quad \forall i \in V \quad \forall k \in K \quad (5.1m)$$

In the provided model, there are n shipments to be delivered. The set P is the set of pickup nodes which is equal to $(1, 2, \dots, n)$. The delivery node set, D is equal to $(1+n, 2+n, \dots, 2n)$. A depot origin and destination depot are also introduced as node 0 and node $2n+1$ respectively. The set N is defined as $D \cup P$. The set V is defined as $N \cup 0 \cup 2n+1$. The graph $G = (V, A)$ consist of all possible arcs between the nodes, $V \times V$. Finally, the set of vehicles is denoted by set K . The arrival time of truck k at node i is presented with decision variable S_{ik} . Each node presents a time window as $[a_i, b_i]$ with the earliest and latest start of service respectively. The c_i variable depicts the service time of a truck at node i . Finally, for all pickup nodes ($i \in P$), a load is assumed, l_i . For the delivery nodes ($i \in D$) the load is assumed to be equal in magnitude but opposite sign, $l_i = -l_{i-n}$. The origin and destination node have a zero demand load and all other pickup nodes have a load larger than zero. The capacity of a truck is denoted as C^k for $k \in K$. The decision variable L_{ik} is introduced which resembles the load of truck k after leaving node i .

The objective function is presented in Equation 5.1a which minimized the total distance travelled. Equation 5.1b ensures that each pickup node is assigned to one truck exactly. In Equation 5.1c the delivery node is also visited if a pickup node is visited by a vehicle. The departure from the depot and arrival at the depot at the end of the route is ensures by Equation 5.1d and Equation 5.1e respectively. In Equation 5.1f the flow conservation over nodes is ensured. Equation 5.1g ensures that S_{ik} is correctly adjusted. In Equation 5.1h the time windows for all nodes are respected. The relation between Equation 5.1g and Equation 5.1h eliminates the possibility of forming subtours. In Equation 5.1i the load of a truck is updated if the node is visited. Finally, Equation 5.1j ensures that the loading of a truck can never exceeds its capacity. It should be noted that all trucks should be used in this formulation. If a truck is not used, it travels from the departure depot (node 0) to the arrival depot (node $2n+1$) immediately.

5.2.3. Set Partitioning

The previous models tried to optimize the arcs to be travelled to come to a optimal route. In contrast to this method, the set partitioning formulation considers all feasible routes and selects the route with minimal cost. Baldacci, Christofides and Mingozzi [31] present an overview of the formulation as below for the simple VRP. In this formulation, R represents the index set of all feasible routes. The coefficient a_{ir} has the unitary value if vertex i belongs to route r and zero otherwise. The cost of each route is represents with c_r . The decision

variable y_r is 1 if route r belongs to the optimal solution and zero otherwise.

$$\text{minimize } \sum_{r \in R} c_r y_r \quad (5.2a)$$

subject to

$$\sum_{r \in R} a_{ir} y_r = 1 \quad \forall i \in V, \quad (5.2b)$$

$$\sum_{r \in R} y_r = K, \quad (5.2c)$$

$$y_r \in \{0, 1\} \quad \forall r \in R \quad (5.2d)$$

In Equation 5.2a the objective function aims at minimizing the total cost of the routes. Equation 5.2b ensures that each node is covered by one route. Finally, the number of routes is limited to the number of vehicles by Equation 5.2c.

5.2.4. Commodity Flow

In contrast to the three methods which are mentioned above, the model can also be programmed in terms of flow over nodes. This method is called the commodity flow formulation. A formulation of this is presented by Baldacci, Hadjiconstantinou and Mingozzi [32]. Below, an overview of the algorithm is presented. In this two-commodity flow formulation, the decision variable x_{ij} represents the commodity flow from node i to node j . The decision variable x_{ji} is introduced which represents the empty space on a vehicle for that edge. The variable ξ_{ij} is introduced which has the unitary value if edge (i,j) belongs to the optimal solution.

$$\text{minimize } \sum_{i,j \in A} d_{ij} \xi_{ij} \quad (5.3a)$$

subject to

$$\sum_{j \in N} (X_{ji} - X_{ij}) = 2l_i \quad \forall i \in N, \quad (5.3b)$$

$$\sum_{j \in N} X_{0j} = q(N), \quad (5.3c)$$

$$\sum_{j \in N} X_{j0} = K \cdot C - q(N), \quad (5.3d)$$

$$\sum_{j \in N} X_{n+1,j} = K \cdot C, \quad (5.3e)$$

$$X_{ij} + X_{ji} = C \cdot \xi_{ij} \quad \forall i,j \in A, \quad (5.3f)$$

$$\sum_{\substack{j \in V \\ i < j}} \xi_{ij} + \sum_{\substack{j \in V \\ i > j}} \xi_{ji} = 2 \quad \forall i \in N, \quad (5.3g)$$

$$X_{ij}, X_{ji} \geq 0 \quad \forall i,j \in A, \quad (5.3h)$$

$$\xi_{ij} \in \{0, 1\} \quad \forall i,j \in A \quad (5.3i)$$

In this formulation, Equation 5.3a is the objective function which minimized the total cost of a route. In Equation 5.3b ensures that the inflow minus outflow has to be equal to twice the demand at that node. The total load which is carried at the start of a route is equal to the sum of demand that route, represented by $q(N)$ and ensured by Equation 5.3c. The capacity left over in the vehicle from the departure node is represented with Equation 5.3d and is the capacity minus the total demand of a route. The capacity which is left at the arrival node of the vehicle is equal to the capacity of the vehicle and is ensured by Equation 5.3e. The sum of the load on an edge and the empty space should be equal to the capacity of the vehicle and is represented by Equation 5.3f. Finally, Equation 5.3g ensures the route flow conservation.

Before starting with the development of the model, the formulation which seems most promising should be chosen and used for the model. Most research paper use the two- or three-index formulation, based on heterogeneous or homogeneous fleet respectively. One of the extensions will be to introduce a heterogeneous fleet, as will be explained in chapter 8. The three-index method also seems to be the method which is intu-

itively best understandable. The implementation of the dock capacity also seems to favour the three-index notation.

The challenge in the to be developed model is to include the dock capacity at the ground handler and produce a schedule from this. The problem is that the proposed models all separate the routing from the time variables. This is most likely done since the time variables are programmed as continuous variables. It is thus not simply possible to take the sum over all vehicles for all time points and set this sum equal to be less or equal than the capacity of a ground handler. One could argue that the time can be discretized. This would however introduce many more decision variables when working with a data set that spends a large period of time. This would result in a very high computational time for relatively small data instances. The next section presents possibilities to perform the dock scheduling in an efficient way.

5.3. Linear Dock Allocation Model

This section presents the linear dock allocation and assignment models. In particular, it is important to integrate this model into the linear vehicle routing problem model. It is possible to produce the dock schedules based upon request from the freight forwarders and afterwards perform the vehicle routing procedure. This method however introduces inefficiencies if a truck travels to more than one ground handler. In addition to that, two individual optimization methods are performed instead of combining the two into a comprehensive model. This section will thus also present how the dock allocation models can be incorporated in the routing problem.

The problem of assigning trucks to a parking spot has been the topic of research for quite some time. Not only in the aviation industry this problem is well known, also in urban regions the population density is increasing and leads to situation in which regulation of parking spots is needed. The research of Roca-Riu et al. [20] focuses on parking spot assignment in city centers at warehouses. This model is based on the principle of the vehicle routing problem with time windows. All loading/unloading operations can give a preference time window and the model initially considers these time windows as a hard constraint. The nodes of the model are the (un)loading request and there is a dummy node which represents the depot but does not have a physical meaning. Each route starts at the depot and the model determines the optimal route and ends at the depot. The sequence of the routes is the order over which the (un)loading operations are performed. Each route represents one parking spot. The number of routes which are created can thus not be higher than the total number of parking spots. The time constraint is taken as a hard constraint, meaning that if the number of trucks arriving is larger than the number of available parking spots, there is no feasible solution. The paper proposes a method to check for a dataset if it is feasible or not, without trying to solve the problem itself. In addition to that, Roca-Riu et al. [20] propose that the time window constraint can also be implemented as a soft constraint. To ensure that the solution converts to the implemented time window, the absolute difference between the actual time the truck is assigned to the parking spot and the desired time, can be minimized in the objective function. In the model of Boysen et al. [12] the vehicle routing method is also used to assign docks to trucks in a cross-dock station.

As mentioned in chapter 3, cross-docking has been introduced to increase the efficiency of the loading of trucks. In the research of Miao, Lim and Ma [33], a limited number of docks is available at the cross-docking station. Miao et al. [33] specifically state that the data-set should be over-constrained (more trucks arrive than the total number of docks). Instead of modelling it as a vehicle routing problem with time-windows, an alternative method is proposed. First of all, the binary variable x_{ij} is introduced. If the departure time of truck i is smaller than the arrival time of truck j , the variable is equal to one, else it is zero. The arrival and departure time of the trucks is not implemented as a decision variable, but is assumed to be known. With this knowledge, the x_{ij} matrix can be set up. In addition to this, two decision variables are defined. First of all, y_{ik} is defined to be one if truck i is assigned to dock k and zero otherwise. Secondary, z_{ijkl} is introduced which is equal to one if truck i is assigned to dock k and truck j is assigned to truck l , and zero otherwise. The model identifies several relations between y_{ik} and z_{ijkl} to ensure these are working according to their definition. The model then incorporates the constraint, presented in Equation 5.4, which ensures that never two trucks are assigned to the same dock.

$$x_{i,j} + x_{j,i} \geq z_{ijkk} \quad (5.4)$$

If truck i and j are both assigned to dock k , z_{ijkk} is equal to one. This means that x_{ij} and/or x_{ji} needs to be one as well to satisfy the constraint. If x_{ij} is equal to one, this implies that the departure time of truck i is before the arrival time of truck j . This will intuitively not violate the constraint. Otherwise, x_{ji} needs

to be equal to one, meaning that the arrival time of truck i is after the departure of truck j (so truck j is the predecessor of truck i). The disadvantage of this approach is that the arrival and departure time need to be known in order to set up x_{ij} and apply this technique. In addition to that, there will be many decision variables as z_{ijkl} has four terms in its subscript.

The method which has been found to be most similar to the assignment of this thesis subject is the paper of Rieck and Zimmermann [29]. In this research, the vehicle routing problem is combined with a limited number of docks and a schedule for docking is produced. The method splits the problem in two. This consist of on one hand the vehicle routing problem which schedules the vehicles to the required nodes. On the other hand, the model performs the docking schedule. This is also done as a vehicle routing problem. The first step is do define the graph G . This consist of nodes C (customers), nodes L (loading bays), nodes S (start) and nodes E (end). Nodes S en E represent the start and end depot nodes where loading and unloading happens. The allocation of docks is done in a similar method as in the parking spot assignment by Roca-Riu et al. [20] as it is a vehicle routing problem, but the underlying method is slightly different. First of all, the model introduces the decision variable y_{ij} . The conditions for this variable are presented in Figure 5.1.

$$y_{ij} := \begin{cases} 1, & \text{if} \\ & \text{case 1: (un-)loading activities at node } i \in S \cup E \text{ are performed} \\ & \quad \text{before (un-)loading activities at node } j \in S \cup E \text{ at the} \\ & \quad \text{same loading bay or} \\ & \text{case 2: (un-)loading activities at node } j \in S \cup E \text{ are carried out at} \\ & \quad \text{loading bay } i \in L \text{ or} \\ & \text{case 3: after node } i \in S \cup E \text{ all (un-)loading activities are} \\ & \quad \text{performed at loading bay } j \in L \text{ or} \\ & \text{case 4: there are no activities at loading bay } i \in L \text{ and we} \\ & \quad \text{continue with } j \in L \\ 0, & \text{otherwise.} \end{cases}$$

Figure 5.1: Explanation of the decision variable y_{ij} [29]

Each node i from of S and E needs to be visited exactly once by a route. If arc $(i,j) \in L$ is included in a route, this means that $i \in L$ is not used in that day (see case 4). The model is set up such that each route again presets a sequence of nodes to visit. The main difference in this approach compared to the provided problem for this thesis is the location where the dock capacity needs to be limited. In this paper, the docking capacity is limited at the loading and unloading depots (freight forwarders). However, this research aims at limiting the number of docks at the nodes to be visited (ground handlers). This approach is however interesting and seems to have potential when incorporating the capacity constraint in the model. In addition to that, this model assumes that the start and end depot node are at the same location where the number of docks is constant. Each route is thus visiting these start and end nodes. However, in the problem which is discussed in this thesis, different ground handlers, with different dock capacities are considered. This means that for each ground handler, this vehicle routing approach should be done with the nodes which are present at that ground handler. A subset of nodes for each ground handler is thus required to perform this type of implementation.

In the aviation industry, a common assignment problem is the gate assignment problem. This is already adresssed in the research of Mangoubi and Mathaisel [34] in 1985. In this research, aircraft are assigned to gates where the passengers walking distance is minimized. The constraints enforce each flight to be assigned to one and only one gate. To prevent two flights from being assigned to the same flight, a subset is developed with flights which may be violating with their arrival and departure times. The gate assignment problem has also been researched by Xu and Bailey [35]. In contrast to the research of Manguobi and Mathaisel [34], an extra set of decision variables is introduced. In this model, z_{ijk} is introduced which is equal to one if flight i and j are assigned to gate k and flight i immediately precedes flight j and zero otherwise. Each flight is assigned a to a gate and then two additional constraints are introduced which ensure that each flight can be followed by at most one flight and at most one flight can be the predecessor of it. With these combination of constraints, the gate capacity can never be exceeded. The main problem with this approach is that the arrival times of the aircraft are known and only the assignment to gates has to be done. The model of Xu and Bailey [35] should however be studied as it seems promising to implement this approach to this suggested thesis problem.

The linear programming language which is set up above needs to be solved with a linear programming solver package. Each solver has specific features and performances on data sets. In the research of Anand, Aggarwal and Kumar [36] the most used solvers are studied and their performance. The paper discusses the well known (commercial) solvers Xpress, CPLEX and GuRoBi. Anand et al. [36] conclude that CPLEX and GuRoBi perform better in real life applications and Xpress outperforms the other two in complex problems. It is therefore difficult to determine right now what solver would be the most efficient as the exact problem formulation is not determined. This evaluation should be done again when implementing the model.

Concluding Remarks

From this section it can be concluded that the research in the vehicle routing problem and dock assignment has been the topic of numerous research projects. The combination between these two is to the best of our knowledge however not implemented in the context of the interaction between the freight forwarders and the ground handlers. In this chapter, initially, the extensions of the vehicle routing problem has been discussed. It is concluded that the pickup-and-delivery problem with time windows is the best suitable model to implement the vehicle routing problem. After this, the vehicle routing formulations are identified. The distinction is made between the two-index, three-index, set partitioning and commodity flow formulation. It is concluded that the three-index formulation is probably most suitable for this thesis subject. In the third part of this research, multiple ways of incorporating the linear dock allocation model have been discussed. Two of these methods implement this as a vehicle routing problem where the sequence of visiting the nodes can be seen as the order over which the trucks can dock at the ground handler. Alternative methods are also presented such as the cross-docking capacity and gate assignment problem so ensure that two trucks cannot occupy the same dock at a time.

6

Routing and Dock Allocation Model: an Heuristic Formulation

The models which are discussed in the [chapter 5](#) can be solved with a (commercial) solver. The problem is NP-hard and the time it takes to find the optimal solution increases drastically with dataset size. For example, the comprehensive model of Bombelli and Tavasszy [10] has two stopping criteria to the model. First of all, the gap optimality is less or equal to 5% and second a maximal computational time of 12 hours. The twelve hours is not a realistic time span if the model would be implemented in reality. The model should be able to run in a limited time span, especially for slight changes in the dataset. This chapter will therefore present an overview of heuristic methods on the vehicle routing problem. In [section 6.1](#) the heuristics of the vehicle routing problem will be presented. After that, the heuristics of the dock scheduling model will be presented in [section 6.2](#). This section will also present possible integration of the model to the vehicle routing problem.

6.1. Heuristics of the Vehicle Routing Problem

The development of heuristics for vehicle routing problems has been the topic of research for many years and in a dedicated chapter, written by Laporte, Ropke and Vidal [37], an overview is given of the developments in this field is presented. The heuristics are divided into the *Construction Heuristics*, the *Classical Improvement Heuristics* and *Metaheuristics*. In Construction Heuristics, the goal is to develop a solution which can be improved in later phases. The improvement of solutions is done in the classical improvement heuristics. These heuristics are in most cases specific and can solve a specific problem. The development of metaheuristics helped in this and these metaheuristics provide a structure for solving different problems. With this in mind, this chapter will specifically focus on the metaheuristics which are found in literature on the vehicle routing problem.

6.1.1. Initial Solution

Finding the initial solution is in most cases done with a simple heuristic. Ropke and Pisinger [30] simply assume that the initial solution is found before the LNS heuristic is applied and is found by a simple construction heuristic. Pisinger and Ropke [38] use a regret-2 heuristic to get a solution. Laporte et al. [37] mention that the *Clarke and Wright* is a simple heuristic which is often used to generate an initial solution. Altnel and Öncan [39] explained this method in more detail. Initially, each request is given an own vehicle. One can imagine that this is a highly undesired solution because the number of trucks will be equal to the number of requests. Routes are represented as $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$. It is then tried to insert the request in other routes. The representation of the route changes into $(0, \dots, i, j, \dots, 0)$. The saving from this route is significant and can be expressed as: $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. For each iteration, the highest possible saving in route cost is accepted and this solution is accepted. If it is not possible to perform a feasible merge, the algorithm is stopped.

6.1.2. (Adaptive) Large Neighborhood Search

The research of Ropke and Pisinger [30] introduces the Adaptive Large Neighborhood Search to the Pickup-and-Delivery Problem with Time Windows. The Adaptive Large Neighborhood Search is a variant on the Large

Neighborhood Search (LNS) which has been introduced by Shaw [40]. In Figure 6.1 an overview is given of the LNS.

ALGORITHM 1: LNS HEURISTIC.

```

1  Function LNS( $s \in \{\text{solutions}\}, q \in \mathbb{N}$ )
2    solution  $s_{best} = s$ ;
3    repeat
4       $s' = s$ ;
5      remove  $q$  requests from  $s'$ 
6      reinsert removed requests into  $s'$ ;
7      if ( $f(s') < f(s_{best})$ ) then
8         $s_{best} = s'$ ;
9      if accept( $s', s$ ) then
10        $s = s'$ ;
11    until stop-criterion met
12  return  $s_{best}$ ;

```

Figure 6.1: Large Neighborhood Heuristic Pseudo Code [30]

The heuristic starts with an initial solution, as explained in the previous section. The initial solution for the pickup-and-delivery problem consists of a set of routes and each route has a set of requests allocated. In case a request is not allocated to a route, it is put into the *request bank*. The requests in the request bank are not served by any route, so a high penalty is given to these routes. In line 5 of Figure 6.1, q request are removed from the current solution s and placed in the request bank. In line 6, the requests from the request bank are placed back into one of the routes. If the cost of the new solution is better than the best solution, the new solution is accepted. The removal and insertion of the request as described in line 5 and 6 is done with special heuristics. These will be considered later in this section. The most basic accepting procedure is to accept any solution which is better than the current solution. This might however lead to situations in which the searched neighborhood is locally optimized, but not globally. The possibility to prevent this situation is explained later in subsection 6.1.3. The number of request which is removed, q , is of impact on the solution outcome. If q is very small, the computational time will be very fast but the solution neighborhood is not explored very well. If q would be equal to the total number of requests, the solution is totally destroyed and build up from scratch. Choosing q wisely is thus of importance for the performance of the heuristic.

Removal Heuristics

The removal heuristics apply to line 5 in Figure 6.1. The research of Ropke and Pisinger [30] provides three methods to remove q request from the existing routes. First of all, the *Shaw Removal Heuristic* which is based on the research of Shaw [40]. The aim of this removal heuristic is to remove requests which are similar to each other. If requests are removed which are very different from each other, it can be imagined that reinserting these requests in the next phase of the algorithm will most likely place them back in their original position. If this happens, the solution space of the problem is not explored optimally. The similarity of two request therefore needs to be defined in quantitative measures. The similarity between requests i and j is called the *Relatedness Measure*, $R(i,j)$. The variables influencing the $R(i,j)$ are the geographical location of the two requests, time windows and number of commodities etc.

The *Random Removal Heuristic* does not consider the relatedness of two requests to each in the removal procedure. This heuristic randomly picks q requests which are removed from the routes. In the shaw removal heuristic, a deterministic parameter is introduced to insert randomness to the selection process. This is not done in the random removal heuristic. The random removal heuristic can be seen as a simplified Shaw removal heuristic where the random parameter is equal to 1.

In addition to the Shaw and Random Removal heuristic, the *Worse Removal Heuristic* computes the cost of a specific request. The cost of a route s is computed with and without request i . Note that if i is removed, the costs of putting it into the request bank are not considered, as this will in most cases always outweigh the benefit from removing i from s . The cost(i, s) is computed as in Equation 6.1.

$$\text{cost}(i, s) = f(s) - f_{-i}(s) \quad (6.1)$$

Where $f_{-i}(s)$ is the cost of solution s without request i . It is desired to have the routes with highest cost(i, s) to be removed from that route, and potentially improving the solution if this q is implemented in another route s . Again, a randomization parameter is added to determine which request is removed. If this is not done properly, it might be that the same (expensive) request gets removed from routes all the time.

Insertion Heuristics

From the previous section, a number of q requests are now placed in the request bank which need to be reinserted in the routes. This section will present the insertion heuristics as explained by Ropke and Pisinger [30]. First of all, the *Basic Greedy Heuristic* is proposed. In this heuristic, the (minimal) cost of placing request i in route s is computed. For each i , the best position in a route s is thus computed. This cost is referred to as $\Delta f_{i,s}$. If request i cannot be placed in route s , $\Delta f_{i,s}$ is equal to ∞ . From these value, it is possible to determine, c_i which is defined to be the $\min_{s \in S} (\Delta f_{i,s})$. In other words, the best position to place i in a route s . We have to define c_i for all request in the request bank. The request with the lowest cost are then placed in route s and the procedure starts over again until all requests are implemented. The disadvantage of this approach is that each request only takes into consideration its own best preference and that future requests are not considered.

To prevent this situation, the *Regret Heuristic* tries to incorporate a look ahead feature. This procedure is also explained by another research of Pisinger and Ropke [38]. In that research, Δf_i^k represents the change in objective function if request i is inserted to the best position of the k 'th cheapest route. If we perform a regret-2 heuristic, the difference between $\Delta f_i^2 - \Delta f_i^1$ is maximized. In other words, if the difference in costs of inserting the second choice compared to the first choice is really high, we have to insert this request i at its first priority. This ensures that the insertion is also looking forward when inserting the requests. The regret heuristic can be extended to a regret- q heuristic.

In the Large Neighborhood Search, a decision is made which removal and insertion heuristic is used throughout the algorithm. To extend this, one could use combinations of the proposed removal and insertion heuristics. This is referred to as the Adaptive Large Neighborhood Search. In each iteration, the decision is made which removal and insertion heuristic will be used. This decision is based upon successes from the individual heuristics themselves. Ropke and Pisinger [30], [38] developed a method to define the probability of selecting a heuristic method which is called the *Roulette Wheel Selection*. If there are k heuristic types, with corresponding weight, $w_i, i \in \{1, 2, 3, \dots, k\}$, the probability of selecting heuristic j is calculated with Equation 6.2.

$$\sigma_j = \frac{w_j}{\sum_{i=1}^k w_i} \quad (6.2)$$

In their research, Ropke and Pisinger present a method to update w_j during the iteration. If a heuristic (combination of removal and insertion) performs well, a score is given to this combination. After 100 iterations (referred to as a segment), the scores of all heuristics are compared. After that, w_{ij} is updated as in Equation 6.3 which represents the weight of heuristic i in segment j .

$$w_{i,j+1} = w_{ij} (1 - r) + r \frac{\pi_i}{\theta_i} \quad (6.3)$$

In this equation, r represents the *reaction factor*. This factor reflects how fast the heuristic changes if a particular heuristic has success. The π_i is the score of heuristic i in the last segment and θ_i is the number of times that heuristic i has been performed in the last segment.

6.1.3. Simulated Annealing

In Line 7 of Figure 6.1 the acceptance criterium of the LNS is very straight forward. If the solution after inserting the requests is better than the best solution known up to that moment, the new solution is accepted. Ropke and Pisinger [30], [38] and Laporte [37] established that this might lead to a local optimum instead of the desired global optimum. To explore a larger solution space, the simulated annealing method can be

implemented to accept a new solution. To accept the solution which is worse than the current solution, Equation 6.4 is used.

$$p_s = e^{-\frac{f(s') - f(s)}{T}} \quad (6.4)$$

In this equation, $f(s')$ is the new solution to be accepted or not, $f(s)$ is the current solution and p_t is the probability that the new solution is accepted. The T parameter is the *temperature* parameter. This parameter has to be larger than one and is decreasing over time. As the parameter decreases over time, the probability of accepting solutions which are worse decreases over time.

In the book of Hillier and Lieberman [41] it is proposed to use the following temperature schedule. $T_1 = 0.2Z_c$, where Z_c is the objective function value for the initial solution. $T_2 = 0.5T_1$, $T_3 = 0.5T_2$, $T_4 = 0.5T_3$ and $T_5 = 0.5T_4$.

6.1.4. Tabu Search

With the method mentioned above, the requests which are removed from a route are either random or selected based upon their cost. In order to explore the solution space wider, the method of *Tabu Search* can be applied. In the research of Cordeau et al. [42], the tabu search is applied to the vehicle routing problem with time windows. The research incorporated also vehicle load and duration constraints, as well as time windows. The principle of the tabu search is to implement a feature which remembers the last steps taken by the algorithm. These steps are referred to as an operator by Cordeau et al. [42]. For example, if (i,k) represents that request i is served by vehicle k . From previous sections we concluded that we could select a heuristic to remove (i,k) and implemented it, so that we have (i,k') , where $k' \neq k$. In the tabu search, it is now forbidden to reinsert i to k for a number of iterations. The operator (i,k) is placed in a tabu-list.

Instead of the vehicle routing problem, the pickup-and-delivery problem is more suitable for this specific thesis research. The implementation of the tabu search algorithm to the pickup-and-delivery problem is done by Nanry and Wesley Barnes [43]. It is also suggested to incorporate the possibility of including non-feasible solutions. The violation of constraints has to be taken into the objective function in order to steer the final outcome into the direction of a feasible solution. This is also suggested by Laporte [37] and Cordeau et al. [42]. A penalty function can be set up as Equation 6.5.

$$f'(x) = f(x) + \sum_k \alpha_k V_k(x) \quad (6.5)$$

Where $V_k(x)$ is the violation of type k in solution x and α_k is the weighting parameter to this constraint which is initially set equal to 1. For each iteration it is checked if x is feasible for k . If this is the case, α_k is divided by $1 + \delta$ and otherwise the weighting parameter is multiplied by this factor. The value of δ should be larger than 0 to force the solution to find a feasible solution.

6.1.5. Genetic Algorithm

In this section the genetic algorithm will be discussed. This method is based on the principle of evolution and the survival of the fittest individuals. Widely implementation of this method to the Pickup and Delivery problem with Time Windows is not done as the solution representation can be very difficult as explained by Pankratz [44]. In that research, the PDPTW is solved for the first time using the GA method. The solution representation for this method is often done using a binary string. The method starts with an initial set of parent solutions. From these solutions, combinations are made which form child solutions which is called the crossover process. The crossover process can be done with different operators to have a variety in children solutions. In addition to crossover processes, mutations can be applied to a solution. Just as in DNA, a mutation is the sudden change of a genetic property, or in the case of the VRP, a change in schedule. This unexpected change in schedule can improve the solution or explore the solution space more widely. Pankratz [44] concludes that the GA can produce solution outcomes which have a good solution quality compared to benchmark datasets. In another research, Baker and Ayechev [45] apply the GA principle to the vehicle routing problem. In their research, it is concluded that applying GA performs less compared to tabu search when the solution quality is considered. It is concluded that GA is more applicable to explore the solution space of the problem.

For this specific thesis subject, the application of GA seems to be quite complex. The Pickup and Delivery Problem with Time Windows on itself would already be difficult to represent in a binary string. The addition of the dock scheduling model and possible extensions seem to complicate the problem even further.

6.1.6. Dynamic Programming

The application of dynamic programming to the vehicle routing problem is only limited. Psaraftis [46] applied this method to the dial-a-ride vehicle routing problem but only solved it efficiently for a limited number of nodes. In the research of Mahmoudi and Zhou [47] the dynamic programming method is applied to the pickup-and-delivery problem with time windows. The method used is based on a state-space-time method and the method is based on time stages. To reduce the complexity of the problem, a Lagrangian operator is introduced. It is shown that the algorithm works with a high number of nodes, but the number of customers is only limited.

6.2. Heuristics of the Dock Allocation Model

In the previous section, the heuristics which respect to the vehicle routing problem are presented. In this section the dock allocation heuristics will be presented. Applying a heuristic is often desired as the computational time needed to optimally solve a linear programming can be very high.

The parking spot assignment of Roca-Riu et al. [20] developed a simple heuristic for their problem. The heuristic first orders the parking spot requests in increasing order. Ties are broken by also considering the latest arrival time and duration. The model then tries to construct schedules by inserting the request over the sorted orders. If a request cannot be inserted due to time violations, this request remains in the 'unassigned' list and it is tried to insert this request in the next parking space. Finally, after the schedules for all parking spaces have been developed, there still may be some 'unassigned' request. These are then implemented in the model at the start or end of the sequence where the violation of the preference time window is the least. The heuristic is compared to the linear model and in most cases the linear model performs better than the heuristic. It is shown that in a specific data set the heuristic outperforms the linear model in the number of feasible solutions.

The research of Miao et al. [33] incorporate the linear model to the truck scheduling of the dock allocation. Especially for larger data sets, the development of a heuristic is desired. In this research, a Tabu Search method and Genetic Algorithm are implemented separately. In the Tabu Search model, the solution is represented as a list where each truck is assigned to a dock. As expressed here: $(s_1, s_2, s_3, \dots, s_n)$ where truck 1 is assigned to dock s_1 and truck n to dock s_n . The order of assigned docks is changed by *The Insert Move* where a single truck is assigned to another dock than it is currently assigned to. Alternatively, *The Interval Exchange Move* exchanges two truck intervals in the new assignment. A truck interval is defined as a group of consecutive trucks which are assigned to the same dock. In the Genetic Algorithm two parents are compared to each other and two cross over operations are included. From this research it is concluded that the heuristic outperforms the linear model for medium and large data sets. It is also concluded that the Tabu Search method outperforms the Genetic Algorithm in terms of computational time and solution quality.

The gate allocation problem developed by Xu and Bailey [35] proposes a Tabu Search heuristic to solve larger data sets. The operations to search the neighborhood space are very similar to the Tabu Search method of Miao et al. [33]. First of all, a *Insert Move* is proposed in which the gate to which an aircraft is scheduled is changed. The second operation is the *Exchange I*, where the gate assignment of two aircraft is simply swapped around. In the *Exchange II*, not two single flights are swapped, but flight pair assigned to a dock is swapped with another dock. The paper presents a detailed method to compute the cost of each operations, instead of computing the solution of the entire new solution. This method seems to be potential as it will decrease the computational time. The number of iterations at which a specific move is set tabu is determined uniformly within a specified interval. The tabu iterations for each of the iterations can be different. In this paper, a very clear explanation of the Tabu Search algorithm with pseudo-code is presented. In this paper it is shown that the Tabu Search finds the same results for all implemented data sets as the linear model, while the computational time is significantly reduced.

The literature on the dock assignment on its own is not very useful since it needs to be combined with the vehicle routing problem. The research above however gives an impression on what methods can be used if the problem would be split up in two and the dock assignment is done separately from the routing process. In the research of Wang et al. [48] the cross-docking process is studied in which the number of docks is limited. In this research, a method is presented in which simulated annealing or tabu search has been introduced. In the construction phase of this model, first the orders are allocated to a cross dock. After that, vehicles are allocated to the cross docks. The third step is to determine the route of the trucks and after that the arrival times are determined. The scheduling of the initial phase is thus done starting from the orders by scheduling

them to a dock. To improve the solution of the construction heuristic, a two stage Tabu Search or two stage simulated annealing method is applied. It is concluded that the simulated annealing method performs better on medium sized instances. For larger data sets, the tabu search obtains better solutions.

Concluding Remarks

From this section is it concluded that applying heuristics to the pickup-and-delivery problem can have a significant impact on the computational time of the problem. The (adaptive) large neighbourhood search, tabu search, genetic algorithm and dynamic programming methods have been addressed. An accepting criteria for the solution such as done in simulated annealing is also discussed. The most frequently used heuristics in this field are the adaptive large neighborhood and tabu search which are often combined with simulated annealing. The simulated annealing procedure prevents solution from converging to a local minimum but tries to find the global optimum.

7

Truck Docking Priority Factors

This chapter is directly related to the fifth sub question and aims at providing an overview of the truck docking priority factors. These factors can be part of the constraints of the problem. Alternatively, these factors can also be related to the objective function of the problem. These truck docking factors are mostly relevant for a data set for which finding a feasible solution is difficult or not possible. A feasible solution is seen as a solution which respects the time window and dock capacity constraint and delivers all shipments. So if there are not enough docks available to serve all shipments, the priority to specific goods should be given. The build up of this chapter will be as follows. First of all, in [section 7.1](#) the minimization of the waiting time is presented. After that, in [section 7.2](#) the different type of commodities to be transported are discussed. In [section 7.3](#) the truck load factor influence is discussed. At last, multi-objective optimization is discussed in [section 7.4](#).

7.1. Waiting Time

In the research of Zare-Reisabadi and Hamid-Mirmohammadi [49] the site dependent model is presented, which is explained in the next chapter. Apart from this implementation, the model proposes another interesting feature in the linear model which is interesting for the proposed thesis subject. First of all, the main problem of this thesis is that truck congestion appears before the ground handlers. In most models which are presented in literature, the time window is considered as follows. The time window is defined as $[a_i, b_i]$ which means that the trucks prefers to arrive after a_i and before b_i . Most models incorporate the feature that a truck is allowed to arrive before a_i , but service cannot start before a_i . In this case however, it is not desired for the trucks to arrive early, because this might result into the fact that trucks still have to wait outside the ground handler. This problem can be solved in different methods. First of all, each delivery node can be allocated to a dock. If the model also ensures that a dock can never be occupied by more than one node, the problem is solved. This implies that there should be a link between the delivery nodes and trucks as well. Each truck needs to be assigned to a dock, and all the delivery nodes for this truck should be assigned to the same dock. This method would also help in determining the unloading sequencing in the development of the dock schedule. If this method turns out not to be desired or optimal, the method of Zare-Reisabadi et al. can be implemented. They introduce the *earliness* and *tardiness* decision variables. The earliness and tardiness of each node is computed which is the absolute difference between the desired arrival/departure time and the actual scheduled time. The earliness and tardiness are then incorporated into the minimization objective function.

7.2. Commodity Types

This section will present an overview of different type of commodities and how the priority of these goods can be implemented in the model. Shepherd, Shingal and Raj [50] researched the air cargo supply chain values. An overview is presented on the most imported goods by value percentage by air. From this it is concluded that live animals, plants (including cut flowers), and precious stones and metals are the dominant import products in the EU. The airport of Schiphol is known for its location near to Aalsmeer where the flower auction takes place. For the transport of flowers, it is essential to keep these cooled as good as possible. This also implies to pharmacy. It could thus be argued that these types of goods have a priority compared to other products.

The implementation of priority to certain shipments can be done as follows. First of all, in [chapter 4](#) it was discussed that the time window on the side of the ground handler can be seen as a soft constraint regarding the start time. This soft time constraint has to do with the limited storing capacity of the ground handler and to prevent overfilling the building. For the priority on goods as flowers and pharmacy a penalty function can be added. This function could be in the form of [Equation 7.1](#).

$$\text{if } s_i > a_i: \quad P(i) = c(i) (s_i - a_i) \quad (7.1)$$

This equation is only applicable if s_i (arrival time at node i) is larger than the earliest possible arrival time. This equation is not applicable if otherwise, as there is already a penalty introduced for shipments arriving early and this does not introduce additional costs for these priority goods. The absolute difference between the actual arrival time and earliest available time is then multiplied with $c(i)$. This parameter is the good type specific cost. For shipments which include pharmacy or flowers, this parameter can be set nonzero. For nodes which have no docking priority, $c(i)$ can be set to zero.

7.3. Truck Load Factor

The load factor of the truck can also be a parameter which is on influence on the priority to dock a truck or not. However, the load factor itself does not immediately introduce a very efficient operations. If this optimization method would be implemented, it might occur that only smaller trucks will be used so that the load factor is very high. In addition to that, the idea behind this method is that trucks which are filled optimally have a priority as the unloading of such a truck is more efficient compared to non fully filled trucks. In the proposed model, this increase in efficiency is however not realistic as docking the truck at the ground handler does not introduce docking time. For each delivery node, a the service time is introduced but for a truck no additional time is scheduled in case it docks. In order to introduce this feature, it is thus more realistic to introduce in parallel the truck docking time. So for each truck it is assumed that it takes x_k minutes before unloading can start. After that, the implementation of the load factor can increase the efficiency of the model. The aspect mentioned before about the load factor of small trucks can be worked around as follows. Apart from the load factor of the truck, also the capacity of the truck should be taken into consideration. The combination can be made between of the load factor and the total load (mass or volume) which is transported.

7.4. Multi-Objective Optimization

The implementation of different type of shipments to the model into the objective function can be difficult as the problem becomes a multi-objective optimization. Let us consider the example of a truck with a load factor of 10% which carries pharmacy which needs to be remained cooled. On the other hand, there is a truck with a load factor of 90% which carries post letters only. There is only one dock available. For efficiency, it is assumed that a truck has to finish unloading once is started. A single optimization would allow the first truck to dock if the optimization function is set to priority to the type of good. Alternatively, if the optimization is to set priority to load factor, the second truck would be scheduled to dock. In reality, it is however a combination of the two and the problem is a multi-objective optimization. Deb [51] states that there are two methods to solve multi-objective optimization. First of all, the different optimization functions can be separated from each other and multiple single-objective functions are solved. Deb refers to this as the *ideal multi-objective optimization procedure*. If the single-objective functions have been evaluated, the user can select the desired solution using higher-level information. If this high-level information is known on forehand, this can be incorporated into the objective function. This method is called the *preference-based optimization procedure*. A weight vector is introduced with weight for each of the objective functions. In the papers research for this literature review, most papers introduce this second method if multiple objectives are introduced. This procedure thus seems most promising and allows to compare different weight vectors to each other.

Concluding Remarks

The purpose of this chapter is to develop a brief overview of parameters which can be introduced to solve situations where the dock capacity is not sufficient enough to accommodate all vehicles. This means that priority has to be given to a specific truck or item. First of all, this research aims at reducing the waiting time outside the ground handlers. The proposed literature models all assume that a vehicle can arrive at a customer node before the time windows, but service cannot be started. This situation is undesired in this

model and the proposed solution is to assign each delivery node to a dock, as well as a truck. This chapter presents a cost function for specific commodity types which are identified to be very costly. In addition to that, a review of the truck load factor is presented. It is concluded that incorporating the load factor on its own does add a lot of value. This method can however be implemented if extra time is allocated to trucks which dock at the ground handler. Finally, the multi-objective optimization function can be incorporated in two different methods. It is concluded that introducing a weight factor for these objective functions is mostly implemented and seems to be the best for this thesis subject.

8

Model Extensions

The basic model which routes the vehicles from the freight forwarders to the ground handlers and assigns dock forms the basis of this research. It is however interesting to extend the model with specific features to make the model more realistic and complete. This is addressed by the sixth sub question as proposed at the beginning of this literature review. In this chapter, different extensions will be described in detail and the available literature on these extensions will be presented. The extensions which are considered as the following. First of all, in [section 8.1](#) the two-dimensional bin packing model is presented. After that, in [section 8.2](#) stochastic modelling is presented to make the model more robust. In [section 8.3](#) the site specific feature of the model is described. Finally, in [section 8.4](#) the time dependent feature is elaborated.

8.1. Two-Dimensional Bin Packing Problem

In [chapter 5](#) the capacitated vehicle routing problem is presented. In that problem, the capacity is simply limited by giving a weight to each shipment. The weight of all shipments in the truck cannot exceed the capacity of the truck. The dimensions of the shipment are however not taken into consideration in that model. To increase the reality of the model, the dimensions of each shipment should be added. This ensures that relatively lightweight shipments, but with very large dimensions occupy the space they do in reality. To implement this modelling technique, the two-dimensional bin packing problem can be implemented. In the research of Iori and Martello [52] an overview is given of the vehicle routing problems with loading constraints. The distinction can be made between the *Two-Dimensional Bin Packing Problem 2BPP* and the *Two-Dimensional Strip Packing Problem (2SPP)*. In the 2BPP, the items are placed in a minimum number of bins to allocate the items. In contrast, the 2SPP tries to stack all items into a single strip. The objective is then to minimize the total length of this strip. For the application of this thesis subject, the 2BPP seems more applicable. The trucks represent the bins and the number of trucks which transport the items should be minimized. The 2BPP can thus be seen as additional constraints, rather than an entire new optimization technique and new objective function. The vehicle routing problem which incorporates the 2D-loading is referred to as the *Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints (2L-CVRP)*. For reality purposes, it can be imagined that if multiple pickup and delivery nodes are visited, the order over which this is done is important. The items which are pickup up first are loaded in the front of the truck (closest to the driver). When unloading, this delivery node should thus be visited last. This principle is the Last-In First-Out (LIFO).

Iori, Salazar González and Vigo [53] present an implementation of the linear programming of the 2D-loading model. The disadvantage of the approach in this research is that a set of feasible routes is assumed to be known and this is used to see the feasibility of filling the trucks. In this model, the left bottom corner of each item expresses the placing of the item. The research incorporates the following constraints. (i) The weight capacity of a vehicle cannot be exceeded. This is already implemented in the capacitated vehicle routing problem. (ii) For all items, the dimensions cannot exceed the dimensions of the vehicle. This is done using [Equation 8.1](#) and [Equation 8.2](#) which limit the width and height of an item to be placed respectively.

$$0 \leq x_{il} \leq W - w_{il} \quad (8.1)$$

$$0 \leq y_{il} \leq H - h_{il} \quad (8.2)$$

In these equations, x_{il} and y_{il} represents the x and y coordinates of item l at node i respectively. W and H present the width and height of the truck. (iii) A sequence of constraints is added to ensure that two items do not overlap each other. (iv) Ensure the LIFO constraint. This feature is programmed as follows. If node i is visited before node j , one of the three equations below ensures that the LIFO constraint is satisfied.

$$y_{il} \geq y_{jl'} + h_{jl'} \quad (8.3)$$

$$x_{il} + w_{il} \leq x_{jl'} \quad (8.4)$$

$$x_{jl'} + w_{jl'} \leq x_{il} \quad (8.5)$$

However again, it should be noted that for this approach, it is not entirely clear how the vehicle routing is implemented in this model. It would be desired to have the 2D bin packing model included in the vehicle routing and the model above requires the order over which the nodes are visited to be known in advance.

In the literature review of Pollaris, Braekers, Caris, Janssens and Limbourg [54] an overview of the current literature on vehicle routing problems with loading constraints is presented. Pollaris et al. conclude that only two papers implement the two dimensional bin packing problem as an exact approach. The first is the research of Iori et al. [53] which is mentioned above. The other paper is the research of Martínez and Amaya [55] where circular items are considered. In this research, the LIFO constraint is not implemented and it is modelled as a vehicle routing problem where split deliveries are not allowed. In this model, two decision variables are added which determine the position of the an item. Martínez and Amaya defined these as follows. x_{igrk} is the x-location of the center of the circle for item i , for product reference g , if this order is located in route r with vehicle k . The same variable is defined for the y-location. The model then presents three constraints which ensures that the circular shapes of different items do not overlap. The disadvantage of using circular items is that a specific set of constraints is not linear anymore. The model thus becomes non-linear which increases the computational time significantly. The advantage of this research compared to the research of Iori et al [53]. is that in this research the full model is presented. However, as mentioned before, the LIFO constraint is not considered in this research. For this thesis subject ideally the two models would be combined. First of all, the model of Pollaris et al [54]. is changed to rectangular items and afterwards, the LIFO constraints from the model of Iori et al [53]. should be implemented.

To increase the complexity of the problem even further, the *Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints (3L-CVRP)* can be introduced. According to Pollaris et al. [54] solving this problem exact is only done by Junqueira, Oliveira, Carravilla and Morabito [56]. In this model, the decision variables are modelled time dependent to allow for loading and unloading conditions to be known at all times. The number of decision variables is therefore really large. For example: a_{xyz}^{ikt} which is equal to 1 if box type i for customer k is visited at time t by vehicle v and the package is placed at the (x,y,z) position and 0 otherwise. The model then proposes a set of complicated constraints to ensure that items are not overlapping. However, this paper presents a different method as compared to the model of Pollaris et al [54]. and Iori et al [53]. since it presents a binary representation instead of a continuous variable approach.

In the research of Khebbache-Hadji, Prins, Yalaoui and Reghioui [57] the linear programming model is presented first which is followed by a heuristic. This model implements the vehicle routing problem with time windows with the two-dimensional loading constraint. In this model, y_{tk} this binary representation equals 1 if item t is transported by vehicle k . In addition to that, two decision variables are introduced, (u_{tk}, v_{tk}) , which represent the x and y coordinate of an item respectively. This methods seems more promising compared to the method from Junqueira et al [56]. since the number of decision variables seems to be significantly lower. The linear model however does not implement the LIFO constraint. This model also presents a heuristic method which is based on a sequential insertion heuristic. An overview is presented in Figure 8.1.

The research presents three different packing heuristics. The most promising methods are the *Maximum Touching Perimeter Method (MTP)* and the *Shelf heuristic filling (SHF)*. For the first, the touching perimeter is computed which is defined to be the total length of an item touching another item or the loading walls of the truck. The starting point is set to be the left bottom corner. The truck starts filling from this side. The SHF in contrast, tries to place items based on horizontal shelves. As in this specific thesis case, there is no horizontal collaboration between freight forwarders, the LIFO constraint is simplified slightly. Namely, the ordering sequence on how the freight forwarders are visited is not relevant. It is only required to be able to implement the packages as the final order of the schedule.

Another method to approach this problem is to introduce the system of loading stacks. One can make the distinction of a truck with a single stack or multiple stacks. In a single stack system, a truck has to visit the

Algorithm 1 – General structure of 2L-CVRPTW heuristics derived from SIH

```

1: repeat
2:   initialize a new route  $T$  as a loop on the depot and an empty vehicle
3:    $L \leftarrow 0$ 
4:   loop
5:      $C_{min} \leftarrow +\infty$ 
6:     for each unrouted customer  $i$  with  $L + q_i \leq Q$  do
7:       call a packing heuristic (CAV, MTP or SHF) to see if the items of  $i$  can be loaded
8:       if yes then
9:         compute the cheapest insertion cost  $C$  of  $i$  in  $T$  and the associated position  $k$ 
10:        if  $C < C_{min}$  then
11:           $C_{min} \leftarrow C$ 
12:           $i_{min} \leftarrow i$ 
13:           $k_{min} \leftarrow k$ 
14:        end if
15:      end if
16:    end for
17:    exit when  $C_{min} = +\infty$ 
18:    insert customer  $i_{min}$  at position  $k_{min}$  in route  $T$ 
19:    update the loading of the vehicle
20:     $L \leftarrow L + q_{i_{min}}$ 
21:  end loop
22: until all customers are inserted

```

Figure 8.1: Heuristic Structure of the 2L-CVRPTW [57]

delivery nodes in exactly the opposite order of the pickup nodes. This is also the strict LIFO method mentioned before. However, if the two-stack system is implemented, this is not strictly needed as two unloading locations are present. The implementation of the single stack method is presented by Cordeau, Iori, Laporte and Salazar [58]. In this research, three set of constraints are introduced which ensure the LIFO constraint in a smart way. The equations are presented below.

$$Q_j \geq (Q_i + q_j)x_{ij} \quad \forall (i, j) \in A \quad (8.6)$$

$$Q_{n+i} = Q_i - q_i \quad \forall i \in P \quad (8.7)$$

$$\max\{0, q_i\} \leq Q_i \leq \min\{Q, Q + q_i\} \quad \forall i \in N \quad (8.8)$$

Where Q_i is the load of the truck at node i and q_i is the demand at node i . In this set, A is the set of arcs, P the set of pickup nodes and N the total node set (including origin and destination node). In Equation 8.6 the load of a node is computed if the route is travelled. This equation is however non-linear as both Q_i and x_{ij} are decision variables. To linearize this equation, Equation 8.9 can be implemented where $W_j = \min\{Q, Q + q_j\}$.

$$Q_j \geq Q_i + q_j - W_j(1 - x_{ij}) \quad (8.9)$$

In Equation 8.7 the demand at the delivery node ($n+i$) is expressed with respect to the pickup situation. Finally, in Equation 8.8 the values for Q_i are expressed within the physical boundaries. With these three equations, the LIFO constraint is implemented based on a single loading stack in the truck.

Alternatively, one could introduce the multi stack system. Côté, Gendreau and Potvin [59] introduce for each vehicle a set of M loading stacks. The same principle to introduce the LIFO constraint is done for each loading stack as expressed by Cordeau et al. [58]. Each loading stack is limited in size and this is incorporated in the proposed constraints. This method seems to be very promising for introducing the LIFO constraint combined with the loading constraint. Côté et al. [59] introduce a new decision variable which is s_{ik} . This variable should not be confused with the arrival time at a node as introduced previously. It represents the load of stack k after leaving node i . Where k is selected from the set of stacks. In addition to that, the variable y_{ik} represents if the demand at pickup location i is loaded in stack k . Constraints enforce that for example: each pickup location is assigned to one stack exactly. Then the LIFO constraint is included and capacity limits for each of the stacks. To enforce that the loading starts at a specific location in the truck and to avoid multiple possible solutions, the model tries to break symmetry with two rules. First of all, the first pickup location is loaded in stack 1. After that, it is ensured that demand can only be loaded in a new stack if the previous stack is already used.

The disadvantage of this model is that it is only applied to a simplistic pickup and delivery problem and is for a single vehicle only. The model should thus be extended from the two- to three-index formulation. The

introduction of the stack model simplifies the model in terms that each item does not have to be precisely scheduled in a 2D-plane. This simplification also introduces inaccuracies and infeasibilities. The most likely assumption for a truck would be to introduce a two-stack system. It is then assumed that each shipment can be fitted in this stack. If the dimensions exceed the width of the stacks, the shipment could in reality not be transported.

8.2. Stochastic Modelling

One can imagine that the input variables of a model are very important for the outcome. For example, the time it takes to travel from node i to node j . In most models, this time is assumed to be static which results in the optimal solution. In reality, the travel time between two nodes is not static. It is depending on traffic and human resources for example. To incorporate this feature, stochastic modelling can be applied. By doing so, the reliability and robustness of the model is increased. In the research of Ritzinger, Puchinger and Hartl [60] a distinction is made between the *dynamic* and *stochastic* vehicle routing problem. The dynamic vehicle routing problem refers to input data which gets available during the execution of the model. The stochastic variant includes uncertainty of the input variables to the model. In their research, Ritzinger et al. [60] consider the following aspects to incorporate in the uncertainty model: stochastic travel times, stochastic demand, stochastic customers and multiple stochastic aspects. Especially the first aspect is relevant for this specific thesis. If a truck is delayed from the freight forwarder to the ground handler, situations might appear in which the number of docks is violated whereas it would not have been if the truck was not delayed.

The second aspect, the stochastic demand is for example applied in the field of garbage collection. A garbage truck is sent out on a specific route based on historical data on that route. In combination with the capacity of the trucks, the optimal route is determined. The demand is however fluctuating and cannot be seen as static. Implementing stochastic demand can thus be helpful for this problem. For this proposed thesis subject, the application of applying stochastic to the demand does not seem very needed and helpful. The demand for this problem (which are the orders to be shipped from freight forwarder to ground handler) is not stochastic. The orders are known in advance and this forms the basis of the optimization problem. Further research in this direction is therefore not done.

The third aspect which is discussed in the research of Ritzinger et al. [60] is about the stochastic customers. This implies that there are customers which only present their demand during the operation of a day. This application is also not considered to be very realistic in the application to this specific thesis subject. As the shipments need to be flown by aircraft, these shipments are normally reserved a spot in the aircraft not last minute. Changing the number of customers (extra freight forwarders) during the day is therefore also not considered to be realistic and this feature is therefore not studied in more depth.

To summarize, the only promising implementation of stochastic features into the model is time related. This could be the travel time between nodes, but also the processing time could in reality be susceptible to uncertainty. Ritzinger et al. [60] found that there is only a single paper which tries to implement time uncertainty in a linear model. This is done by Toriello, Haskell and Poremba [61] where the arc cost is determined with a stochastic approach. The proposed model is however very complicated and including this type of model does not seem realistic as the basic model can already be very complex. Berhan, Beshah, Kitaw and Abraham [62] also present a literature review on the stochastic vehicle routing problem and conclude that most papers focus on stochastic customer demand. Berhan et al. [62] found that Campbell, Gendreau and Thomas [63] also present an exact method to incorporate this feature. This research presents a very probability based outline. First the model defined D as the time limit at which the node should be visited. The time at which a node is visited is referred to as A_i . If $A_i \leq D$, the reward r_i is given. If the vehicle arrives too late, the penalty e_i is awarded. The model then expressed the expected profit to be as in Equation 8.10 where $v(\tau)$ represents the expected profit of tour τ .

$$v(\tau) = \sum_{i \in \tau} [P(A_i \leq D) r_i - (1 - P(A_i \leq D)) e_i] \quad (8.10)$$

The model presents three different methods to incorporate this feature, based on different scenarios. After that, the research proposed a heuristic based on the variable neighborhood search. The implementation of this feature in the model seems interesting to make the model more robust. However, all papers which are found in literature apply the problem to a relatively less complex vehicle routing problem compared to the proposed problem for this thesis. As the model should already incorporate vehicle routing, time windows, dock capacity and possibly the two-dimensional load constraint, this feature seems to be very complicated at this stage.

In the research of Weskamp, Koberstein, Schwartz, Suhl and VoSS [64], a two stage programming method is introduced. This research does not specifically focus on the vehicle routing problem but considers the air cargo supply chain more generally. In the first stage of this approach, decision variables are defined for which full information is available. In the second stage, uncertainty is introduced into the data. In the second stage, postponed decisions are made with respect to the introduces randomness. The model presents a exact method in the form of a linear program which is solved using CPLEX. Weskamp et al. [64] also present two other methods to solve the problem. Applying this specific research to the subject of this thesis does not seem very intuitive especially since the research is not applied to the vehicle routing problem and the detailed working principle in terms of programming and steps which are taken in the algorithm do not become very clear from this research.

8.3. Site Dependent Model

The assignment of aircraft to gates can be very specific which can be easily visualized. Larger aircraft as the A380 or B747 cannot occupy a gate intended for small aircraft due to physical dimensions. The same principle can be applied in the truck scheduling problem. The distinction can be made regarding size of the truck but also other properties such as the presence of a cooling compartment in a truck. Only specific docks can handle these cooling compartments or are designed for larger trucks. This section will present the literature on this implementation of the model.

First of all, in section 5.2 the distinction has been made between the two and three-index decision variable for the vehicle routing problem which are represented as x_{ij} and x_{ij}^k respectively. The disadvantage of the two-index model is that one has to assume a homogeneous fleet. The three-index model allows a heterogeneous fleet, but the computational time is generally larger as the number of decision variables is increased. The three-index model makes it possible to assign certain shipments to specific trucks only. For example: if flowers or pharmacy needs to be delivered, these items can only be shipped by trucks which have a cooling department. These trucks should then be scheduled to a dock which can also serve this cooling compartment. To implement this site specific features, the three-index model seems to be the best model. In addition to that, this model is also more intuitive in modelling and simplifies the understanding of the model. The increased computational time should be accepted in that case.

The vehicle routing problem which incorporates this feature is the *Site Dependent Vehicle Routing Problem* which is addressed by Zare-Reisabadi and Hamid-Mirmohammadi [49]. The model proposes a heuristic to include this in the model based on an ant colony optimization. The first stage is to produce a feasible solution and after that a local search is performed in which the solution is improved.

The model of Alonso, Alvarez and Beasley [65] provide a linear method to include site specific features. For their research, this was the accessibility restriction. Only a subset of vehicles can service specific nodes. To incorporate this feature, the set of vehicles has been broken down. The P_i set is introduced which is the set of feasible vehicle types for customer i . For each node, these subsets are then used in the constraints. This model especially incorporates the possibility of multiple routes per vehicle and the time span of the data set can be multiple days. For this specific thesis subject, the implementation of multi routes does not immediately seem to have any added value. The subset of constraints is however usable in the model.

8.4. Time Dependent Model

As mentioned earlier in this report, Soysal et al. [15] and Franceschetti et al. [17] developed a time-dependent model. The addition of this to the model which will be developed does not immediately seem to be quality improving. However, there are some time dependent features which could be interesting. The most interesting seems to be that the number of docks which is available at the ground handler is time dependents. At peak hours, the number of docks is larger and at non-peak hours the capacity is decreased. Having less docks available also means that less workers are required to handle the docks, meaning a cost reduction.

The literature on dock capacity combined with the pickup and delivery problem is very limited. The papers on cross-docking where the dock capacity is taken into consideration also does not provide this feature. All assume that the number of doors is constant throughout the model. The implementation of time dependent dock availability would be less complex if the decision variables include a time aspect, which they do not. In order to work around this, the following method could be used. For example: during peak hours 5 docks are available, and otherwise only 4. For the one docks which is only partially available, a dummy node and truck can be generated. The arrival time of this dummy truck to the dummy node is set equal to the

time the dock closes and the departure time to the opening of the dock. This dummy truck should then be assigned to this one dock. This ensures that no other truck can unload at that dock when it is closed.

Concluding Remarks

The purpose of this chapter was to study possible extensions of the vehicle routing problem. First of all, a basic program should be developed as presented in [chapter 5](#) and [chapter 6](#) which performs the truck routing and dock scheduling. After that, the model can be extended so that it represents reality better. This chapter provided three methods to do this. First of all, the two dimensional bin packing problem has been presented. Two different methods are presented here. The research of Khebbache-Hadji et al. [57] model the x and y coordinate of an item with a continuous variable. This is also done by Iori et al. [53] and here a method to incorporate the LIFO constraint is presented. Alternatively, if the modelling is done using binary variables, the method of Junqueira et al. [56] can be applied. In this model, the decision variable is in the form of x_{xyz} which has the unitary value if the package is placed at (x,y,z) and 0 otherwise.

The second extension is the stochastic behaviour of the input data to make the model more robust. Literature defines three aspects where a stochastic approach can be implemented: travel times, demand and customers. In this chapter it is concluded that the uncertainty of demand and customers does not seem realistic for this thesis subject as the reservation of freight forwarders to air cargo companies cannot be done last minute. The uncertainty related to travel time and service time is however realistic and would be an enrichment of the model. From the available literature on this topic it is however concluded that the implementation of this will not be trivial. The models found in literature are often build specifically around the stochastic behaviour and the considered model is often less complex than is considered in this thesis. Implementation of this feature to the existing model therefore seems to be very challenging.

The site dependent vehicle routing model has been studied which implies that the three-index notation should be used, x_{ijk} . The constraints should then be made specific by creating subsets of the vehicle types for the nodes. The implementation of this model could be for example related to trucks which have cooling compartments. This thus implies that only specific nodes can be served by a type of vehicle which has these features. In addition to that, the model can be elaborated to give each dock properties. This implies that the scheduling of the trucks at the ground handlers' dock also incorporated. These two extensions of the model seem to be realistic and can be implemented if the basic model has been developed. At last, the time dependent model has also been studied. This model is especially interesting in case certain docks are only open for peak hours.

9

Conclusion

The aviation cargo industry is continuously growing and this introduces the need to improve the efficiency of the air cargo supply chain. The currently observed problem is that the number of trucks which arrive at ground handlers is larger than the available number of docks. This leads to congestion and delays for the trucks. The purpose of this research is to develop a model which performs truck routing from the freight forwarders to the ground handlers. In addition to that the model should take into consideration the available docks at the ground handler and create a schedule for all individual docks. The research question is defined as follows.

What is the efficiency improvement of introducing an optimization model which incorporates dock scheduling within the pickup-and-delivery model for the landside air cargo supply chain

In addition to this research question, six sub-questions have been developed. The purpose of this report is to answer these six sub-questions. After these are answered, the model can be developed and evaluated. This conclusion will go through each of the sub-questions and present the result of the chapter.

What is the current procedure in the air cargo delivery process?

In the current air cargo supply chain, the freight forwarders and ground handlers play a major role. The freight forwarders receive items from different companies and act as a shipping company to the ground handlers. The freight forwarders send shipments to different ground handlers, depending on the airline which is used for transporting the shipment. The ground handlers process the incoming shipments and consolidate them for airlines. Ground handlers have a limited number of docks available for trucks to dock. To increase the efficiency of the process from freight forwarder to ground handler, multiple efficiency improving concepts have been discussed in literature. These are primarily aimed to reduce the number of trucks arriving at the ground handlers by increasing the efficiency and load factor of the trucks.

First of all, the introduction of a time slot booking system could prevent uncoordinated congestion. The disadvantage of this approach is that the interaction between different ground handlers to be visited along the route of a truck can introduce major inefficiencies if the allocated time slots are not well attuned to each other.

Collaboration between ground handlers and freight forwarders can be done by vertical and horizontal collaboration. In vertical collaboration, collaboration is implemented between companies who operate at different levels in the supply chain. Alternatively, horizontal collaboration is the cooperation for companies within the same level in the supply chain. Introducing this full collaboration can reduce the transportation cost in the air cargo supply chain up to 40% theoretically. In practise, full collaboration is however rarely implemented due to practical problems.

If collaboration between freight forwarding companies is difficult to implement, cross-docking can help to increase the load factor of departing trucks. At the cross-dock stations, trucks from all freight forwarders arrive and unload their shipments. The shipments from different freight forwarders are then combined in a truck which travels to the ground handler.

The last efficiency improving concept is the milk run principle. A truck travels a route and picks up shipments at each stop. On forehand, the amount of shipments to be picked up at each freight forwarder should

be known to load the truck as efficiency as possible. This principle prevents that each freight forwarder drives half-full trucks to the ground handler. This concept has been introduced at Schiphol Airport since May 2015 and showed a decrease of 40% truck movements at the ground handlers and 30% decrease in carbon dioxide emissions one year after the introduction.

What indicators are relevant to determine the efficiency of the ground handling process?

The key performance indicators (KPI) can be used to evaluate the quality of a proposed solution to each other. The objective function in the optimization model can be altered and the resulting outcome can be compared with the proposed KPI's. For most vehicle routing problems, the travel distance is to be minimized. The cost of a solution is simply represented by the total distance which is travelled by all vehicles. In this specific thesis subject, the travel distance is however not the only parameter which should be considered in the evaluation of the solution. The aspect of time is very relevant as well. The time component can be broken down into travel time and waiting time. The waiting time especially should be minimized to avoid trucks waiting at ground handlers for a dock. The fuel costs are also incorporated in some vehicle routing problems but seem to be too detailed for this thesis subject. The financial costs of a solution are however interesting to incorporate in the model. If an extra truck is scheduled, this introduces additional fixed cost per truck. Also missed deliveries significantly contribute to the cost function. The time window preferences should be evaluated in the model as well. Time windows can be implemented as soft and hard constraint. In case a soft time window is introduced, penalties should be introduced if violation occurs. Finally, a benchmark dataset has been introduced which can be used to evaluate the model and perform validation compared to other models.

What linear programming methods are suitable for the truck routing combined with the scheduling problem?

The linear model should combine the routing of trucks and allocation of docks at the ground handlers. The routing model can be seen as a variation of the vehicle routing problem. In the vehicle routing problem, a set of nodes should be visited by a set of vehicles which minimized the total distances travelled. Numerous extensions of the model are presented in literature such as capacity, time windows and a multi depot variant. The variant which is most suitable for this thesis subject is the Pickup-and-Delivery Problem with Time Windows. In this model, shipments consist of a pickup and delivery node. The number of nodes is thus twice the number of shipments. In addition to that, a departure and arrival depot are included in the set of nodes. The pickup nodes represent the freight forwarders where a shipment is loaded in a truck. The delivery nodes represent the ground handlers where a shipment is unloaded from a truck. Time windows are incorporated for both the pickup and delivery nodes. For the pickup nodes, this can be a hard constraint from what time a shipment can be picked up. For the delivery nodes a hard time constraint can be incorporated which ensures that a shipment is delivered before the departure time of the intended flight. An earliest time arrival for the delivery nodes can be incorporated as a soft time window which is related to the maximum storage capacity of the ground handlers.

The implementation of the pickup-and-delivery problem in a linear model can be done with different formulations. First of all, a two index formulation, x_{ij} , is one if the route from node i to j is travelled and zero otherwise. To extend this model, the three-index formulation can be used for a heterogeneous fleet, x_{ij}^k . This indicates if vehicle k travelled from node i to node j . In the set partitioning formulation, all feasible routes are considered and the route with minimal costs is selected. Finally, the commodity flow formulation models the flow over each arc. It is concluded that the three-index formulation seems to be most suitable for this thesis subject.

The disadvantage of the formulations presented above is that these are not time related. It is thus not possible to simply set for each time point: the number of trucks which is assigned to a dock has to be equal or less than one. The scheduling should thus be done separately. In literature, different models tackle this type of problem. First of all, the trucks can be assigned to a dock using another vehicle routing problem. Each route represents a dock and the nodes in a route represent the trucks that have to be assigned to a dock. The order in which the nodes are visited in a route is equal to the sequence over which the trucks can visit the dock. Another method is to have a decision variable which indicates if node i is visited after node j . All nodes are assigned to a specific dock and with special interaction between constraints it is ensured that for each dock, never more than one truck is assigned. This principle is applied in the dock assignment of cross-docking in literature. Also the assignment of flights to gates can be seen as a variation of the truck dock assignment

problem. In literature a slightly different method is introduced. Each flight is assigned a to a gate and then two additional constraints are introduced which ensure that each flight can be followed by at most one flight and at most one flight can be the predecessor of it. This ensures that at each gate never more than one aircraft is assigned.

What heuristic methods are suitable for the truck routing combined with the scheduling problem?

The linear model can solve a small dataset to optimality in a reasonable time limit. The problem is however NP-hard which means that for larger datasets, a linear solver might take very long and not able to present a feasible solution. For that reason a heuristic should be developed which performs good in terms of solution quality and computational time. In this literature review, an overview is presented of the relevant meta-heuristics for the vehicle routing problem and the dock assignment problem.

For a heuristic of the pickup-and-delivery problem, the procedure starts with an initial solution. This can be done by the Clarke and Wright simple heuristic. First of all, each request is given to a separate truck. It is then tried to combine and merge different routes into each other. The operation which has the best saving is accepted. The initial solution is then forwarded to the improvement heuristic phase which aims at improving the initial solution.

In the (Adaptive) Large Neighborhood Search (ALNS) a number of request is removed from the current solution and placed in the request bank. The request are then reinserted in the solution. If the cost of the new solution is better than the 'old' solution, the new solution is accepted. The removal and insertion of the requests is done using dedicated heuristics. The specific feature of the Adaptive Large Neighborhood Search is that the success of each heuristic is recorded. Over time, the heuristics which perform best have a higher probability of being selected.

The accepting criteria of the ALNS as mentioned before is that if the new solution is better than the old, it is accepted. This might however lead to a local optimum instead of the global optimum. This is where Simulated Annealing can be of added value. A solution which is better than the best known solution is always accepted. A solution which is worse than the best known solution is accepted with a probability related to the difference in solution quality. Over time the probability of accepting a worse solution is decreased.

Tabu Search (TS) is another type of heuristic in which an operation is set tabu for a number of iterations. This ensures the diversification of the solution space en preventing a solution to go back to its original solution. Also a method is presented in which non-feasible solutions can be accepted and a penalty function is introduced.

The Genetic Algorithm (GA) is based on the principle of evolution, survival of the fittest and mutations. Solutions should ideally be represented as a binary string. Parent strings form children and over time the solution quality is improved by selecting the best parents for each iteration. Mutations can also be introduced to explore a broader solution space. The application of GA for the PDPTW is complex as the representation of the solution is not trivial. Finally, Dynamic Programming (DP) has also been found in literature. It is however shown that only a limited number of customers can be visited.

Applying a heuristics for the dock allocation problem itself is not directly useful as this implies limited interaction with the truck routing model. Most heuristic models make use of a combination of the above mentioned types. Most models use ALNS or TS with Simulated Annealing.

Which factors influence the priorities to allocate trucks to a dock?

If two trucks would arrive at exactly the same time at a ground handler, one should be given priority. This can be incorporated in the objective function of the model. First of all, the waiting time can be minimized. This can be done by incorporating a penalty function for the earliness and tardiness variables. These are the absolute difference between the desired arrival/departure time and the actual scheduled time.

Alternatively, the commodity type which is carried by the truck can be of influence for the priority. For each commodity type, a cost function can be discussed. For example, pharmacy has a higher cost for arriving later compared to post letters. The cost parameter can be scaled directly to the time difference between the actual scheduled time and the earliest arrival time.

The truck load factor can also be used to give priority to certain trucks. This method is however not entirely applicable, as this could lead to situations in which smaller trucks are used to ensure a high load factor. A combination should thus be made between load factor and truck capacity to introduce this effectively.

Finally, the objective introductions above lead to a multi-objective optimization. It is concluded that most effective procedure is the preference-based optimization procedure in which weight factors are given

to different optimization functions.

What are possible extensions of the model to be implemented later to make the model more complete?

The last sub-question is related to the extensions of the model to represent reality better. First of all, the two-dimensional bin packing problem is considered. The most complex method is to pack all items in the two-dimensional plane. This implies that each shipment is given an x and y coordinate and constraints are added to ensure that no shipments overlap with each other. The second method is based on a (multiple) stack truck. The width of the truck is divided into multiple stacks. Stacks are loaded until the capacity of a stack is reached. The method of implementing stacks seems to be most promising for this thesis model.

In addition to the packing problem, the Last-In First-Out (LIFO) principle can be implemented. This implies that the shipment which is picked last, needs to be delivered first. With this procedure, no unloading of other items has to be done in order to access the shipment which has entered in the truck last. This can be done in two ways. First of all, the order over which the delivery nodes can be based upon the x and y coordinates of the items. Secondary, the order over which the pickup nodes are visited can be inverted and this is the order over which the delivery nodes have to be transported.

Stochastic modelling has also been considered which makes the model more robust. Stochastic behaviour can be implemented in travel time, demand, customers and a combination of the three. The stochastic models which have been studied in this literature review are all entirely built around the stochastic behaviour of the model. The implemented vehicle routing problems are also less complex compared to the proposed model for this thesis. The implementation of stochastic modelling for this thesis subject is therefore concluded to be probably too complex.

The site-dependent model allows to have features which are only applicable to a specific site. In this thesis subject, this can be implemented by incorporating the three-index formulation, which is vehicle dependent. The vehicle dependent formulation allows to dock specific trucks to specific docks.

A time dependent model can be implemented to have a limited number of docks available at specific times. This can be done by incorporating a dummy truck and allocate it to the docks which are not available at that time. A more advanced time dependent model allows to have travel times which are dependent on the time of the day. This feature however does not seem to be very relevant for this thesis subject.

In conclusion, the Pickup-and-Delivery Problem with Time Windows seems to be the best method to model the proposed thesis problem. In addition to this, a dock assignment feature should be added which can be based on the vehicle routing problem or time dependent variables. A heuristic should be developed and the most promising method is the (Adaptive) Large Neighborhood Search or Tabu Search with Simulated Annealing as solution accepting criteria.

III

Supporting work

Appendix 1

Verification Case Heuristic Model

This section presents an example case for the heuristic model based on an instances that is easily solvable by hand. First of all, the data instance is presented in [Table 1.1](#). Pickup nodes are represented by *P* in the *Node Type* column and delivery nodes with *D*. It can be seen that all pickup nodes are located at UPS and have to be delivered at MNZ. Also the time windows for all nodes are presented. It should be noted that pickup node 1 belongs to delivery node 6, pickup node 2 to delivery node 7 etc. All nodes have a processing time of 10. All time variables are represented as minutes.

Table 1.1: Verification data instance

Node ID	Node Type	Location	FF	GH	Start Time	End Time	Processing Time
1	P	UPS	UPS	MNZ	20	20	10
2	P	UPS	UPS	MNZ	30	30	10
3	P	UPS	UPS	MNZ	40	40	10
4	P	UPS	UPS	MNZ	60	65	10
5	P	UPS	UPS	MNZ	59	75	10
6	D	MNZ	UPS	MNZ	1	130	10
7	D	MNZ	UPS	MNZ	1	120	10
8	D	MNZ	UPS	MNZ	1	90	10
9	D	MNZ	UPS	MNZ	1	110	10
10	D	MNZ	UPS	MNZ	1	100	10

The start phase orders the delivery nodes according to the start time window. It thus creates the pickup route: [1,2,3,5]. The associated service time of the nodes is: [20,30,40,58]. Servicing pickup node 5 is finished at $t = 68$. It then wants to implement pickup node 6, but this is not feasible with respect to the provided time window. It thus creates a new route for pickup node 4. However, we know that we could get a feasible solution if we have the pickup route [1,2,3,4,5] with associated service time of the nodes at: [20,30,40,60,70]. The order over which the delivery nodes should be visited is restrained by the end time window of the delivery nodes. It should thus be in the order: [8,10,9,7,6]. This change is achieved in the improvement phase of the heuristic. The outcome after the heuristic phase is presented in [Figure 1.1](#). The x-axis represent the time in minutes when a node is serviced. The node is indicated by the number above the line. The vertical black lines indicate the start and end time of servicing the node. The start depot and end depot node are included as well.

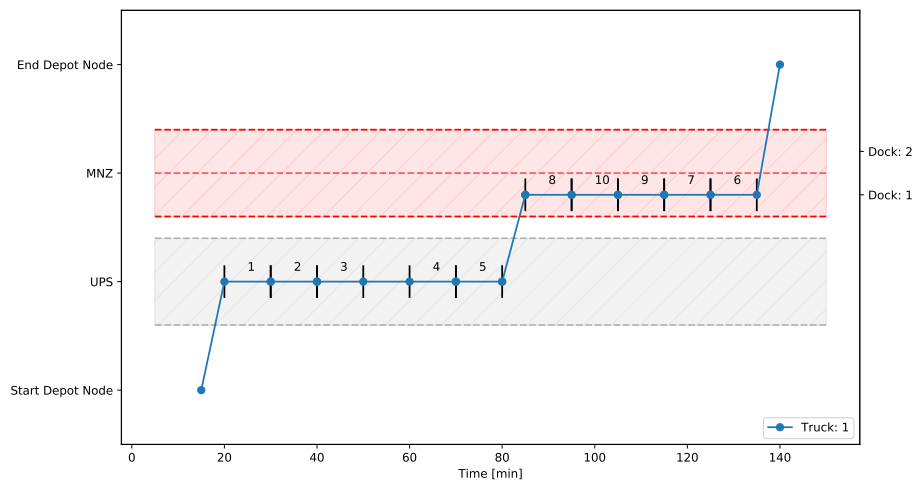


Figure 1.1: Simple verification case route outcome

It should be noted that the travel time from UPS to MNZ is set equal to 5 minutes. The earliest time at which a delivery node can be serviced is equal to: $70 + 10 + 5 = 85$. The total time that the solution takes is equal to: 10 nodes with each a service time of 10 minutes. In addition to that the travel time of 5 minutes. Finally, we have 10 minutes inter-node time due to the time windows. In total, the minimum duration is thus a total of 115 minutes. In addition to that, one truck is used. The total solution cost should thus be equal to: $115 \cdot 0.45 + 112.26 = \164.01 . This cost is exactly the cost that is outputted by the heuristic model after the improvement phase has been completed. It is thus assumed that the working principle of the heuristic is as it is expected to work. The loading constraint is also satisfied and output is presented in Figure 1.2. The position where *Truck 1* is plotted should represent the driver cabin and the unloading door is presented at the bottom of the figure. This loading pattern is feasible for the Strict-LIFO and Side-Accessible LIFO model variant.

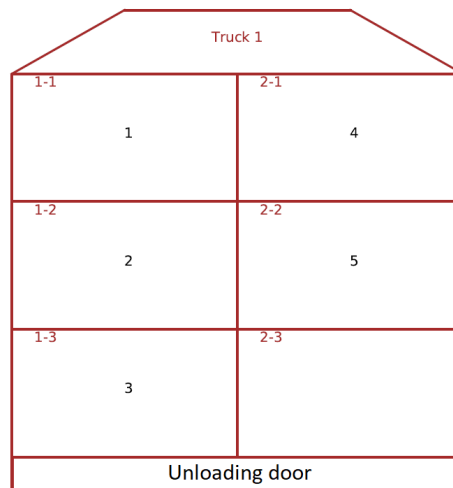


Figure 1.2: Truck loading verification case

2

Appendix 2

Data Instance Generation Process

In this appendix, a brief explanation to the data generation process is presented. In [Table 2.1](#) an overview of all possible ground handlers and freight forwarders is presented. The datasets are generated using a random generator. The user selects the number of shipments for the dataset. Also the number of freight forwarders and ground handlers is selected. From [Table 2.1](#) the number of freight forwarders and ground handlers are randomly selected. The generator ensures that each freight forwarder has assigned at least one shipment. The number of shipments should therefore always be larger or equal than the number of freight forwarders. The ground handler for each shipment is randomly selected. If all nodes have been assigned to a freight forwarder and ground handler, time windows are introduced to the nodes. In [Table 2.2](#) an overview is presented for the determination of the time windows.

Table 2.1: Freight Forwarder and Ground Handler location

Freight Forwarder	Ground Handler
DHL Global Forwarding (DHL)	KLM Ground Handling (KLM)
Expeditors International Forwarding (EX)	Dnata (DNT)
Kuehne Nagel (KN)	Menzies (MNZ)
DB Schenker (SCH)	Worldwide Freight Services (WFS)
UPS (UPS)	Swissport (SCS)

Table 2.2: Time Window Data Generation

Variable	Distribution Type	Parameter Value
Start Time Freight Forwarder	Discrete Uniform Distribution	$U(t_{start}, t_{end})$
Travel Time	Normal Distribution	$N(\mu_{tt}, \sigma_{tt}^2)$
FF Time Window	Normal Distribution	$N(\mu_{ff}, \sigma_{ff}^2)$
GH Time Window	Normal Distribution	$N(\mu_{gh}, \sigma_{gh}^2)$
Parameter	Value	
End Time Freight Forwarder	Start Time Freight Forwarder + FF Time Window	
Start Time Ground Handler	Start Time Freight Forwarder + Travel Time	
End Time Ground Handler	Start Time Ground Handler + GH Time Window	

When generating the data instance, the user selects a start and end time window where the shipments can be generated. The start time window at the freight forwarders is created from an uniform distribution between these two times. An uniform distribution is chosen to prevent peaks during the entered data instance

time duration. For the ground handler and freight forwarder, a random time window is introduced that is based on a normal distribution. In addition to that, a normal distribution is introduced for the travel time from freight forwarder to ground handler. The end time window of the freight forwarder is then simply the start time of the freight forwarder plus the random time window of the freight forwarder. The start time window of the ground handler is defined as the start time window of the freight forwarder plus the random travel time. Finally, the end time window of the ground handler is defined as the start time window of the ground handler plus the random time window of the ground handler.

It should be noted that not for all data instances, a start and end time window for ground handlers and freight forwarders is used. Most times, only a start time window is used for freight forwarders. This indicates the earliest time at which a shipment can be picked up at the freight forwarder. For delivery nodes at the ground handler it was decided to include in most cases only a end time window. This indicates the time at which the node should be at the ground handler in order to be on time for the scheduled flight. The way that the time windows are computed is not changed, but the time windows which are not used are simply set to values that

Bibliography

- [1] Trepel Airport Equipment, “Cargo high loaders.” <https://trepel.com/images/max/trepel-airport-equipment-cargo-high-loader-01.jpg>. Retrieved on: 21-06-2021.
- [2] Boeing Commercial Airplanes, “World air cargo forecast 2018 - 2037.” https://www.boeing.com/resources/boeingdotcom/commercial/about-our-market/cargo-market-detail-wacf/download-report/assets/pdfs/2018_WACF.pdf. Retrieved on: 25-09-2020.
- [3] Schiphol Airport, “Schiphol cargo figured out.” <https://www.schiphol.nl/en/cargo-statistics/>. Retrieved on: 25-09-2020.
- [4] T. Verduijn, S. Vunderink, A. Bombelli, L. Tavasszy, D. v. Damme, R. Romero-Silva, and M. Mujica Mota, *Smartest Connected Cargo Airport Schiphol: Onderzoek naar innovaties die de logistieke prestaties van luchtvrachtketens op Schiphol versterken*. Amsterdam: Hogeschool van Amsterdam, Urban Technology, 2019.
- [5] Schiphol, “Solving truck congestion at amsterdam airport schiphol: Fresh perspective.” <https://www.schiphol.nl/nl/cargo/pagina/solving-truck-congestion/>. Retrieved on: 13-10-2020.
- [6] Cargonaut, “Smartest connected cargo airport schiphol.” <https://www.cargonaut.nl/smartest-connected-cargo-airport/>. Retrieved on: 15-10-2020.
- [7] Nallian, “Slot booking at brussels airport’s cargo hub.” <https://www.nallian.com/applications/slot-booking/cs-slot-booking-bru>. Retrieved on: 15-10-2020.
- [8] A. Prakash and S. Deshmukh, “Horizontal collaboration in flexible supply chains: a simulation study,” *Journal of Studies on Manufacturing*, vol. 1, no. 1, pp. 54–58, 2010.
- [9] S. Ankersmit, J. Rezaei, and L. Tavasszy, “The potential of horizontal collaboration in airport ground freight services,” *Journal of Air Transport Management*, vol. 40, pp. 169–181, 2014.
- [10] A. Bombelli and L. Tavasszy, “A pickup and delivery formulation for landside air cargo supply chain optimization.” Unpublished, 2019.
- [11] F. Basso, S. D’Amours, M. Rönnqvist, and A. Weintraub, “A survey on obstacles and difficulties of practical implementation of horizontal collaboration in logistics,” *International Transactions in Operational Research*, vol. 26, no. 3, pp. 775–793, 2019.
- [12] N. Boysen, D. Briskorn, and M. Tschöke, “Truck scheduling in cross-docking terminals with fixed outbound departures,” *OR Spectrum*, vol. 35, no. 2, pp. 479–504, 2013.
- [13] G. S. Brar and G. Saini, “Milk run logistics: literature review and directions,” in *Proceedings of the world congress on engineering*, vol. 1, pp. 6–8, WCE, 2011.
- [14] Air Cargo Netherlands, “Milkrun results after 1 year.” <https://acn.nl/milkrun/milkrun-resultaten/?lang=en>. Retrieved on: 15-10-2020.
- [15] M. Soysal, J. M. Bloemhof-Ruwaard, and T. Bekta, “The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations,” *International Journal of Production Economics*, vol. 164, pp. 366–378, 2015.
- [16] G. Laporte, “Fifty years of vehicle routing,” *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.
- [17] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bekta, and G. Laporte, “The time-dependent pollution-routing problem,” *Transportation Research Part B: Methodological*, vol. 56, pp. 265–293, 2013.

- [18] M. Barth, , T. Younglove, , and G. Scora, "Development of a heavy-duty diesel modal emissions and fuel consumption model," report, UC Berkeley: California Partners for Advanced Transportation Technology, 2005.
- [19] G. Ioannou, M. Kritikos, and G. Prastacos, "A problem generator-solver heuristic for vehicle routing with soft time windows," *Omega*, vol. 31, no. 1, pp. 41–53, 2003.
- [20] M. Roca-Riu, E. Fernández, and M. Estrada, "Parking slot assignment for urban distribution: Models and formulations," *Omega*, vol. 57, pp. 157–175, 2015.
- [21] V. Kachitvichyanukul, P. Sombuntham, and S. Kunnapapdeelert, "Benchmark problem instances for generalized multi-depot vehicle routing problems with pickup and delivery requests," in *Asia Pacific Industrial Engineering & Management Systems Conference*, pp. 290–297, 2012.
- [22] H. Li and A. Lim, "A metaheuristic for the pickup and delivery problem with time windows," *International Journal on Artificial Intelligence Tools*, vol. 12, no. 02, pp. 173–186, 2003.
- [23] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.
- [24] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [25] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992.
- [26] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," *Journal of King Saud University - Science*, vol. 22, no. 3, pp. 123–131, 2010.
- [27] S. Wang, L. Zeyuan, L. Wei, G. Ji, and J. Yang, "Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem," *SIMULATION*, vol. 91, 2015.
- [28] A. Rais, F. Alvelos, and M. S. Carvalho, "New mixed integer-programming model for the pickup-and-delivery problem with transshipment," *European Journal of Operational Research*, vol. 235, no. 3, pp. 530–539, 2014.
- [29] J. Rieck and J. Zimmermann, "A new mixed integer linear model for a rich vehicle routing problem with docking constraints," *Annals of Operations Research*, vol. 181, no. 1, pp. 337–358, 2010.
- [30] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, 2006.
- [31] R. Baldacci, N. Christofides, and A. Mingozzi, "An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts," *Mathematical Programming*, vol. 115, no. 2, pp. 351–385, 2008.
- [32] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, "An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation," *Operations research*, vol. 52, no. 5, pp. 723–738, 2004.
- [33] Z. Miao, A. Lim, and H. Ma, "Truck dock assignment problem with operational time constraint within crossdocks," *European Journal of Operational Research*, vol. 192, no. 1, pp. 105–115, 2009.
- [34] R. S. Mangoubi and D. F. X. Mathaisel, "Optimizing gate assignments at airport terminals," *Transportation Science*, vol. 19, no. 2, pp. 173–188, 1985.
- [35] J. Xu and G. Bailey, "The airport gate assignment problem: mathematical model and a tabu search algorithm," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 2001.
- [36] R. Anand, D. Aggarwal, and V. Kumar, "A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 623–635, 2017.

- [37] G. Laporte, S. Ropke, and T. Vidal, *Vehicle Routing: Problems, Methods, and Applications*, book section Chapter 4: Heuristics for the Vehicle Routing Problem, pp. 87–116. SIAM, 2014.
- [38] D. Pisinger and S. Ropke, “A general heuristic for vehicle routing problems,” *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007.
- [39] I. Altnel and T. Öncan, “A new enhancement of the clarke and wright savings heuristic for the capacitated vehicle routing problem,” *Journal of the Operational Research Society*, vol. 56, no. 8, pp. 954–961, 2005.
- [40] P. Shaw, “A new local search algorithm providing high quality solutions to vehicle routing problems,” *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.
- [41] Hillier, Frederick S. and Lieberman, Gerald J., *Introduction to operations research*. New York, NY: McGraw-Hill, tenth edition. ed., 2015.
- [42] J. F. Cordeau, G. Laporte, and A. Mercier, “A unified tabu search heuristic for vehicle routing problems with time windows,” *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.
- [43] W. P. Nanry and J. Wesley Barnes, “Solving the pickup and delivery problem with time windows using reactive tabu search,” *Transportation Research Part B: Methodological*, vol. 34, no. 2, pp. 107–121, 2000.
- [44] G. Pankratz, “A grouping genetic algorithm for the pickup and delivery problem with time windows,” *OR Spectrum*, vol. 27, no. 1, pp. 21–41, 2005.
- [45] B. M. Baker and M. A. Ayechew, “A genetic algorithm for the vehicle routing problem,” *Computers & Operations Research*, vol. 30, no. 5, pp. 787–800, 2003.
- [46] H. Psaraftis, “A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem,” *Transportation Science*, vol. 14, pp. 130–154, 1980.
- [47] M. Mahmoudi and X. Zhou, “Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on statespacetime network representations,” *Transportation Research Part B: Methodological*, vol. 89, pp. 19–42, 2016.
- [48] J. Wang, A. K. Ranganathan Jagannathan, X. Zuo, and C. C. Murray, “Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries,” *Computers & Industrial Engineering*, vol. 112, pp. 84–98, 2017.
- [49] E. Zare-Reisabadi and S. Hamid Mirmohammadi, “Site dependent vehicle routing problem with soft time window: Modeling and solution approach,” *Computers & Industrial Engineering*, vol. 90, pp. 177–185, 2015.
- [50] B. Shepherd, A. Shingal, and A. Raj, “Value of air cargo: Air transport and global value chains,” *Montreal: The International Air Transport Association (IATA)*, 2016.
- [51] K. Deb, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, book section Chapter 15: Multi-objective Optimization, pp. 403–449. Boston, MA: Springer US, 2014.
- [52] M. Iori and S. Martello, “Routing problems with loading constraints,” *TOP*, vol. 18, no. 1, pp. 4–27, 2010.
- [53] M. Iori, J.-J. Salazar-González, and D. Vigo, “An exact approach for the vehicle routing problem with two-dimensional loading constraints,” *Transportation Science*, vol. 41, no. 2, pp. 253–264, 2007.
- [54] H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg, “Vehicle routing problems with loading constraints: state-of-the-art and future directions,” *OR Spectrum*, vol. 37, no. 2, pp. 297–330, 2014.
- [55] L. Martínez and C. A. Amaya, “A vehicle routing problem with multi-trips and time windows for circular items,” *The Journal of the Operational Research Society*, vol. 64, no. 11, pp. 1630–1643, 2013.
- [56] L. Junqueira, J. F. Oliveira, M. A. Carravilla, and R. Morabito, “An optimization model for the vehicle routing problem with practical three-dimensional loading constraints,” *International Transactions in Operational Research*, vol. 20, no. 5, pp. 645–666, 2013.

- [57] S. Khebbache-Hadji, C. Prins, A. Yalaoui, and M. Reghioui, "Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows," *Central European Journal of Operations Research*, vol. 21, no. 2, pp. 307–336, 2013.
- [58] J. Cordeau, M. Iori, G. Laporte, and J. J. Salazar González, "A branch and cut algorithm for the pickup and delivery traveling salesman problem with lifo loading," *Networks*, vol. 55, no. 1, pp. 46–59, 2010.
- [59] J.-F. Côté, M. Gendreau, and J.-Y. Potvin, *Large neighborhood search for the single vehicle pickup and delivery problem with multiple loading stacks*. CIRRELT, 2009.
- [60] U. Ritzinger, J. Puchinger, and R. F. Hartl, "A survey on dynamic and stochastic vehicle routing problems," *International Journal of Production Research*, vol. 54, no. 1, pp. 215–231, 2016.
- [61] A. Toriello, W. B. Haskell, and M. Poremba, "A dynamic traveling salesman problem with stochastic arc costs," *Operations Research*, vol. 62, no. 5, pp. 1107–1125, 2014.
- [62] E. Berhan, B. Beshah, D. Kitaw, and A. Abraham, "Stochastic vehicle routing problem: A literature survey," *Journal of Information & Knowledge Management*, vol. 13, no. 03, p. 1450022, 2014.
- [63] A. M. Campbell, M. Gendreau, and B. W. Thomas, "The orienteering problem with stochastic travel and service times," *Annals of Operations Research*, vol. 186, no. 1, pp. 61–81, 2011.
- [64] C. Weskamp, A. Koberstein, F. Schwartz, L. Suhl, and S. VoSS, "A two-stage stochastic programming approach for identifying optimal postponement strategies in supply chains with uncertain demand," *Omega*, vol. 83, pp. 123–138, 2019.
- [65] F. Alonso, M. J. Alvarez, and J. E. Beasley, "A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions," *Journal of the Operational Research Society*, vol. 59, no. 7, pp. 963–976, 2008.