An Improved Optimization Method to Increase the Truck Planning Efficiency of Landside Air Cargo Operations

Master Thesis Report

T. W. Wu

TU Delft

An Improved Optimization Method to Increase the Truck Planning Efficiency of Landside Air Cargo Operations

By

Ting Wei Wu

to obtain the Master Degree of Science
in Transport, Infrastructure and Logistics
at the Delft University of Technology
to be defended publicly on 26th of August 2019 at 15:30

Thesis Committee:

Prof. dr. ir. L.A. Tavasszy	TU Delft, CEG, Transport & Planning (Chair)
Dr. Victor Knoop	TU Delft, CEG, Transport & Planning
Dr. Milan Janic	TU Delft, CEG, Transport & Planning
Dr. Bilge Atasoy	TU Delft, 3ME, Transport Engineering & Logistics
Dr. Alessandro Bombelli	TU Delft, CEG, Transport & Planning

TU Delft Delft University of Technology

Preface

This thesis is the final part to obtain my degree of Master of Science in Transport, Infrastructure and Logistics at the Delft University of Technology. It was a challenge during the thesis period, but I have learned a lot and enjoyed it at the same time. I would like to thanks the people who supported me during this period.

First and foremost, I would like to thank my graduate committee for all the support. I am very grateful to Dr. Alessandro Bombelli for all his support. We always have had great discussions about the research during the meeting. In addition, he was always positive and encouraged me at the end of the meeting, which gave me a lot of confidence. Next, I would like to thank Dr. Bilge Atasoy, for discussing the method chosen at the beginning of the thesis, discussing the structure of the thesis and for being available when I needed it. Thanks to Dr. Milan Janic for the detailed feedback to help me see the bigger picture of the thesis. My gratitude also goes out to Prof. Dr. Ir. Lóránt Tavasszy, who provided helpful feedback during every meeting. I am especially grateful to Dr. Victor Knoop for being one of my committees three weeks before graduation.

Second, I would like to thank my friends, especially my friends for making my unforgettable time in TU Delft. Thank all my friends who were working hard together in the thesis room. Because of them, I could gain different opinions for my thesis during our coffee break. I want to express my gratitude to my friends who have known python. I would not finish the meta-heuristic method without their help in the python.

Lastly, I want to thank my family for supporting me during my study years. They have given the best mental support, especially I am 13-hours flight away from them. A big thank to my brother, Shao-Fuw, for helping me to construct the meta-heuristic method with python when he was in the exam week. Without anyone of your support, I would not be possible to finish my thesis.

With pride I present to you my Master Thesis. Enjoy the read!

Ting Wei, Wu
Delft, August 2019

List of Abbreviations

ALNS	Adaptive Large Neighborhood Search
LACSC-PDPTW	Landside Air Cargo Supply Chain Pickup and Delivery Problem with Time Window
LIFO	Last-In-First-Out
LNS	Large Neighborhood Search
GA	Genetic Algorithm
MILP	Mixed Integer Linear Programming
PDPTW	Pickup Delivery Problem with Time Window
SA	Simulated Annealing
TS	Tabu Search
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Window
ULD	Unit Load Device

Summary

The growth of air freight raises challenges to the air cargo chain to different levels. Due to the lack of manpower at the ground handler's side, the cargo loading/unloading process becomes insufficient and jeopardize the timeliness of ground cargo services. In addition, uncooperative relationships between freight forwarders make them plan truck routes without considering the impact of truck belonging to others. As a consequence, trucks might experience delays when docks are occupied and further causes the truck congestion at the landside airport. Previous researches show that horizontal collaboration, the collaboration between stakeholders who are in the same level of the supply chain, leads to cost reduction in the transportation process. Bombelli and Tavasszy (2018) identify a mathematical formulation called Landside Air Cargo Supply Chain Pickup and Delivery Problem with Time Window (LACSC-PDPTW) that models the optimal transport cost solution for airport operations. In many cases, computing optimal solutions can be time consuming and pose a challenge for real-time or quasi-real-time applications. The goal of this thesis is to develop a method to reach at near-optimal solutions within a shorter time. In order to achieve this, the research question is formulated:

How can truck routes be optimized efficiently without significantly compromising the solution accuracy in the context of the air cargo supply chain?

To achieve this goal, a meta-heuristic method is developed to plan truck route in the landside air cargo supply chain. Every meta-heuristic method must satisfy the same assumptions and requirements as the associated mathematical model. The basic operation constraints that limit the solution space in the meta-heuristic method is based on the constraints of the LACSC-PDPTW model. There are five constraints in the model:

1. **Time window constraint:** shipments are only allowed to be picked up or delivered within a certain period.
2. **Total weight constraint:** the total maximum weight of all shipments that trucks can load in the assigned route.
3. **Sequence constraint 1:** trucks must follow the sequence of visiting the pickup point and then the corresponding delivery point in the assigned route.
4. **Sequence constraint 2:** truck are only allowed to visit freight forwarders and

ground handlers once in its route, which eliminate unnecessary movements.

5. **Sequence constraint 3:** trucks have to apply LIFO approach to pickup and deliver shipments.

Moreover, a dock capacity constraint is considered in the meta-heuristic method. This constraint is defined as a violation when more arrival trucks than the available docks at the same time. Time related constraints are the core for the meta-heuristic method while searching for the near-optimal solutions. Therefore, time window and dock capacity constraints are defined as the time-dependent constraint, and the total weight constraint and the sequence constraints 1 to 3 are time-independent constraints.

Simulated Annealing (SA) is chosen as the core of the algorithm, and Adaptive Large Neighborhood Search (ALNS) is used as the neighborhood search. The SA allows visiting the infeasible solution to find a global best solution, and ALNS applies the ruin and recreate principle to use different removal and insertion methods in each iteration.

To provide more flexible solutions, visiting infeasible time-independent constraints is allowed in the computation process. A small penalty is given while a time-independent constraint is violated, while time-dependent constraints are set as hard constraints which associated with sufficiently high penalties once violated. The objective function is given as follows:

$$J^* = J + \gamma_1 \sum_{i \in N} TWV_i + \gamma_2 DCV \tag{0.1}$$

Term J represents the feasible component of the objective function, which is the same cost function as the MILP model. The two additional terms represent time window and dock capacity violations, respectively. γ_1 is the violation cost per time unit if shipment i is not picked up or delivered before its latest arrival time and TWV_i is total time units that shipment i violates its time window intervals. γ_2 is the violation cost per time unit while truck k waits for an available dock at the warehouse and DCV is the overall time units where more trucks than available docks are assigned to each ground handler.

To eliminate infeasibility of the dock capacity constraint in solutions, a local improvement calls time slack strategy is applied. This strategy assigns the truck with the greater time slacks to delay its arrival time so the violation time is minimized at the ground handler's side. The second approach in the algorithm is to verify the visiting time of all the nodes upstream, starting from the origin depot. If the delayed truck can be postpone by the same time length as computed at the origin depot, the truck would not cause any dock capacity violation nor unnecessary ride time increase.

To provide a more specific assessment of the truck congestion at the landside issue, Amsterdam Schiphol Airport is chosen as a reference. The real location of 5 freight forwarders and 5 ground handlers are applied when designing computation experiments. The results for 10 instances are shown in the following table. In order to provide a benchmark time in our comparison, a time limit of 120 minutes (2 hours) is set when computing the solution of the MILP model. Three outcomes are possible for each MILP instance. (1) no solution was found within the time-limit, (2) an incumbent solution was found, with a non-zero gap optimality, and (3) the optimal solution was found (i.e., gap optimality is 0%).

The parameters for the time-dependent constraints are set equal to 100, and the dynamic probability approach is applied in the neighborhood search. In the FF column we report the freight forwarders considered in each instance, while in the GH column we report the associated ground handlers.

Table 0.1: Computational result of 10 instances in Schiphol Airport

Instances	**Total shipments**	**FF**	**GH**	**Total cost of heuristics method**	**Computing time [mins]**	**Total cost of exact model**	**Computing time [mins]**	**Gap [%]**
1	15	F1 & F2	G1 & G5	236.8	2.3	240.6	120	-1.58
2	15	F1 & F2	G3 to G5	203.5	1.5	203.9	120	-0.17
3	20	F3 & F4	G1 to G4	315.0	3.3	309.1	120	1.91
4	25	F4 & F5	G1 to G5	397.7	6.2	386.7	120	2.83
5	20	F3 to F5	G4 & G5	279.0	2.9	279.9	120	-0.35
6	30	F3 to F5	G3 to G5	428.4	10.6	432.3	120	-0.91
7	35	F1 to F3	G2 to G4	564.8	15.2	533.6	120	5.86
8	40	F1 to F3	G1 to G4	668.5	20.5	N/A	480	-
9	50	F1 to F5	G1 to G5	865.2	44.8	N/A	480	-
10	55	F1 to F5	G1 to G5	927.4	54.9	N/A	480	-

The result is evident that the meta-heuristic method is effective in producing comparable or even better solutions than the exact model in a shorter time. Out of 10 instances, the meta-heuristic method arrive at better solutions in 5 cases. In addition, the meta-heuristic method is able to find out feasible solutions where the exact model failed to deliver.

In order to apply the method in practice, a study about the location of the central depot should be carried out. Here, the location of the central depot is assumed to be a building-free area. Furthermore, a study about the monetary benefit to the stakeholders is needed so the applicable airport is determined.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, the purpose and the content of this thesis are discussed. Background information about the air cargo supply chain is given in the first section. An overview of the problems of an airport is illustrated in Sec. 1.2. Section 1.3 discusses some of the developed approaches that solve the congestion problems introduced in Sec. 1.2. Section 1.4 presents the problem statement and the corresponding research question is stated in Sec. 1.5. Lastly, the scope and the structure of the thesis are outlined.

1.1 Background

The air cargo supply chain is a complex process that involves several stakeholders cooperating together to carry out door-to-door transportation of a shipment from a shipper to a consignee (ICAO, 2016). An overview of the air cargo supply chain is given in Fig. 1.1. In general, the more actors involved in the process, the more complex the process becomes.

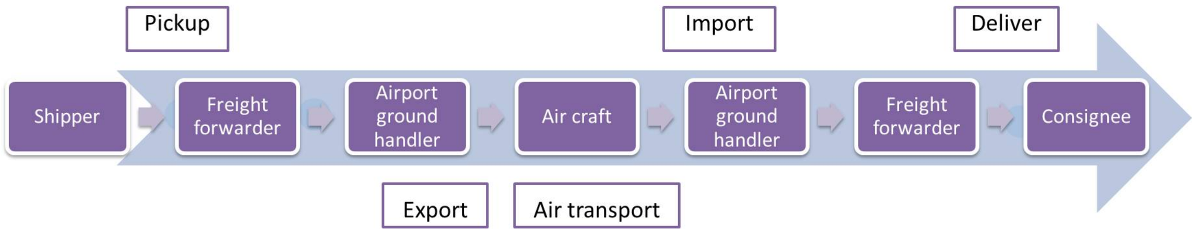

Figure 1.1: Air cargo movement overview

The air cargo supply chain is characterized by different bottlenecks and inefficiencies that might hinder the soundness of overall transportation process. In the air cargo supply chain, as an example, most documentation is still paper-based. Document verification and custom inspection processes can be time consuming therefore translate into delays in the air supply chain. This is especially the case when transporting the shipments from freight forwarders to ground handlers for export operations (Perez Bernal et al., 2012). Another bottleneck occurs at ground handler's side when arrival trucks exceed the dock capacities at the ground handler's warehouse. Because freight forwarders are of the interests to assign their own trucks and shipments to the ground handler's side first. Moreover, the lack of manpower at the ground handler's side results in inefficiency in the uploading or unloading process, which generates the service delays and truck queuing (Max, 2018).

1.2 Problem Overview in Schiphol Airport

To provide a more specific assessment of the aforementioned issues, this thesis studies the case of Amsterdam Schiphol Airport.

Schiphol Airport stated the third largest European airport in terms of air freight in 2017 (Portal, 2018). Goal of the airport is to further cement its position and to become the most preferred European airport for logistic service providers (Cargo, 2019). There were over 1.75 millions tonnes of air freight transported through Schiphol Airport in 2017 (Group, 2018). Out of which, 326 direct flights travel from Schiphol Airport to different destinations, attracting logistic service providers transporting their goods via Schiphol Airport. To better cope with the rising demand, Schiphol Airport was one of the first cargo airports to initiate E-freight and E-link projects involving paperless documentation which reduce truck waiting time and congestion at aprons.

Figure 1.2 illustrates the differences in the air cargo supply chain procedure between Schiphol Airport and other airports under the research. Thanks to the electronic documentation system, the original 30 minutes waiting time for trucks entering Schiphol Airport aprons is now significantly decreased (ACN, 2019). Trucks are now able to reach the ground handler's side once they enter Schiphol Airport aprons.

However, a potential queuing issue would still occur if there is no free dock at ground handler's side in both Schiphol Airport and the other airports. The waiting time depends on the capacity of ground handlers and the number of visiting trucks. Because of the "first come first served" principle, the truck arrival time depends on shipments flight schedule. In the worst case, trucks would wait up to several hours for an available dock during the peak periods.

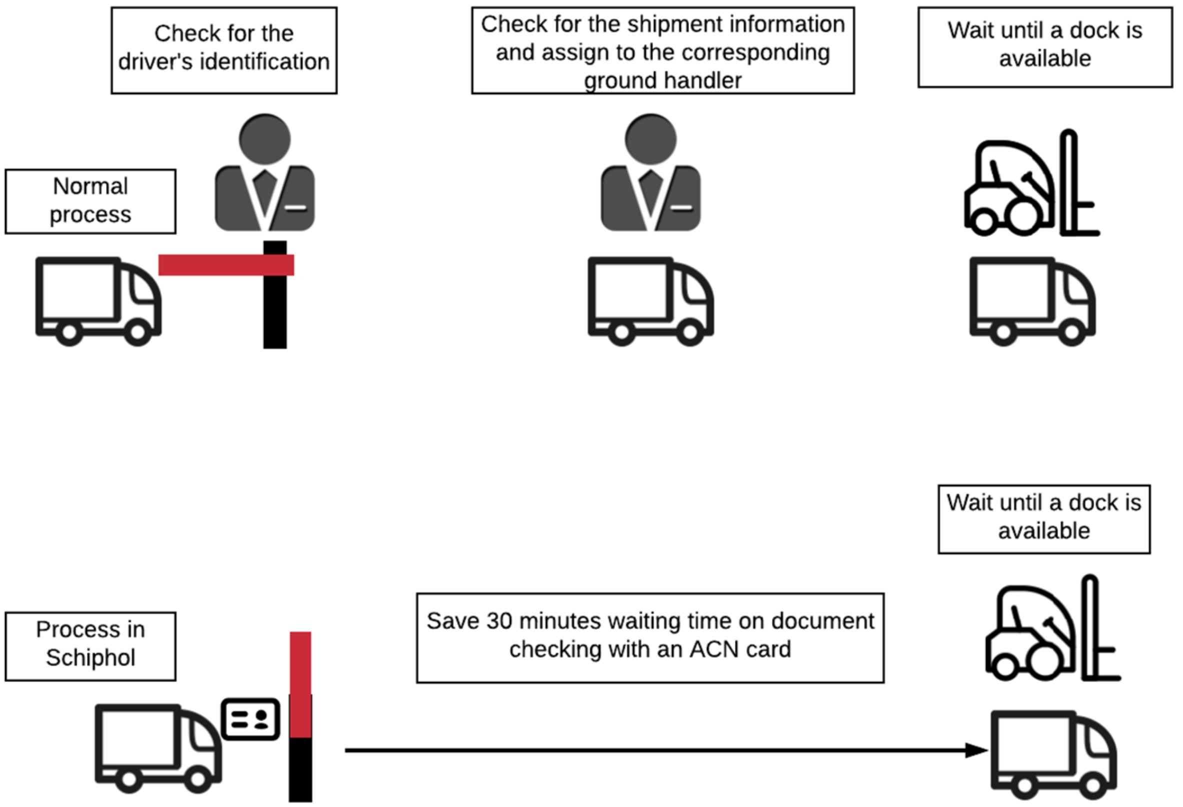

Figure 1.2: Truck queuing at the landside airport

Since ground handlers act as an intermediary between the landside and the airside in an airport, only ground handlers are able to locate in airport aprons and freight forwarders are located outside. Hence, a truck is assigned from a freight forwarder and transports export shipments to several ground handlers. While the number of air freight increases, the number of the arrival trucks is increased at the ground handler's side. When the demand (arrival trucks) is greater than the supply (available docks), the result is likely to generate truck queuing at the landside. For instance, Pieters (2014) finds out that trucks have to wait longer time for loading or unloading shipments in the weekend (Friday-Saturday and Sunday-Monday) in Schiphol Airport.

1.3 Related Work

The main advantage of air transport compares to other transport modes is time-saving. Shippers prefer to transport goods via airplane for a long distance and expecting a good quality service regarding its high transport price. There are two delay situation that creating customer dissatisfaction. An irresistible situation is when shipments are delivered too late from the shipper and further too late be boarded on the planned airline. A preventable situation is when shipments are congested at the at the landside airport because of the truck queuing. Some researches have been studied to reduce the truck congestion problem at the landside airport.

First, Ankersmit et al. (2014) propose an application of the collaborative framework between freight forwarders in the air cargo industry. Horizontal collaboration is a concept where actors in the same layer of a supply chain work together (Cruijssen et al., 2007). The researchers use a discrete-event simulation to compare the effect whether horizontal collaboration is applied in the air cargo industry or not. The result shows that freight forwarders often ship the shipments with an extensive number of transport movement in the non-collaboration scenario. In horizontal collaboration scenario, freight forwarders combine their shipments with a similar delivery time to ground handlers. This consequence improves the transportation performance of freight forwarders, such as increasing the total number of delivered shipments in a given time and the average load factor, reducing transport cost and total transport movements.

Another study that tried to shed some light on horizontal collaborative framework is called Milkrun. The name is inspired by an American milkman who does a round trip to distribute full milk bottles and collect empty ones at the same time (ACN, 2019). A freight forwarder assigns a truck to transport shipments to ground handlers in an uncoordinated transport process. A round trip is used to serve various ground handlers and freight forwarders in Milkrun project. In Milkrun project, truck routes for import shipments are assigned by a ground handler. With available pickup shipments and truck available arrival time information, the ground handler could fulfill pickup information to the truck driver and freight forwarders. Therefore, a truck can distribute import shipments to various freight forwarders in the same round. Buso (2017) used a discrete-event simulation to determine the result when applying the Milkrun concept to import of loose cargo at Schiphol Airport. The outcome shows 30% of CO_2 emission is reduced at the landside and load factor is improved in a truck.

Bombelli and Tavasszy (2018) designs a Landside Air Cargo Supply Chain Pickup and Delivery Problem with Time Window (LACSC-PDPTW) model to compute optimal routes with horizontal collaboration in the landside air cargo supply chain. The LACSC-PDPTW model is posed as a Mixed Integer Linear Programming (MILP) problem and solved exactly for small instances. A comparison of non-collaborative and collaborative scenarios is addressed by the exact model. Compare to the non-collaborative scenario, the result of the collaborative scenario shows a 10% reduction of transportation cost and a 25% reduction of fleet size.

The above studies all include the concept of horizontal collaboration to solve the truck congestion at the landside airport. To apply theoretical approaches into the real world, a chosen model should be discussed. A discrete-event simulation is used to learn uncertain situations by providing a response to different operating conditions. After understanding the impact, the operator decides whether to apply it. However, it is unable to validate results through the simulation. The MILP model is used to compute optimal solutions. It considers most of factors of the real world and is able to evaluate the result. However, it is difficult to search the optimal result for NP-hard problem in a short time. Table 1.1 summarizes the advantages and disadvantages of these theoretical researches.

Table 1.1: Advantage and disadvantage of relevant studies

<table>
<tr><td>Study</td><td>Ankersmit et al.</td><td>Buso</td><td>Bombelli and Tavasszy</td></tr>
<tr><td>Methodology</td><td colspan="2">discrete-event simulation</td><td>exact method</td></tr>
<tr><td>Advantage</td><td colspan="2">1. Forecast uncertain situation
2. Easy to understand
3. Learn the effect in a similar environment</td><td>1. Compute the optimal solution
2. Consider most of the factors
3. Be able to validate the method</td></tr>
<tr><td>Disadvantage</td><td colspan="2">1. Provide a response to different operation conditions
2. Good theories are needed
3. Unable to consider every factor
4. Difficult to validate</td><td>1. Compute for a long time
2. Not appropriate for NP-hard problem</td></tr>
</table>

1.4 Problem Statement

The growth of the air freight raises challenges to the air cargo supply chain to different levels. In particular, for landside operations, cargo loading/unloading at ground handler's side has been characterized by congestion issues. One of the reasons behind the truck congestion is uncooperative relationship between freight forwarders. Forwarders plan truck routes without considering the impact of truck belonging to other forwarders. As a consequence, trucks might experience delays when docks are occupied. This could cause snowball effects on the whole delivery tour. Previous researches have shown that horizontal collaboration, i.e., the collaboration between stakeholders belonging to the same level of the supply chain, is beneficial in reducing transportation costs. To the best of our knowledge, there has no previous study specifically focused on landside air cargo operations, and our goal is to study this research gap.

In Bombelli and Tavasszy (2018) research, a mathematical formulation, called LACSC-PDPTW, was described to model horizontal collaboration between forwarders for landside operations. Apart from small instances, the computational time to obtain optimal solutions poses challenges, which make the exact formulation not appropriate for real-time or quasi-real-time applications. The goal of this thesis is to develop a method to reach good-quality solutions of the LACSC-PDPTW with a limited computational effort.

1.5 Objective and Research Question

The main objective of this thesis is to explore a heuristic method while horizontal collaboration is applied among freight forwarders. To reach the objective, the research question is proposed as follows:

How can truck routes be optimized efficiently without significantly compromising the solution accuracy in the context of the air cargo supply chain?

To answer the research question, following three sub research questions are proposed:

1. What causes the truck congestion problem between air- and land side handlers and who are the related actors?

   The sub research question 1 investigates a clear understanding of the target issue and its associated actors to design a model that can be used at the airport. Furthermore, it is an input for some of the assumption/rules of the model.

2. What are the feasible efficient methods and which is suitable to the collaborative pickup and delivery problem in this thesis?

   Sub research question 2 looks at the feasible efficient methods and adopt the suitable method to apply in this thesis.

3. How to reduce the time-dependent violation when applying the efficient method?

   Besides designing an efficient method in the thesis, another objective of the research is to eliminate the truck congestion in airport operations. This targets a reduction of truck waiting time at the ground handler's side, therefore, sub research question 3 investigates how to reduce the time-dependent violation in the model.

1.6 Research Scope

The truck route can plan to pickup export shipments from freight forwarders and then deliver to ground handlers or pickup import shipments from ground handlers and then deliver to freight forwarders. In this research, however, it is considered export service first due to the strict schedule of flights. When a congestion happens on the ground handler's side in the export process, shipments easily miss their scheduled flight. Compared with airlines, freight forwarders have more flexible schedules when receiving goods. In addition, it is convenient to change the model of truck schedule for import cargo.

In order to compare how efficiency is the efficient method, the benchmark is Bombelli and Tavasszy (2018) research. They take Schiphol Airport as the case study, same as this research. However, the meta-heuristic method can apply in other airports with the similar problem.

1.7 Research approach

The visualization of the thesis structure is shown in Fig. 1.3. Chapter 1 illustrates the purpose and the objective of the research. Chapter 2 discusses MILP formulations for logistics and the suitable meta-heuristic method for this problem. The problem and stakeholders descriptions are presented in Chap. 3. Chapter 4 illustrates the solution approach of the meta-heuristic method. Chapter 5 shows computational results for different case studies. Lastly, conclusions and recommendations are drawn in Chap. 6.

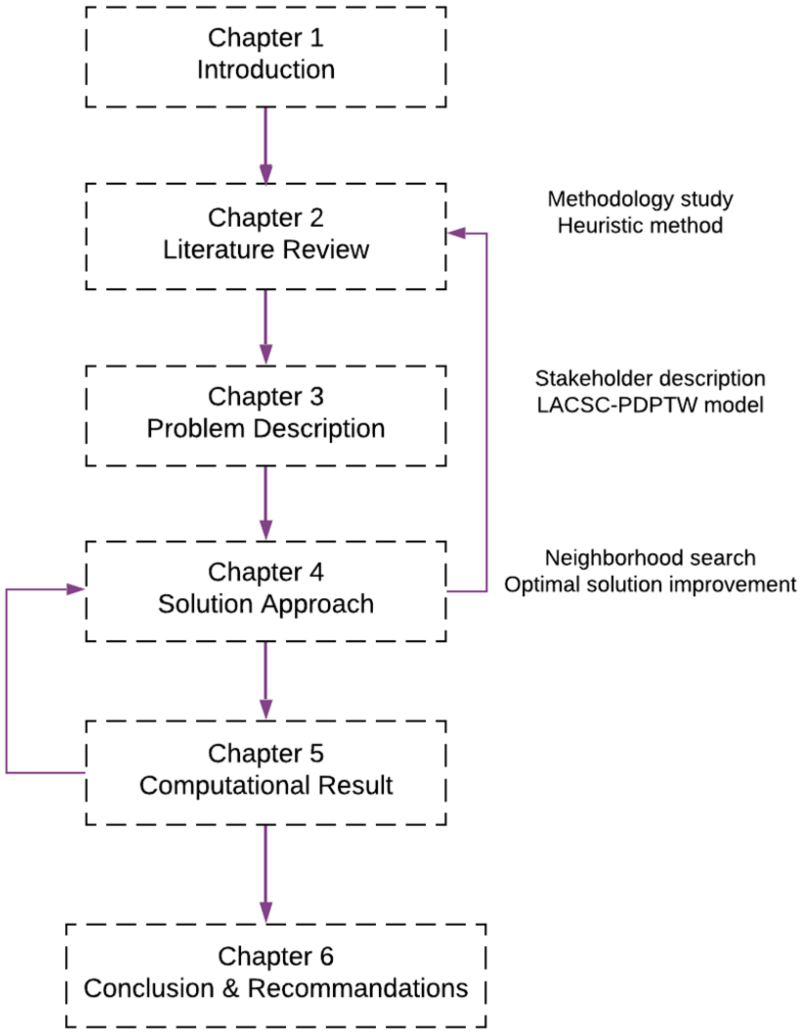

Figure 1.3: Thesis flow diagram

2 | Literature Review

As stated in Sec. 1.4, the goal of this thesis is to develop of a meta-heuristic method to plan truck routing in the landside air cargo supply chain. Every meta-heuristic method must satisfy the same assumptions and requirements as the associated mathematical method. Therefore, Sec. 2.1 provides an overview of mathematical models addressing vehicle routing in logistics, while Sec. 2.2 discusses meta-heuristic methods.

2.1 MILP Formulations for Vehicle Routing in Logistics

Truck routing planning is a problem of paramount importance in logistics. Although different methods can be used to address this problem, approaches based on MILP formulations are considered as the gold standard. In fact, they can compute an optimal solution, i.e., a best lower bound that can be used for benchmarking, as example.

The Vehicle Routing Problem (VRP) is proposed to solve the routing problem for a fleet of vehicles to collect pickup or delivery goods at their corresponding request nodes (Savelsbergh, 1985). In general, vehicles leave the depot and deliver shipments to their demand nodes or pickup shipments from their supply nodes and return to the depot. In a similar fashion, in the landside air cargo supply chain, trucks perform a pickup tour among freight forwarders, and then a delivery tour among ground handlers. A major difference is that, due to horizontal collaboration, both tours are built progressively and not carried out in a single distribution center. Some general requirements of the VRP, and of every model inspired by the VRP, are: (1) each vehicle starts and ends at the same depot, (2) each customer is serviced exactly once by a vehicle, (3) the total capacity of a vehicle should not exceed its capacity all the time.

The Vehicle Routing Problem with Time Windows (VRPTW) was introduced as a variant of the VRP when bounds on pickup and delivery time are present. Vehicles service each customer within an associated time window and the vehicle must remain at the customer location during the service. The time window constraint can be applied either as a soft constraint or a hard constraint. The Pickup and Delivery Problem with Time Windows (PDPTW) is another important extension of the VRP, which is frequently used in logistics and the transportation industry (Li and Lim, 2003). The PDPTW is similar to the VRPTW because of capacity limitations on vehicles and on the fact that each truck should leave and go back to the same depot. However, the biggest difference is that requests are paired in the PDPTW, but not in the VRPTW and the VRP (Hosny, 2011). In other words, the VRPTW is used when all request nodes are formed by the same characteristic requests, which is either delivering shipments from the deport or picking up shipments to be returned to the depot. The PDPTW is used to assign an empty truck from the depot to pickup a shipment at the pickup node and then deliver to its corresponding delivery node then return an empty truck to the depot. Due to the paired request characteristics of the PDPTW, there is a precedence constraint. Pickup nodes must be visited before the corresponding delivery nodes (Li and Lim, 2003).

Typically, there are two approaches to solve such VRP, VRPTW or PDPTW problems. One is the exact method that obtains an optimal solution, and another is a meta-heuristic method that searches a near-optimal solution in a short time. When the size of the problem grows, the exact method generates a computational challenge. Therefore, meta-heuristic methods are proposed to reduce the computational time required to compute a near-optimal solution. As discussed in the next section, the meta-heuristic method tackle the same problem without explicitly modeling is as a MILP.

2.2 Meta-heuristic Methods for Routing Problems

Meta-heuristic methods are a viable solution to obtain sub-optimal, or even optimal solutions for models and instances where computing the exact solution of the MILP problem is computationally intractable. Most meta-heuristic methods are designed as two-stage methods. First, an initial solution is generated. Second, the solution is refined/improved via a neighborhood search. The most common neighborhood search moves include moving one request to another route, exchanging two requests between two different routes or shifting one request within a route (Li and Lim, 2003). The goal of the neighborhood search is to drive the solution towards a global minimum (or maximum, depending on the problem specifications), by applying a set of moves at each iteration.

The most common meta-heuristic methods are Tabu Search (TS), Simulated Annealing (SA), Adaptive Large Neighborhood Search (ALNS) and Genetic Algorithm (GA). Li and Lim (2003) apply tabu-embedded SA algorithm, which restarts the current best solution when the iterations do not improve within the SA structure, to solve the PDPTW problem. Ropke and Pisinger (2006) introduce ALNS, which is an extension from the Large Neighborhood Search (LNS), to effectively solve the PDPTW problem. ALNS apply different removal and insertion approaches in each iteration, and an adaptive weight is applied to select one of the approaches. In addition, SA is used to be a stopping criteria in the ALNS. Baños et al. (2013) use SA to solve the multi-objective VRPTW problem. Li et al. (2016a) apply ALNS to solve transport routing planning when people and parcels are sharing vehicles. Two solution evaluation approaches are used in the objective function in Li et al. (2016a) research, one only allows visiting feasible solutions and other considers infeasible solutions by adding a penalty to the constraint violation. In addition, the penalty weights for infeasible solutions are modified according to the feasibility of the solution. Li et al. (2016b) use ALNS to address collaborative logistics routes, which is so called Pickup and Delivery Problems with Time Windows, Profit and Reservation Requests (PDPTWPR).

A general description of the different heuristic methods is provided. TS applies local search as its neighborhood search outline but forbids the non-improving moves in the next iterations. The forbidden movement is stored in a list and used to avoid cycling the visiting solution from continuously moving back and forth (Cordeau and Laporte, 2003). GA requires a population of candidate solutions to be evaluated; moreover, it is often used to cluster the requests but not is not aware of how they are routed (Hosny, 2011). ALNS applies the ruin and recreate principle but uses different neighborhood searches in different iterations to find a better solution (Ropke and Pisinger, 2006). SA allows visiting infeasible solutions to find a global best solution, it accepts or rejects the solution according to its relative cost (Ropke and Pisinger, 2006). A comparison for solving PDPTW between GA and SA methods is carried out in Hosny (2011), where it is shown that the SA method yields a better average result than GA method. In addition, SA is proven to be faster.

3 | Problem Description

In this Chapter, a thorough analysis of the landside air cargo supply chain is performed to better characterize the problem under examination in Sec. 3.1 and 3.2. In addition, the MILP model that provides the basis for this work, i.e., the LACSC-PDPTW, will be presented and critically discussed in Sec. 3.3.

3.1 Truck Congestion Problem at the Airport Landside

Congestion, in supply chain processes, occurs whenever a specific demand exceeds the supply. As an example at the landside airport, when more trucks than the available docks visit simultaneously a warehouse for pickup and/or delivery operations (Weisbrod and Fitzroy, 2011). While congestion, in some cases, implies that shipments will be stocked later than expected, the consequences can be more severe in the air cargo supply chain. In fact, delays in export operations might cause shipments missing their chance to be loaded on the intended aircraft. Customers will be dissatisfied since the speed and reliability are the factors for them to pay the premium price to ensure delivery time windows are satisfied (Barnett, 2019).

In general, three reasons can be identified that contribute to truck congestion at the airport landside. The first reason is an unbalanced number of ground handlers and freight forwarders at an airport. For example, there are 31 ground handlers' companies and over 130 freight forwarders' companies in Schiphol Airport. A shortage of manpower cannot handle the shipments effectively (Max, 2018). Second, the uncoordinated truck schedule of each freight forwarder might cause more trucks arriving at each ground handler's side than the actual dock capacity in the peak period. Barnett (2019) found out freight forwarders ship their trucks at the same time of the week, which is unavoidable. The last reason that causes congestion at the ground handler's side is the limit space that ground handlers own in airport aprons. Ground handlers are not willing to store shipments for more than 24 hours to restrict their usage space. This provides a limit time window for freight forwarders to ship their goods.

Because of the above reasons, when the demand (incoming trucks) is greater than the supply (available docks) for export shipments, truck congestion is generated in correspondence of airport aprons.

3.2 Stakeholders Description

This section describes the practical impact of truck congestion on the different stakeholders involved in the air cargo supply chain.

3.2.1 Freight Forwarder

A freight forwarder is a person or an enterprise who organizes the cargo shipping for a shipper. Freight forwarders distinguish themselves by offering different prices, the quality of services and targeting different markets to their customers (Cargo, 2019). Because of the security issue, freight forwarders are only allowed to have warehouses near an airport but not inside. This makes them regarded as second line companies in an airport.

There are two reasons why freight forwarders have an interest in reducing truck congestion at the airport landside. First, shipments might miss their original scheduled flights because of truck queuing on the ground handler's side. Once shipments miss their flights, forwarders need to reschedule the flight and inform their customers, both shippers and consignees. This might hinder their reputation and cause shippers to choose different forwarders for future services. Second, forwarders are not willing to pay truck drivers to wait in a line (Barnett, 2019), which translates into an additional monetary loss.

3.2.2 Airline

There are two types of aircraft transport cargo from the origin to the destination airport, one is a passenger aircraft and another is a freight aircraft. A full freight aircraft can be a bit more flexible in the departure time while a passenger aircraft cannot. Passengers have the priority and airlines do not want to lose their slot if a delay occurs. Thus, shipments will not be loaded if they arrive at the ground handler's side late. A full freight aircraft has to wait if trucks are congested and cargo cannot be loaded onto the aircraft due to the agreements with freight forwarders. The extra waiting time brings extra costs and delays to the airline.

3.2.3 Shipper and Consignee

A shipper is a person or a company who owns the commodities shipped to a consignee. Both shippers and consignees expect air freight transport to be time-saving and reliable.

When the truck congestion causes delays, there is a chance that the consignee will not receive goods within the intended time-frame. This result in a reduced trust towards the shipper, even if the later might not be explicitly responsible for the missed delivery, the airline and the freight forwarder (if known). The consignee might lose his reputation or even his customer if the shipment needs to be delivered to a third party.

3.2.4 Ground Handler

Ground handlers are specialized service providers who take care of all operations that connect landside and airside in an airport. More specifically, they are in charge of aircraft operations between the arrival at a terminal gate and the next departure time (Gomez and Scholz, 2009). The services include ground operations, cargo & terminal facility planning, baggage service and aircraft turnaround management. Only ground handlers who service air freight are involved in this research. After receiving shipments from freight forwarders, ground handlers build up the shipments into ULDs and then load them onto the plane. Once they receive shipments from an aircraft for import operations, they break down ULDs and load the shipments onto outbound trucks. This process characterizes ground handler as the bridge that connects a freight forwarder and an aircraft in the cargo delivery.

When the truck congestion at landside airport generates the delay shipments, customers would not be reliable to the service. Shipper and consignee might be willing to switch to different airport if delays are frequent and hence, a tangible loss can be identified as a result of these delays (Aircargo news, 2018). Therefore, airlines would shift their assets and flights to neighborhood airports to guarantee a better service. For example, United Airlines moved their major New York operations from John F. Kennedy International Airport (JFK) to Newark Liberty International Airport in New Jersey because of JFK congestion problem (Barnett, 2019). Ground handlers might lose their profit when airlines move their assets to another airport.

3.3 LACSC-PDPTW Model

Horizontal collaboration between a consortium of freight forwarders is modeled via a shared fleet of trucks (Agarwal et al., 2009). A central planner, with information regarding weight, destination, time windows of shipments, computes the optimal routing strategy that also accounts for dock capacity on the ground handler's side. By knowing what the other trucks do in advance, there should be no unexpected delays/queues (although trucks might still be required to wait a bit, but they know that in advance).

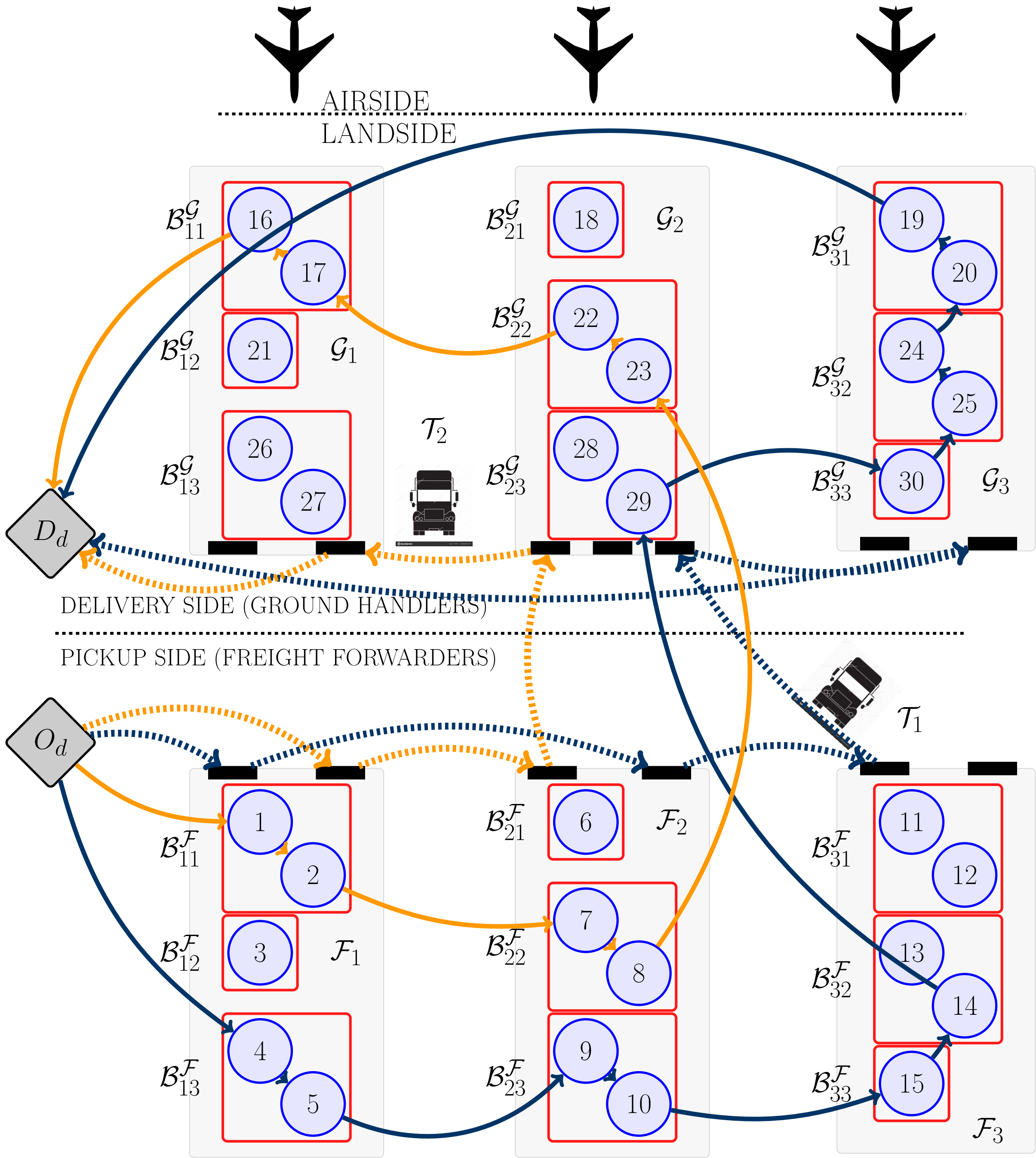

Figure 3.1: LACSC-PDPTW framework (reference: Bombelli and Tavasszy (2018))

Figure 3.1 displays the example of the truck's optimal route in Bombelli and Tavasszy (2018) research. The gray diamonds O_d and D_d represent the central depot, which may be the same geographic location in an airport. The solid line demonstrates the pickup and delivery order of the shipments and the dashed line presents the truck route to the corresponding docks while picking up/delivering the shipments. In addition, the blue line represents truck 1 route and the yellow line represents truck 2 route.

The problem contains σ requests and k vehicles, which is based on a direct graph $G = (N, E)$ where $N = O_d \cup D_d \cup N_P \cup N_D$ and E is the set of edges. Subsets O_d and D_d represent the origin depot and the destination depot, which could be the same geographic location in reality. N_P is a set of pickup nodes and N_D is a set of delivery nodes. As σ is a number of shipments to be delivered, total number of N is $2\sigma + 2$. O_d is node 0, pickup nodes $N_P = \{1, ..., \sigma\}$, delivery nodes $N_D = \{\sigma + 1, ..., 2\sigma\}$ and the destination node D_d is node $2\sigma + 1$. K is a set of vehicles, which $k \in K$.

A set of freight forwarders N_F and ground handlers N_G are considered in the model, which are grouped into blocks (Figure 3.1). The notation of freight forwarders is $F = F_1, F_2, ..., F_{N_F}$ while ground handlers is $G = G_1, G_2, ..., G_{N_G}$ in the model. Each shipment at freight forwarder N_F is characterized in a block B^F_{ij}, which contains the information about the shipment that has to be picked up from the freight forwarder i and then delivered to the ground handler j. Meanwhile, each shipment at ground handler N_G is characterized in block a B^G_{ij}, meaning that the shipment is delivered from the freight forwarder j to the ground handler i.

For each $(i, j) \in E$ is assigned a distance $d_{ij} \geq 0$ and a travel time $t_{ij} \geq 0$. In addition, the travel time between node i and j is defined as $t_{ij} = t'_{ij} + T_c$. t'_{ij} is the travel time between node i and j, while T_c is a fixed additional time if node i and j belong to different warehouses. An additional check-in time that truck drivers need to spend on document checking procedure if visiting different warehouses. Each shipment request i has its own service time s_i and a time window $[e_i, l_i]$ for both pickup and delivery procedure. s_i indicates the procedure time for loading or unloading request i in the warehouse, and the time window represents the earliest and the latest time for request i to be picked up or delivered at the warehouse. For instance, the truck driver is allowed to arrive at the warehouse earlier than e_i but has to wait until e_i for picking up the shipment i, and he is not allowed to arrive at the warehouse later than l_i. The capacity of truck $k \in K$ is set by a maximum weight capacity Q_i

The goal of the LACSC-PDPTW model is to maximize the number of shipments delivering on time while minimizing transportation costs at horizontal collaboration landside shipping. The objective function and the set of variables and parameters, as given by Bombelli and Tavasszy (2018), are as follow:

- X^k_{ij}: binary variable is 1 if truck k goes from node i to node j, 0 otherwise
- Z_i: binary variable is 1 if shipment i is not picked up by any truck, 0 otherwise
- $\tau^k_{D_d}$: a continuous time variable when truck k leaves the origin depot
- $\tau^k_{O_d}$: a continuous time variable when truck k goes back to the destination depot
- d_{ij}: distance from node i to node j
- C_d: the transport cost per distance unit

- C_τ: transport cost per time unit
- α, β, γ: the parameters that control the relative importance of different terms

$$Min:\ \alpha \sum_{i \in N} Z_i + \beta C_t \sum_{k \in K} \sum_{(i,j) \in E} d_{ij} X_{ij}^k + \gamma C_\tau \sum_{k \in K} (\tau_{D_d}^k - \tau_{O_d}^k) \tag{3.1}$$

Some operational constraints are added to limit the solution space the model can explore, and are consistent with logistics requirements. For instance, a Last-In-First-Out (LIFO) approach is used for the delivery tour to avoid the potential unnecessary unloading and reloading of shipments. This is particularly relevant if some shipments have already consolidated into the ULD at freight forwarder's side. LIFO approach is the only viable solution to avoid unnecessary operations if the lateral occupancy of shipments (e.g., ULDs) makes it difficult to reach shipments loaded further down inside the trailer. All five constraints are shown in the following:

1. **Time window constraint:** shipments are only allowed to be picked up or delivered within a certain period, especially for export cargo so the ground handler has sufficient time to consolidate and load them onto the airplane.
2. **Total weight constraint:** the total maximum weight of all shipments that truck can load in the assigned route.
3. **Sequence constraint 1:** trucks must follow the sequence of visiting pickup points and then the corresponding delivery points in the assigned route.
4. **Sequence constraint 2:** trucks are only allowed to visit the freight forwarder and the ground handler once in their routes. This sequence constraint decreases the unnecessary movement for truck driving back and forward to the same warehouse.
5. **Sequence constraint 3:** trucks apply LIFO approach to pickup and deliver shipments.

Apart from small instances, an optimal solution cannot be computed by the LACSC-PDPTW within a reasonable computational time. Ideally, the central planner would process each morning (or every few hours) shipment information and output the optimal routing schedule. Therefore, a heuristic method is applied to search a near-optimal solution in a reasonable time. To compute the same case study in the heuristic method, it needs to follow the above LACSC-PDPTW constraints. To eliminate the truck waiting, the time window constraint is seen as the time-dependent constraint; the total weight constraint and the sequence constraints 1 to 3 are seen as the time-independent constraints.

4 | Solution Approach

The core of the chosen meta-heuristic method in this thesis is SA and the neighborhood search is based on the ALNS method described by Ropke and Pisinger (2006). This chapter present the detailed description of SA, neighborhood search and how to decrease truck waiting time at airport aprons.

Section 4.1 presents the structure of the solution approach and Sec. 1 illustrates the SA-embedded ALNS algorithm. Section 4.3 illustrates the objective function of the algorithm. In Sec. 4.4, a brief description of how to generate the initial solution is given. Section 4.5 describes detailed neighborhood approaches. Section 4.6 and 4.7 illustrate two methodologies to eliminate dock capacity violations on the ground handler's side and reduce the waiting time trucks might incur into.

4.1 Structure of the Solution Approach

The steps of the heuristic method are presented in Fig. 4.1. Within the SA framework, ALNS is applied to carry out the neighborhood search. An initial temperature T and a maximum iteration L are set at the beginning, and then $s_{current}$ is produced by an initial solution.

Once the initial solution is created, a new solution s' can be generated by the ALNS neighborhood search and then compute the feasibility of its time-independent constraints. The constraints of the meta-heuristic method follow the LACSC-PDPTW model, where time windows are time-dependent constraints and the others are time-independent constraints. In addition, a dock capacity constraint is added as another time-dependent constraint. The dock capacity constraint is defined as a violation when more arrival trucks than the available dock at the same time.

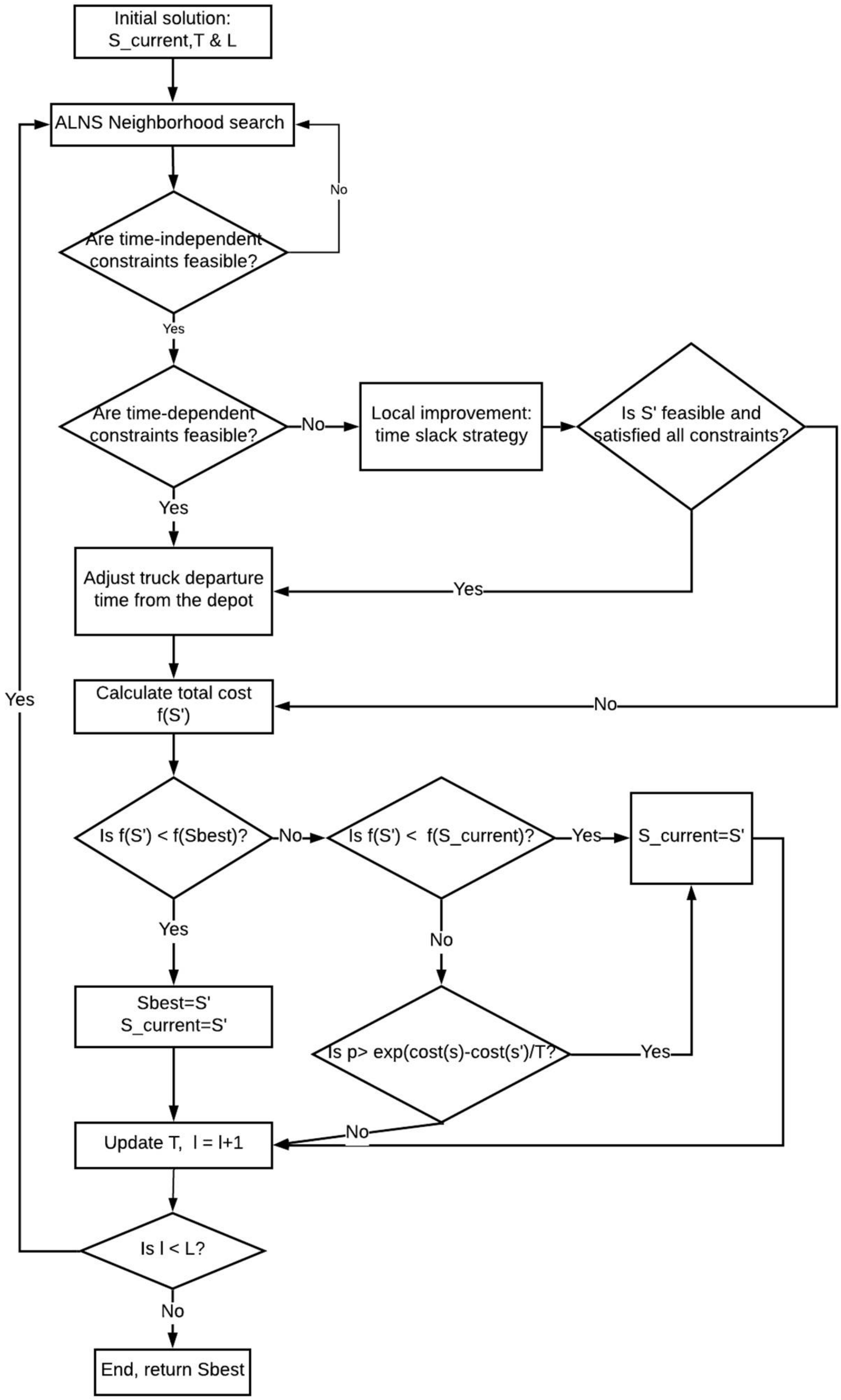

Figure 4.1: Solution approach flow diagram

The time-independent constraints include: total weight constraint and the truck visiting sequence constraints. Every time-independent constraint is a hard constraint that associated a sufficiently high penalty when violate. As an example, if a truck route is designed where the total shipments weight is more than the upper limit, no recovery action is possible for such the infeasibility to resolve. To accept an infeasible solution in the time-dependent constraints, a smaller penalty is used when violating. The idea of accepting these time-dependent constraints provides greater flexibility in finding the best solution. As shown in Sec. 2.2, a proper weight must be tuned when accepting this infeasible contribution, such that the final best solution is feasible. In the rest of the thesis, when mentioning infeasible solutions, it is referring to solutions where only time window or dock capacity constraints are violated. Infeasible solutions of any other kind are immediately discarded instead.

To avoid or reduce as much as possible infeasibility in infeasible solutions, a local improvement time slack strategy is applied. When there is a dock capacity violation, a truck with the greater time slack is assigned to delay its arrival time at the ground handler's side so trucks wait less or do not wait. For the truck that has just been delayed, an increase in ride time will be generated that is equal to the delay if no further action is taken. The second step in our routine checks the visit time of all the nodes upstream, starting from the origin depot. If the departure of the delayed truck from the origin depot can be delayed by the same quantity as the computed delay in the time slack strategy, the truck will not cause any dock capacity violation and, at the same time, will not increase its ride time. On the other hand, if this is not possible because of possible conflicts at previous ground handlers or time window violations upstream, the starting time from the origin depot will be delayed as much as possible, thus minimizing the increase in ride time.

If a new solution is feasible, the total cost is computed. The new solution will replace the current best solution if the total cost of new solution is lesser than the current best one. In order to explore the global solution, a worse solution is accepted with a certain probability that depends on the characteristics of the SA framework. After each iteration, the temperature T that controls the acceptance rate of worse solutions will be updated and the model will not stop until a stopping criterion is met.

4.2 The SA framework

SA is used to search the global optimum of a given function. The approach is described in Algorithm 1, where an initial temperature T, maximum iteration and solution are set at the beginning. A new solution is calculated after the neighborhood search is done, and then it will compare with the current solution. The new solution will replace the current solution if the new solution is better, otherwise it will be accepted as the new solution within a certain probability (Hosny, 2011; Wang et al., 2015; Afifi et al., 2013).

Algorithm 1 Simulated Annealing

procedure SIMULATED ANNEALING
  initial solution $s_{best} = s$, initial temperature $T > 0$, maximum iteration L;
  while $l < L$ **do**
    s'_k =local search(s_k) ▷ $s'_k \in N(s_k)$
    if cost(s'_k) < cost (s_k) **then**
      $s'_k = s_k$; update candidate solution $s_k \leftarrow s'_k$
    if cost(s'_k) > cost (s_k) **then**
      calculate probability p=exp(($cost(s_k) - cost(s'_k)$)/T)
      if $u \leq p$, $s_k \leftarrow s'_k$ ▷ u ∼ U(0,1)
    update temperature T
    $l \leftarrow l + 1$
  return s_{best}

4.3 Objective Function

As the time-independent constraints are the hard constraints, the solution is discarded while a time-independent constraint violates. To provide more flexibility of the solution, a penalty is given in the objective function when the time-dependent constraint is violated. The objective function is the minimum costs of the feasible solutions and the infeasible solutions. The objective function is defined as follows:

$$J^* = J + \gamma_1 \sum_{i \in N} TWV_i + \gamma_2 DCV \tag{4.1}$$

Term J represents the feasible component of the objective function (see equation 4.2), which is the same cost function as the MILP model. On the other hand, two additional terms represent time window and dock capacity violations, respectively. γ_1 is the violation cost per time unit if shipment i is not picked up or delivered before its latest arrival time. TWV_i is total time units that shipment i violates its time window interval. γ_2 is the violation cost per time unit while arrival truck k violates available docks at a warehouse. DCV is the overall time units where more trucks than the available docks are assigned to each ground handler.

The feasible solution J includes three different costs: (1) travel time cost of each truck (2) travel distance cost of each truck (3) unloaded shipments cost.

$$J = \sum_{k \in K} \sum_{(i,j) \in E} (Cd_{ij} D^k_{ij}) + \sum_{k \in K} \sum_{(i,j) \in E} (Ct_{ij} T^k_{ij}) + \alpha \sum_{i \in N} Z_i \tag{4.2}$$

The set of parameters in the objective function is described below:

- Cd_{ij} : travel distance cost from node i to node j
- Ct_{ij} : travel time cost from node i to node j
- α : cost that shipment i is not picked up by any truck

The set of notation in the objective function is provided as:

- D_{ij}^k : the distance that truck k travels from node i to node j
- T_{ij}^k : the travel time that truck k spends from node i to node j
- Z_i : a binary variable is 1 means shipment i is not picked up by any truck, 0 otherwise

Adaptive weight of parameters of the infeasible solutions

An adaptive weight is applied to the parameters of the time-dependent constraint. A factor $1+\theta$ is introduced to modify the weight of r_1, r_2, where $0 < \theta \leq 1$. It is applied after the new solution is computed. If there is no time-dependent constraint violated in the new solution, r_1, r_2 are divided by $1+\theta$; otherwise r_1, r_2 are multiplied $1+\theta$.

Notations

The model follows the same constraints that were described in Sec. 3.3: time window of a shipment, total weight and truck visiting sequence constraints. In order to eliminate time-dependent constraints at warehouses, notations that indicate the time stamp of shipments are given as follows:

- A_i^k: the arrival time that truck k at node i
- W_i^k: the waiting time that truck k serves shipment i
- B_i^k: the beginning time that truck i starts to serve shipment i
- D_i^k: the departure time once truck i finishes the serving process of shipment i and leaves the warehouse
- P_i: the procedure time that shipment i is needed before uploading or unloading

Shipment i is not allowed to violate its earliest service time, but truck k is allowed to arrive earlier than the earliest service time e_i. A waiting time is computed as $W_i^k = max\{0, e_i - A_i^k\}$. B_i^k represents the beginning of service time, which is set as the sum of A_i^k and W_i^k. D_i^k represents the departure time of truck k, which is the sum of B_i^k and P_i.

4.4 Initial Solution

An initial solution is constructed by a *basic greedy insertion*, which is detailed described in Sec. 4.5. The minimum objective in the initial solution is J^*. We assign all shipments in an unassigned shipment list, and build solution by iteratively inserting nodes from the unassigned shipment list into a route. When no more shipments can be added in the route, because one of the time-independent constraints would be violated, a new route is created.

Figure 4.2 displays the example of the available positions that shipment k can insert into the exist route. Initially shipment k can be inserted into three potential positions: before node i, between node i and j or after node j. After computing the least cost of the insertion position, shipment k is selected to be picked up in the middle of node i and node j in this example. If the shipment could not fit in the current route, a new truck is assigned to handle this shipment. The insertion process is finished when no shipment left in the unassigned shipment list. Once the initial solution is created, a neighborhood search can be processed.

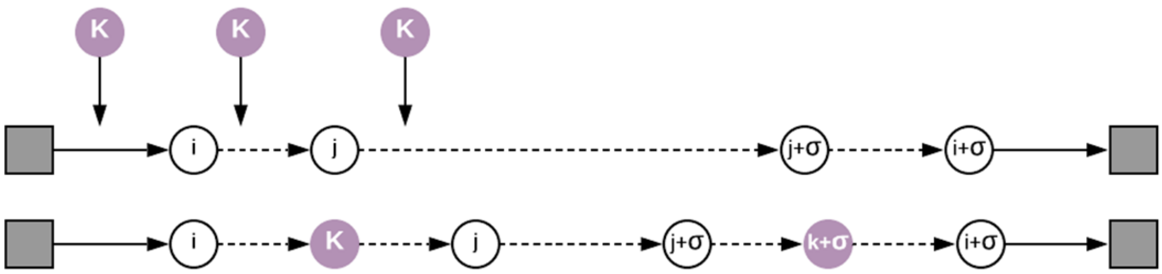

Figure 4.2: A paired pickup-delivery shipment is inserted in the current route

4.5 Methodology of ALNS

The route planning in this thesis is computed with paired pickup-delivery customers. The neighborhood search is based on the ALNS which described by Ropke and Pisinger (2006) and Azi et al. (2014). Four different removal and insertion approaches describe in the following sections. The number of removal requests is determined by the parameter $r \in \{0, ..., N\}$ and stored into the unassigned shipment list. The number of r decreases during the insertion process. When r is equal to zero, it means that there is no request waiting for the insertion and no search will be performed. Shaw removal, random removal, worst removal and the shortest route removal are introduced in the removal approach. Basic greed insertion, regret insertion, regret-k insertion and new route insertion are discussed in the insertion approach.

4.5.1 Removal Approach

Shaw removal.

The Shaw removal randomly chooses a paired request and then the requests with similar characteristics, such as distance (a request that pickup and delivery nodes are close to the seed request), time window (a request that has similar start time with the seed request at pickup and delivery nodes) or other factors. It is easier to shuffle similar requests around and creates a new or even a better solution. If the chosen requests are very different from each other, it is possible to gain nothing while reinserting those requests.

A relatedness measure $R(i, j)$ is used to search the similarity between the chosen request i and request j. Four parameters are included in the relatedness measurement calculation:

1. pickup and delivery distance difference between the chosen request i and the request j
2. time window difference between the chosen request i and the request j
3. the available weight to serve the chosen request i and the request j
4. the time slack difference that the chosen request i and the request j have at the warehouse

The weights of different parameters are α_d, β_t, γ_w and δ_{st}, respectively. The algorithm of relatedness measure is given as follows:

$$\begin{aligned} R(i,j) = \alpha_d(|d_{P(i)} - d_{P(i)}| + |d_{D(i)} - d_{D(j)}|) + \beta_t(|Te_{P(i)} - Te_{P(j)}| + \\ |Ta_{P(i)} - Ta_{P(j)}| + |Te_{D(i)} - Te_{D(j)}| + |Ta_{D(i)} - Ta_{D(j)}|) + \\ \gamma_w|W_i - W_j| + \delta_{st}(|ST_{P(i)} - ST_{P(i)}| + |ST_{P(i)} - ST_{P(i)}|) \end{aligned} \tag{4.3}$$

$P(i)$ and $D(i)$ indicate pickup and delivery locations of request i, Te denotes the earliest arrival time and Ta is the last arrival time. Hence, $Te_{P(i)}$ is the earliest arrival time of node i at its pickup warehouse, while $Ta_{P(i)}$ is the last arrival time of node i at its pickup warehouse. W_i is the weight of request i and $ST_{P(i)}$ is the time slack of request i at its pickup warehouse. $R(i, j)$ is firstly sorted sequential in an array L, and then employed a random selection to choose the number of related request N_j by the following method (Ropke and Pisinger, 2006).

$$N_j = N_j \cup L[y^p|L|] \tag{4.4}$$

A user-defined parameter $p \geq 1$ is used to randomly select the related requests in the list $R(i, j)$. When p has a low value, it introduces more randomness in the selection. On the other hand, for a high value of p, the most related shipment with respect to i is chosen.

Random removal.

The random removal algorithm is a simple method that randomly selects r requests and removes them from the current solution of s to a better position. The steps follow by the *Shaw removal* but p is set equal to 1 to achieve a totally random selection. This method is faster than other methods and speeds up the removal of requests process.

Worst removal.

The worst removal is used to search the highest decrease while the request i is not served in the solution s. The cost without request i is given as:

$$cost(i, s) = f(s) - f_{-i}(s) \tag{4.5}$$

$f_{-i}(s)$ is the cost while the solution without request i. The selected requests and the costs $\{i, c\}$ are placed by an increasing order. It seems reasonable to remove the request that locates the position with the highest cost to other lower cost position, so the highest cost request i is selected.

Shortest route removal

The shortest route removal, as the name suggests, removes the shortest route (intended as the route with fewer shipments) from the current solution to the unassigned shipment list. This approach is added to verify if a better solution can be achieved that uses fewer trucks.

4.5.2 Insertion Approach

Basic greedy insertion.

The basic greedy insertion does at most n iteration when inserting the request i into different positions k of a route. The inserted position is chosen at the smallest cost increase $\Delta f_{i,k}$ in the route. The formula is given as follow,

$$\Delta f_{i,k} = f(s') - f_{-i}(s') \tag{4.6}$$

If the request i cannot be inserted into position, it will stay in the unassigned shipment list. The process stops until all unassigned shipments have been inserted or no more requests can be inserted into any route.

Regret insertion.

The regret insertion is used to improve the basic greedy insertion because it only considers the least change after request i is inserted in the position k. The regret insertion looks further information by computing a regret value. The regret value calculates the difference for a request insertion cost between its best position and its second best position. From the basic greedy insertion knowing every $\Delta f_{i,k}$ and sorting them in order, that is, $\Delta f_{i,k} < \Delta f_{i,k'}$ where $k < k'$. The regret value R_i is the request i that inserts at the position k where has the maximum regret value but minimum cost of $\Delta f_{i,k}$. The algorithm 4.7 shows the formula of the regret value.

$$R_i = \{\Delta f_{i,k'} - \Delta f_{i,k}\} \tag{4.7}$$

Regret-k insertion.

Regret-k insertion is an extension approach from the regret insertion approach that calculates the regret value among other. Instead of calculating the regret value between the best position and the second best position of the request i, it calculates the regret value R_i^* between the solution with the least cost and the $k-1$ best solutions (equation 4.8).

$$\max_{i \in N}\{\sum_{k=2}^{k} \Delta f_{i,k} - \Delta f_{i,1}\} \tag{4.8}$$

The request i is inserted at the minimum cost position while the maximum regret value is found in either regret insertion or regret-k insertion. Regret insertion is included in regret-k insertion while $k = 2$ because it only compares the regret value between the best and the second best solution. While $k > 2$, the regret-k insertion approach investigates the request value between the best and the $k-1$ best routes, it discovers the limitation of a insertion position of a request i earlier than regret insertion.

In this thesis, the lower bound of k is 3 and a upper bound of k is given by the number of the routes in the current solution. The maximum regret value is hence chosen to be inserted above all unassigned shipments.

New route insertion

New route insertion creates a new truck route that accommodates as many shipments as possible among the ones present in the unassigned shipment list.

4.5.3 Removal and Insertion Approaches Selection

There are different removal and insertion approaches used in the neighborhood search, therefore, two methods can be used in choosing the approach. The first method applies equal probability to select four removal and insertion approaches in every neighborhood search. The second method is a dynamic probability of the selection of the removal and the insertion approaches which is inspired by Ropke and Pisinger (2006). Instead of a equal random selection of the removal and the insertion approach pattern, the weight of different heuristics is based on the previous iteration result and would be adjusted dynamically.

Dynamic Probability to Choose the Removal and Insertion Approach

The basic idea of the dynamic probability is to monitor the performance of the removal and the insertion approach. The measurement of the weight is calculated by different result performances in the meta-heuristic method, which means a score is given to the corresponding result. To avoid a bias selection from happening, a number of segments is defined to reset the score. The probability of different neighborhood searches is determined by the performance of the previous segment, and the score would be updated at the beginning of every segment.

Three different scores are given while different performances are achieved. The first criteria of the score σ_1 is defined when the selected removal and insertion approaches are able to find a new overall solution through the neighborhood search. This is the highest score among three different criteria because it brings the result ahead to obtain a near optimal result. The second criteria of the score σ_2 is when a local improvement is achieved via the selected removal and insertion approaches, meaning a new solution is better than the current solution but not better than the best solution. As it is interesting to diverse the search, an acceptance of the worse solution is taken into account as well. The third criteria of the score σ_3 represents an encouragement of a new solution development, which is the least score among three criteria. The weight of heuristic performance is calculated at the end of the segment. The mathematical formula of the dynamic probability shows as follows:

$$W_{i,j+1} = W_{i,j}(1-r) + r\left(\frac{\pi_{ij}}{\sum_i \pi_{ij}}\right) \tag{4.9}$$

In this formula, $W_{i,j+1}$ represents the weight of heuristic i in the segment $j+1$, π_{ij} is the total score that heuristic i obtains in the last segment j and r represents a dummy variable. If $r = 0$, meaning heuristic i applies the weight that calculates from the last segment j and $r = 1$ means heuristic i begins the new segment and applies the new weight that obtains from the last segment j.

Apply the dynamic probability in the randomly choose would select the better performed heuristic and obtain the solution faster. However, this could not tell whether the selected insertion heuristic or the removal heuristic is more success in the model.

4.6 A Local Improvement: Time Slack Strategy

A time slack strategy is used to adjust while dock capacity constraint is violated. To eliminate its infeasibility, the time slack strategy is employed to reschedule truck arrival time. The basic concept of the time slack strategy is to guarantee every shipment can satisfy its time window constraint by shifting truck arrival time at the pickup or delivery nodes.

The usual approach adjusts the arrival time of a truck by shifting the last arrival truck later if arrival trucks are more than available docks. Table 4.1 and Figure 4.3 present an example that only two docks are available in the warehouse but three trucks arrive and generate dock capacity violation.

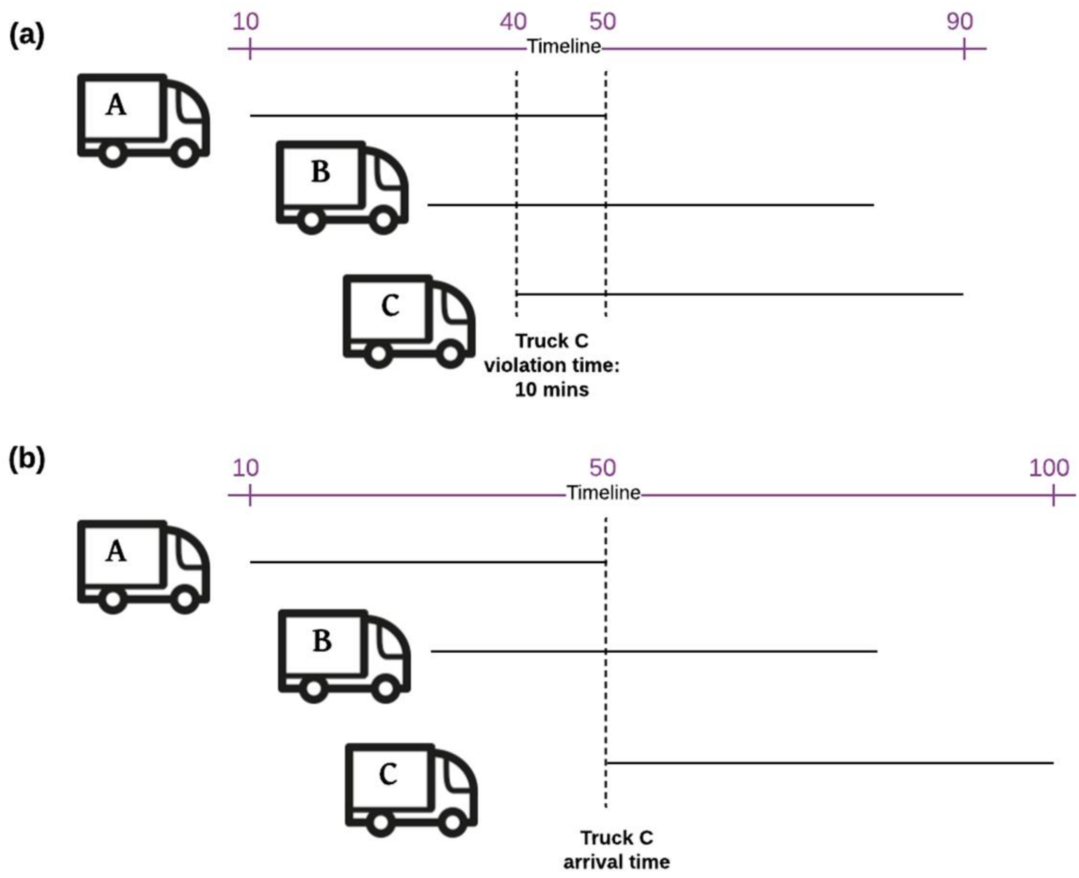

Figure 4.3: Resolve dock capacity violation by shifting the last arrival truck arrival time (a) the initial situation when dock capacity violation occurs(b) shift the last truck arrival time to satisfy dock capacity constraint

Table 4.1: Example of shifting the last arrival truck while dock capacity constraint is violated

Truck	A	B	C
Arrival time	10	30	40
Departure time	50	80	90

The arrival and departure time of each truck at a ground handler's side are known in Table 4.1. Fig. 4.3 shows while truck A and B are still processing their task, truck C arrives and violates the dock capacity constraint for 10 minutes. Therefore, truck C is shifted to arrive 10 minutes later. This movement could satisfy the dock capacity constraint of the current solution, no more arrival trucks are than available docks, but does not guarantee the satisfaction of time window constraints.

The time slack strategy is inspired by Li et al. (2016a) research which applies a different idea to shift the arrival time of the truck with the largest time slack. This approach consists the information of arrival time A_i^k, beginning of the service time B_i^k, departure time D_i^k, time window $[e_i, l_i]$ and time slack $\{l_i - B_i^k\}$ of every shipment.The following Table 4.2 and Fig. 4.4 illustrate an example while dock capacity constraint is violated for a period of time. Two docks are available but three trucks are visiting at a freight forwarder's warehouse at different time schedule.

Table 4.2 provides the detailed information about arrival time, departure time and time window of each truck to pickup their corresponding shipments. It shows while both truck A and truck B are still loading the shipments but truck C arrives at the warehouse; therefore, truck C has to wait another 10 minutes for its turn. Due to the shipment pickup time window constraint, a violation would occur if truck C is shifted to 50 minute to pickup the shipment. Understand that truck B has a longer time slack to pickup the shipment. It is more reasonable to shift the arrival time of truck B to 50 minute hence all trucks can pickup their shipments and satisfy the time window constraints.

While truck B is shifted to arrive later at the warehouse, the following schedule of every truck may be affected. Therefore, both time window and dock capacity constraints have to be examined again. However, this strategy may worsen the schedule of all scheduled trucks. A stopping criteria is set to avoid a worse solution from generating.

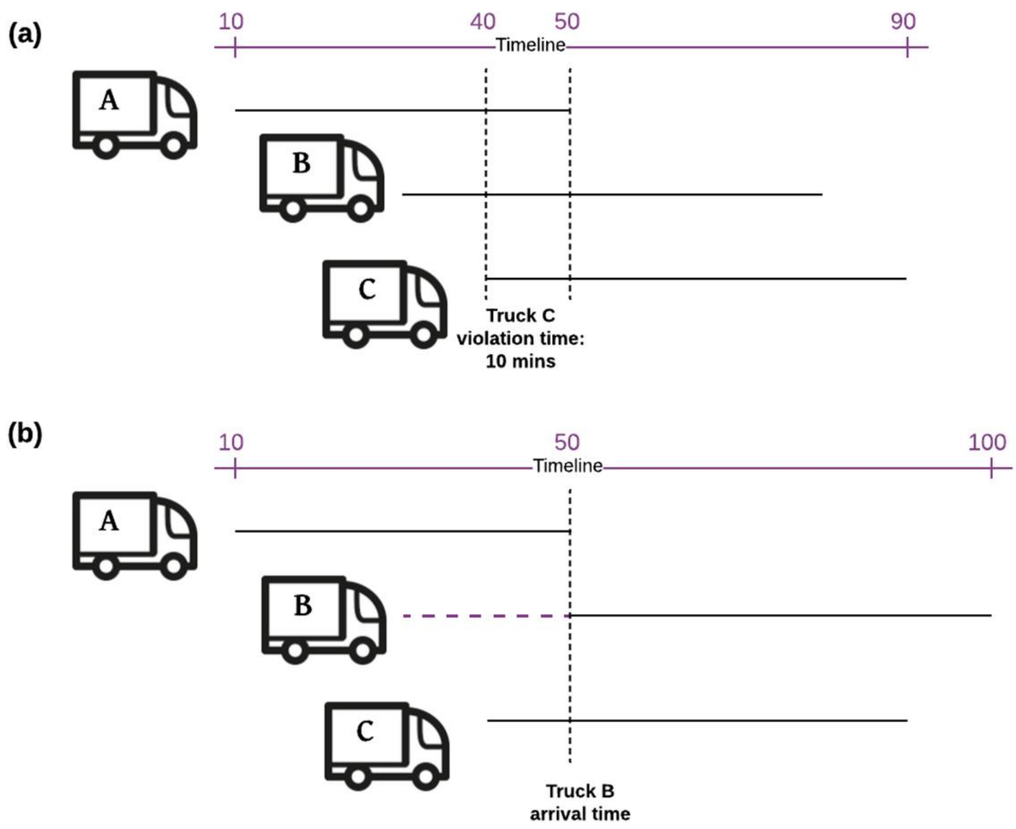

Figure 4.4: Resolve the dock capacity violation by applying the time slack strategy (a) Dock capacity violation occurs at the warehouse (b) After the time slack strategy is applied

Table 4.2: Example of time slack strategy while two docks are available in a warehouse

Truck	**A**	**B**	**C**
Arrival time	10	30	40
Departure time	50	80	90
Time window	[5, 20]	[30, 60]	[20, 45]
Time slack	10	30	5

4.7 Departure Time Adjustment

In general, all trucks are initially assigned to depart the depot as soon as possible. Cordeau and Laporte (2003) showed an approach to delay truck departure time to decrease the total travel time while satisfying time windows and ride time constraints. Given that $B_i^k = max\{A_i^k, e_i\}$ is the earliest possible service time for node i, how much the service at each node can be delayed (i.e., the slack time of node i) depends on the time characteristics of nodes downstream.

The forward time slack F_i^k quantifies the additional time that truck k can delay to start servicing request i before violating a time window or ride time constraint. The formula to determine such value is shown in Eq. 4.10, where q is the index of the last node in the current route, i.e., the destination depot. The formula contains the minimum of the total waiting time until shipment i, plus the difference between latest arrival time and the beginning of service time of shipment i (Cordeau and Laporte, 2003).

$$F_i^k = \min_{i \leq j \leq q} \{ \sum_{i < p \leq j} W_p + (l_i - B_j^k) \} \tag{4.10}$$

Devising a strategy that is based on F_i^k improves the quality of the solution by reducing total transport costs. Because the local improvement might worsen a feasible solution by introducing a time window violation or a new dock capacity violation downstream, the procedure to resolve dock capacity violations is stopped if the potential resolution of the first dock capacity violation would introduce an additional infeasibility. Note that, for every dock capacity violation that was resolved, the delayed truck would incur in an additional delay $Delay_i^k = B_i^k - (A_i^k + W_i^k)$. A last step is carried out to verify if the associated truck can have its departure pushed back by the same quantity, in order to avoid the dock capacity violation without increasing its travel time. The whole procedure is shown as follows:

1. Check if $Delay_i^k > 0$
2. Determine the position of $Delay_i^k$ in the visiting warehouse
3. Set $D_0 = e_0$
4. Compute A_i^k, W_i^k, B_i^k and D_i^k for every shipment i in truck k
5. Compute F_0^k for truck k
6. Determine the sequence of the visiting warehouse
7. Set $D_0^k = \min\{F_0^k, Delay_i^k, Delay_{i-1}^k\}$
8. Update A_i^k, W_i^k, B_i^k and D_i^k for every shipment i in truck k
9. Compute the feasibility of the solution after the change

The additional delay $Delay_i^k$ is checked in step (1) and then the sequence of the shipment i at the visiting warehouse is check in step (2). Note that, since we are delaying the docking operations of a truck, the shipment where the delay is actually applied is the first one in the intended ground handler.

Steps (3) to (5) determine the time stamps and the minimum forward time slack of the route until shipment i. Step (6) checks the sequence of the visiting ground handlers to determine if a departure delay of the current truck might create new dock capacity violations upstream, which is clearly an undesired effect.

Step (7) computes the minimum delay departure time that can applied without increasing a time window violation or creating a new dock capacity violation upstream. Note that a new dock capacity violation can be generated, only if the current truck is delayed at a ground handler that is not the first one in the delivery sequence. If previous ground handlers are visited, the maximum departure push back time for the current track is the minimum among (i) the slack time of all nodes preceding shipment i, and (ii) the maximum delay the truck can incur in all previous ground handler before generating an unplanned dock capacity violation. As mentioned before, if the dock capacity that was resolved occurs in the first ground handler visited, only option (i) is considered.

5 | Computational Results

In this chapter, the applicability and effectiveness of the developed meta-heuristic algorithm in designing good-quality routing solutions is tested. Schiphol Airport is used to the analysis as a reference, and real locations of freight forwarders and ground handlers are applied when designing the computational experiments. One reason why Schiphol Airport was chosen, is that in the past years congestion problems on the ground handler side were often present.

Section 5.1 presents the layout of the landside area that is the basis for all computational experiments. Section 5.2 illustrates the computational results of the instances. Section 5.3 presents a discussion of the computational results.

5.1 Study Area: Schiphol Airport

The layout of Schiphol Airport and the locations of five freight forwarders and five ground handlers are presented in Fig. 5.1, while Fig. 5.2 shows the distances between these ten warehouses.

The locations of freight forwarders and ground handlers are based on the real company locations, the orange rectangles define the locations of ground handlers and the blue rectangles define the locations of freight forwarders. The location of the central depot is chosen at the red rectangle. It is sufficiently close to all warehouses and a building-free area. Note that how to assess what is the best location of the central depot, which would be an interesting research topic, but it is beyond the scope of this work. In addition, the central depot and the fleet of trucks are assumed to be owned by the airport. Therefore, it is assumed that the airport is responsible for planning the truck routes.

Distances between warehouses are not Euclidean distances, but were computed by using Google Maps. The problem under scrutiny is compact from a distance perspective, being the maximum distance between warehouses (between ground handler 1 and freight forwarder 4) less than 10 km. Figure 5.2 shows the distance matrix of the five ground handlers and the five freight forwarders.

As mentioned in the problem description, the number of docks does not represent the overall number of docks available at each ground handler. On the other hand, it represents the number of docks that each ground handler reserves for trucks serving the consortium of freight forwarders. The available dock capacities in this thesis are different in each ground handler. The number of the available docks is [1, 2, 1, 1, 2] from ground handler 1 to ground handler 5, respectively.

Figure 5.1: Location of the five freight forwarders, of the five ground handlers, and of the central depot for the case study (background figure from Google Maps).

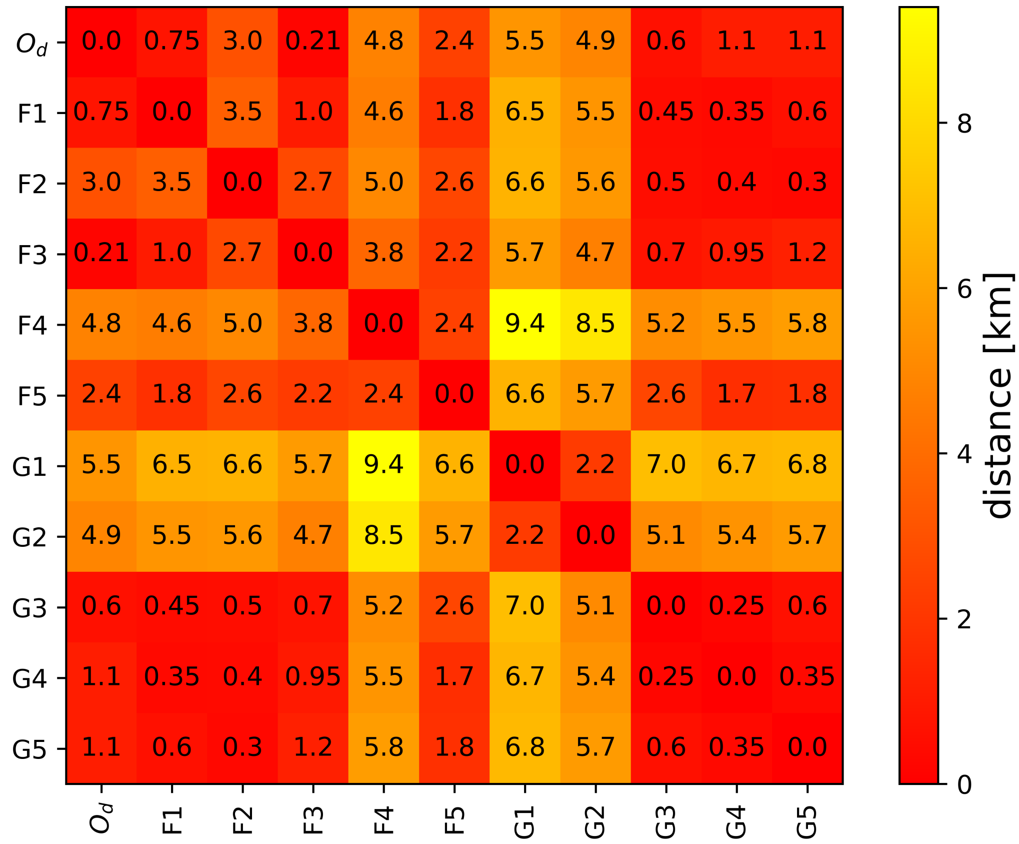

Figure 5.2: Distance matrix between freight forwarders, ground handlers, and the central depot.

5.2 Computational Results

In this section, computational results will be shown. First, a brief sensitivity analysis that addresses the weight of the infeasible contribution of the cost function is carried out. Then, a comparison between an equal probability and a dynamic probability in the selection process of the removal/insertion moves is shown. Finally, a comparison between the solution of the MILP model and the meta-heuristic method is provided for a set of 10 instances. Before showing the aforementioned analyses, an overview of the constants and parameters used in all computations is provided.

The distance and time related transport costs are based on Schonewille (2015), where the travel distance unit $Cd_{ij} = 0.66$ and the travel time unit $Ct_{ij} = 0.6$. Other parameters that are fixed in the model is: the initial temperature $T = 50$, the number of removal request $r = max\{5, \text{ number of unassigned shipments}\}$ and the cost that shipment i is not picked up by any truck $\alpha = 300$

5.2.1 Parameters Sensitivity

As discussed in Chap. 4, the infeasibility of a time-dependent constraint is allowed in our meta-heuristic method to explore and enrich the search space. In fact, exploring a neighborhood of such a solution might identify a feasible new best solution. While focusing on solutions characterized by a time-dependent infeasibility is allowed during the search, but the final output of the meta-heuristic method should be a feasible solution, if such a solution exists. As a consequence, for a specific instance it was assessed how the weight of the time-dependent infeasibility affected the final outcome.

We focused on a case there 50 shipments must be delivered from freight forwarders 1 through 5, to ground handlers 3 and 4. We tested 3 different weights for the time-dependent infeasibility, i.e., 1, 10, and 100. Results in terms of best solution cost, infeasibility of the best solution, and computational time are reported in Table 5.1.

Table 5.1: Result of different weights of time-dependent constraints parameters

parameters of time-dependent constraints	**total cost**	**travel time [mins]**	**infeasibility of time-dependent constraints [mins]**
1	814.6	1156.0	69.8
10	841.3	1314.2	0
100	841.1	1293.6	0

As highlighted in Table 5.1, when the weight is set equal to 1, the best solution is the lowest among the three, but it is not feasible. This is an undesired result we want to avoid. On the other hand, when the weight is set to 10 or 100, the best solution is the same, the time-dependent constraints are infeasible while explored during the search, but is not present in the best solution. While for this particular case, a weight of 10 was already sufficient to avoid infeasibility, we decided to use a weight equal to 100 in all computational analyses.

5.2.2 Comparison between Equal Probability and Dynamic Probability

Section 4.5.3 discussed two different methods to choose removal and insertion approaches in a neighborhood search: (1) select a removal approach and an insertion approach with an equal probability (2) select a removal approach and an insertion approach with a dynamic probability based on the previous performance. A comparison between the equal probability (non-adaptive strategy) and the dynamic probability (adaptive strategy) is reported in Fig. 5.3. The input data is as same as the previous case study, parameter sensitivity experiment. In addition, parameters of time-dependent violations are set equal to 100.

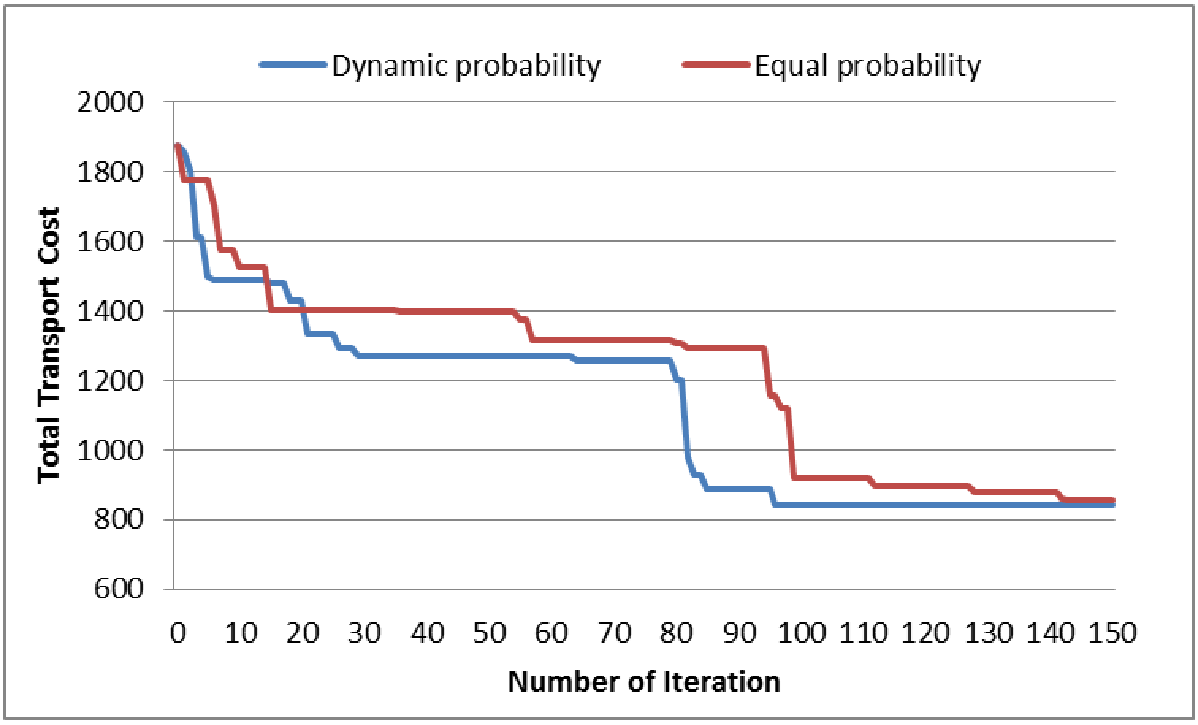

Figure 5.3: Comparison between equal probability and dynamic probability

The blue line indicates the result of applying the dynamic probability in the case study, and the red line indicates the result of the probability remaining unchanged. Although two approaches eventually converge, it is clearly shown how the dynamic probability outperforms the equal probability and reaches the local minimum faster within 150 iterations. Although we showed the comparison for a specific case and did not perform an exhaustive analysis, we decided to use the dynamic probability approach in all the computational results shown next.

5.2.3 Comparison between the Exact Model and the Meta-heuristic

The exact model is solved via a branch and bound (Barnhart et al., 1998). Initially, all integer constraints on variables are removed and a linear relaxation of the initial model is solved. This special situation is identified with the root node. Then, different branches are generated from the root node, where integrality of some of the variables is restored, and a different problem is solved. Each problem is associated with a node, whose solution can be of different kinds. As an example, a node can provide an infeasible solution, and thus there is no need to branch that node any further. A node can provide a solution that satisfying all integrality requirements, and thus does not need to be branched further as well. If that is the best integral solution, it is called the incumbent solution. On the other hand, a node can be associated with a feasible solution, but where not all integrality constraints are satisfied. In this case, branching is still required. The solver continues to iterate until there is convergence between an upper bound and a lower bound. In minimization problems, the upper bound is characterized by the incumbent solution, i.e., the best (lowest) integer solution until that point. The lower bound is the highest solution of a linear relaxation of the original problem, i.e., of one of the nodes of the tree structure that still need branching. The percentile difference between the current upper and lower bound is called gap, or gap optimality. An optimal solution is found when the gap is zero, i.e., the lower and upper bound to coincide. Depending on the characteristics of the MILP model, and on the size of the specific instance, reaching gap optimality of 0% might be computationally intractable, as the main scope of this thesis suggests. Figure 5.4 illustrates an example how the exact model to achieve the optimal solution.

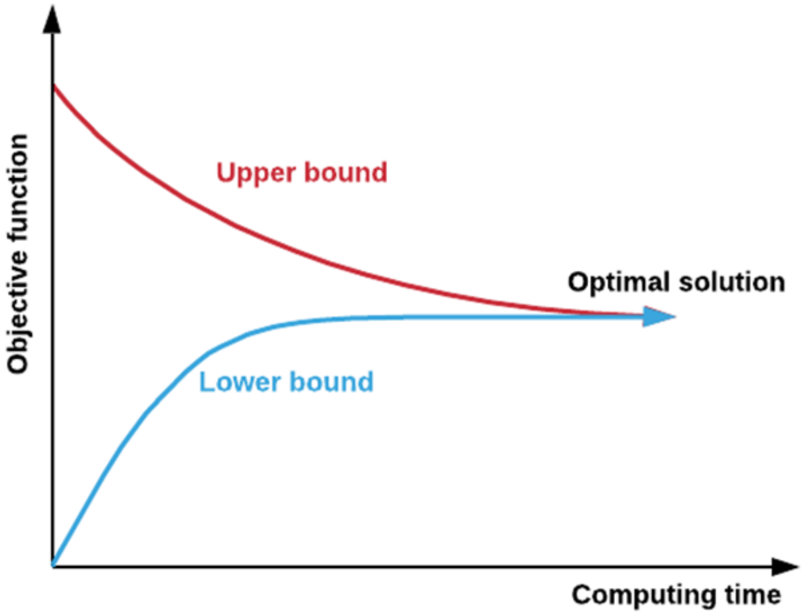

Figure 5.4: Branch and bound solution process for a MILP problem.

In order to provide a benchmark time in our comparison, a time limit of 120 minutes (2 hours) is set when computing the solution of the MILP model. Three outcomes are possible for each MILP instance. (1) no solution was found within the time-limit, (2) an incumbent solution was found, with a non-zero gap optimality, and (3) the optimal solution was found (i.e., gap optimality is 0%).

The parameters for the time-dependent constraints are set equal to 100, and the dynamic probability approach is applied in the neighborhood search. In the FF column we report the freight forwarders considered in each instance, while in the GH column we report the associated ground handlers. The locations of the warehouses can be retrieved in Fig. 5.1.

Table 5.2: Computational result of 10 instances in Schiphol Airport

Instances	**Total shipments**	**FF**	**GH**	**Total cost of heuristics method**	**Computing time [mins]**	**Total cost of exact model**	**Computing time [mins]**	**Gap [%]**
1	15	F1 & F2	G1 & G5	236.8	2.3	240.6	120	-1.58
2	15	F1 & F2	G3 to G5	203.5	1.5	203.9	120	-0.17
3	20	F3 & F4	G1 to G4	315.0	3.3	309.1	120	1.91
4	25	F4 & F5	G1 to G5	397.7	6.2	386.7	120	2.83
5	20	F3 to F5	G4 & G5	279.0	2.9	279.9	120	-0.35
6	30	F3 to F5	G3 to G5	428.4	10.6	432.3	120	-0.91
7	35	F1 to F3	G2 to G4	564.8	15.2	533.6	120	5.86
8	40	F1 to F3	G1 to G4	668.5	20.5	N/A	480	-
9	50	F1 to F5	G1 to G5	865.2	44.8	N/A	480	-
10	55	F1 to F5	G1 to G5	927.4	54.9	N/A	480	-

Results for the 10 instances are shown in Table 5.2. It is evident how the meta-heuristic method is effective in producing comparable, or even better solutions than the exact model in a shorter time. In particular, we believe that this solution is quite close to the best solution for all situations in which an existing solution is found. Unfortunately, the optimality gap is still very high because a very slow growth of the lower bound was experienced.

For 7 of the 10 instances presented, an incumbent solution was found, while for 3 instances no feasible integer solution was found. We highlighted the no feasible integer solutions within the specified time-limit by reporting N/A (not available) in the associated column. Instances 1, 2, 5, and 6 show a better solution in the heuristic model, while for instances 3, 4, and 7 the incumbent solution of the exact model is lower than the outcome of the meta-heuristic method. The maximum gap, intended here as percentile difference between the MILP and heuristic solutions, is 5.86% for instance 7, which still supports the good performance of the devised algorithm.

To provide a more visual comparison between the two solution, for instance 6 the truck routes are highlighted. In both solutions, 4 trucks are used overall. For this instance, forwarders 3, 4, and 5, and ground handlers 3, 4, and 5 are considered. Routes for the 2 approaches are shown, respectively, in Fig. 5.5 and 5.6. The red square is the location of the central depot O_d, the orange squares are the location of the ground handlers and the blue squares are the location of the freight forwarders.

The objective results of the exact model and the meta-heuristic method are similar, 433.3 and 428.4, respectively. An interesting finding while comparing the truck routes in the figures, truck 4 has a similar visiting route in both MILP and meta-heuristic results. From the figures we understand that MILP model searches the solution where trucks visit lesser warehouses than the meta-heuristic method. In other words, trucks are assigned to pickup more shipments in a freight forwarder's warehouse in MILP solution. However, the meta-heuristic method searches the solution where the total distance of four trucks is lesser.

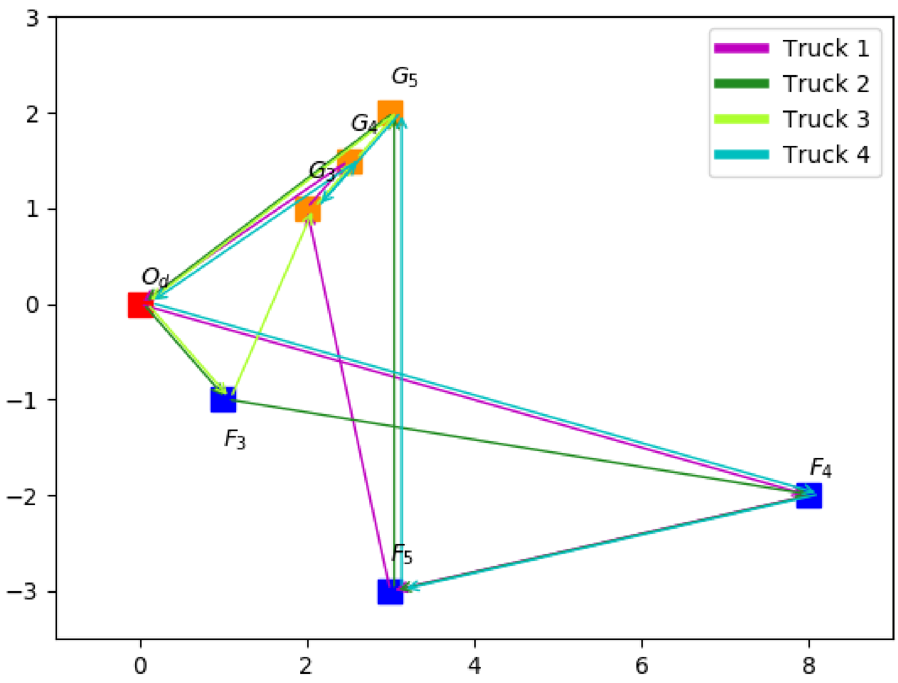

Figure 5.5: Truck route planning for instance 6 via MILP model

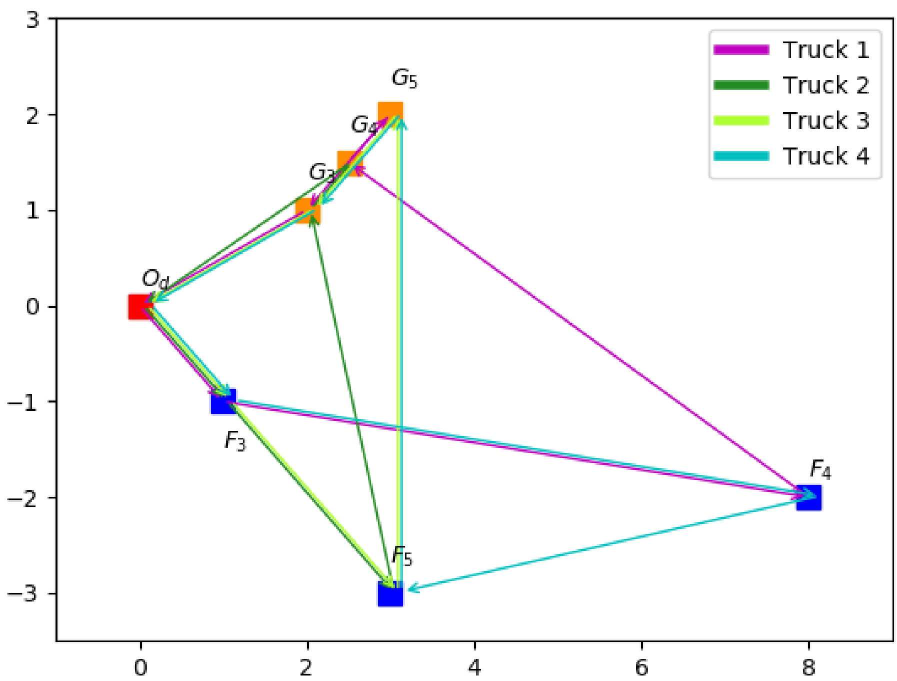

Figure 5.6: Truck route planning for instance 6 via meta-heuristic method

5.3 Discussion of Results

This section presents the discussion of the meta-heuristic results while applying it to the real operations.

Both the MILP model and the associated meta-heuristic are characterized by assumptions that might not entirely reflect practical operations. Some of the deriving limitations are discussed here. As an example, in both models, all shipments are equally considered if offloaded. This means that the extra cost associated with offloading shipment i or j is the same. In reality, each shipment is associated with an expected revenue, and this could be mapped into a shipment-specific penalty in case of offloading. With such an approach, the solution will steer towards routes that offload low-revenue shipments and prioritize high-revenue shipments, in case not all shipments can be delivered.

Besides, trucks are allowed to wait outside ground handlers for a dock to be available in both methods. In fact, in our solution, we do not necessarily make a waiting time to disappear for trucks, even if this is a quantity that the model tries to minimize the overall travel time by leaving the depot at different time. We did not pose any restriction on the number of trucks that can wait outside each ground handler. Due to the limit space in the airport apron, trucks have only one land to go to each warehouse. While this should not be a problem if the considered number of trucks and docks is small, things might be different with larger instances. An extension of the model would be to explicitly include several waiting spots per ground handler, such that the maximum number of trucks that can wait to be served at the ground handler is equivalent to the number of waiting spots.

6 | Conclusions and Recommendations

In this chapter, this research is concluded by answering the research questions in Sec. 6.1. The recommendations for future study and practical applications are discussed in Sec. 6.2.

6.1 Conclusions

A mathematical model can achieve the optimal solution but it is not appropriate for real-time and quasi-real-time applications. The goal of this research is to propose a meta-heuristic method to obtain a sound solution in a reasonable time. To achieve the goal, three sub research questions were formulated. After resolving the sub questions, the main research question is answered.

1. What causes the truck congestion problem between air- and land side handlers and who are the related actors?

The landside air cargo supply chain is a complex system, with multiple stakeholders who have conflict of interests. As such, narrowing down the causes of congestion for export operations can be complicated. Nonetheless, some contributing factors are proven to be a crucial causes to the congestion. First, there is generally an imbalance between the number of ground handlers and freight forwarders around an airport, since the ground handlers need to be located on the airport premises. This means that, even for major airports, a handful of ground handlers must process shipments coming from all the forwarders operating within the airport. We translated this imbalance into a model requirement by explicitly considering dock capacity on the ground handler's side. The second reason lies in the lack of coordination between different freight forwarders when designing truck export tours. This might cause trucks coming from different freight forwarders to head simultaneously to the same ground handler, creating an imbalance between supply (docks available), and demand (trucks). The central planning approach with horizontal collaboration, which is the basis of this work, addresses this problem.

Besides the main stakeholders, freight forwarders and ground handlers that mentioned above, shipper and consignee and airline are also suffering with such truck congestion problem. Freight forwarders have to reschedule the delay freights and inform their customers, both shipper and consignee, which might hinder their reputation and cause their customers to choose different service providers for future services. In addition, freight forwarders are not willing to pay truck drivers to wait in a line (Barnett, 2019), which contributes to an unnecessary monetary losses of economic benefits. A full freight aircraft has a flexible departure schedule which allows it to wait for all shipments to load. However, the delays in uploading processes bring an extra waiting and monetary losses. Shipper suffer the damage of reputations and losing customer bases if the consignee does not receive goods within the intended time-frame. Because of the truck congestion leads to air freight delay, shippers and consignees are more willing to transport their goods by the nearby airport. The corresponding airlines and freight forwarders will move their assets to deliver services to them. As a result ground handlers will incur losses when this series of actions is completed.

2. What are the feasible and efficient methods to address our problem, and which is suitable to this thesis?

Common approaches to solve NP-hard problems are heuristic method, such as SA, ALNS and GA. In order to construct an efficient model, there are constraints to the problem that need to be considered. Due to the structure of the airport operations, the export goods are transported from the freight forwarder to the ground handler, while the imported goods are transported the opposite way. Trucks are assigned by a central depot due to horizontal collaboration between respective freight forwarders.

Tabu search applies local search as its neighborhood search but forbids the non-improving moves to the next iteration and a population of data is required in GA to evaluate. SA accepts infeasible solutions while visit the neighborhood solutions. It is hence chosen to be the basic outline of the meta-heuristic method. To generate a better solution in a shorter time, ALNS is applied to search for neighborhood solutions. ALNS utilizes the ruin and recreate principle for the removal and insertion approaches in order to search the near-optimal solutions in a reasonable time.

3. How to resolve dock capacity violations when designing truck routes?

When designing truck routes, some routes are infeasible in a strict sense. As an example, each truck has a maximum weight capacity, and the overall loaded shipments must not out weight this limit. On the other hand, given a set of truck routes that do not satisfy dock capacity constraints when considered as a whole. The solution might still be feasible with adjustments to some of the routes. These constraints are called time-dependent constraints. To reduce the time-dependent violations, a time slack strategy is proposed.

The time slack strategy is applied to reschedule truck's arrival time when there is dock capacity violation. The main concept of the time slack strategy is to decrease the truck waiting time while satisfying the time window constraint for every shipment. The time slack means the maximum time that the service time at a node can be delayed before missing a time window of one of the nodes. In other words, the time slack of node i is the maximum delay a truck can incur, given the schedule, to start servicing node i before introducing infeasibility in the solution. The solution s' where trucks cause a dock capacity violation, we decide to use the time slack strategy to select which truck to delay. More specifically, we postpone the trucks whose residual time slack (i.e., the time slack after the truck has been delayed to resolve the violation), have reached the maximum threshold. This approach is more efficient than a First-In-First-Served (FIFS), instance, to avoid schedule modifications, we can easily introduce a time window violation while trying to resolve a dock capacity violation.

Based on the provided answers to the sub research questions, the main research question can be answered. The main research question was defined as follows:

How can truck routes be optimized efficiently without significantly compromising the solution accuracy in the context of the air cargo supply chain?

The proposed algorithm embeds the ALNS into the SA algorithm to achieve a near-optimal solution for the truck route design. The initial solution is found by the *basic greedy insertion*. Further, this paper applies ALNS to search for neighborhood solutions. Paired requests are highlighted and purged into a removal list. After which, they are inserted in better positions of the routes. A dynamic probability selection process is engineered to choose the removal and insertion approaches in the neighborhood search. This technique is yield to provide better results than a non-adaptive approach. Given a set of test instances, solutions generated by the meta-heuristic method is equal or superior than the ones produced by the exact model. Out of ten trails, the meta-heuristic method outperforms the exact model in the terms of time usage. In addition, the meta-heuristic method was able to find a feasible solution for those cases where the exact model fails to identify a feasible solution.

6.2 Recommendations and Future Research

In the final section, recommendations for practical applications and future research are discussed.

In this work, only export operations are considered. A logical addition to this research would be to include import operations as well, giving trucks leaving the depot with the freedom to perform either export (freight forwarder side and then ground handler side) or import (vice versa) tours. For example, once an export truck completes its unloading procedures on the ground handler side, it could load some import shipments and deliver them to the intended freight forwarders.

The meta-heuristic method currently has not restricted the number of trucks that can wait outside each ground handler. Land usage on the airport apron grants precedence to aircraft over vehicles. Therefore, limitation to space that trucks are able to occupy at the ground handler's side is in presence. A rational extension to this method is to restrict the number of waiting trucks at the ground handler's side.

In terms of practical implementation of the model, two studies are recommended to make the meta-heuristic method more effective in real business cases.

First, this thesis assumes the central depot is operated by the airport, where the location is a building-free area. However, airport operators may have different opinions when placing the central depot. To determine the optimal location for the central depot, a study regarding the venue is needed.

Second, a study about the monetary benefit to the stakeholders is necessary. Airport operators are responsible for coordinating transport services between freight forwarders and ground handlers. However, it is essential to know which level an airport would be benefit when the mentioned method is applied. For example, Schiphol Airport might benefit from the method due to its high-value air freights. Nontheless, it does not ensure another airport, such as Cologne Bonn Airport, would benefit from this method. Therefore, a cost-benefit analysis should be undertaken to determine the suitable airport to apply.

Bibliography

ACN (2019). ACN website. https://acn.nl/milkrun/milkrun-concept/. Accessed: April 2019.

Afifi, S., Dang, D.-C., and Moukrim, A. (2013). A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints. volume 7997, pages 259–265. Springer Berlin Heidelberg.

Agarwal, R., Ergun, Ö., Houghtalen, L., and Ozener, O. O. (2009). Collaboration in cargo transportation. In *Optimization and Logistics Challenges in the Enterprise*, pages 373–409. Springer.

Aircargo news (2018). Regional airports look to capitalise on congestion at major hubs. https://www.aircargonews.net/cargo-airport/regional-airports-look-to-capitalise-on-congestion-at-major-hubs/. Accessed: July 2019.

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.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167 – 173.

Baños, R., Ortega, J., Gil, C., Fernández, A., and de Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, 40(5):1696 – 1707.

Barnett, C. (2019). Forwarders rally response to congested us air hubs. https://www.joc.com/air-cargo/international-air-freight/forwarders-skeptical-congestion-mitigation-plans-jfk-logan_20190301.html. Accessed: July 2019.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Bombelli, A. and Tavasszy, L. (2018). A pickup and delivery formulation for landside air cargo supply chain optimization. (unpublished).

Buso, F. (2017). Analyzing the "milkrun" service process for inbound freight at schiphol airport with queuing theory and simulation. Master's thesis, University of Mannheim.

Cargo, S. (2019). Shaping the smartest cargo hub. `https://www.schiphol.nl/en/cargo/`. Accessed: March 2019.

Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579 – 594.

Cruijssen, F., Cools, M., and Dullaert, W. (2007). Horizontal cooperation in logistics: Opportunities and impediments. *Transportation Research Part E: Logistics and Transportation Review*, 43(2):129 – 142.

Gomez, F. and Scholz, D. (2009). Improvements to ground handling operations and their benefits to direct operating cots. Master's thesis, Hamburg University of Applied Sciences.

Group, S. (2018). Facts and figures 2017. Accessed: March 2019.

Hosny, M. (2011). Comparing genetic algorithms and simulated annealing for solving the pickup and delivery problem with time windows.

ICAO (2016). Moving air cargo globally. `https://www.icao.int/Security/aircargo/Moving\%20Air\%20Cargo\%20Globally/ICAO_WCO_Moving_Air_Cargo_en.pdf`. Accessed: April 2019.

Li, B., Krushinsky, D., Woensel, T. V., and Reijers, H. A. (2016a). An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research*, 66:170 – 180.

Li, H. and Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186.

Li, Y., Chen, H., and Prins, C. (2016b). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1):27 – 38.

Max, v. I. (2018). Amsterdam airport schiphol doing a case study about creating key performance indicators for measuring the processes of the ground handlers. Master's thesis, Amsterdam University of Applied Science. `https://acn.nl/wp-content/uploads/2019/01/Max_van_Iersel_KPIs_Ground_Handlers_Schiphol.pdf`.

Perez Bernal, M., Val, S., Larrodé, E., and Sainz, R. (2012). Optimization of the air cargo supply chain. *Journal of Airline and Airport Management*, 2.

Pieters, S. (2014). The air cargo supply chain at schiphol, process improvement and collaboration. Master's thesis, Delft University of Technology, Master thesis.

Portal, T. S. (2018). Volume of airfreight processed by europe's leading airports in 2017 , by airport (in 1,000 tonnes). https://www.statista.com/statistics/434381/airfreight-volumes-in-europe-by-airport/.

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. cited By 555.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, 4(1):285–305.

Schonewille, G. (2015). Calculation of transport cost for freight carriers on the last mile. Master's thesis, Delft University of Technology, Master thesis.

Wang, C., Mu, D., Zhao, F., and Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Computers & Industrial Engineering*, 83:111 – 122.

Weisbrod, G. and Fitzroy, S. (2011). *Traffic Congestion Effects on Supply Chains: Accounting for Behavioral Elements in Planning and Economic Impact Models*. http://cdn.intechweb.org/pdfs/18515.pdf.

A | Appendix

Research Paper

An Improved Optimization Method to Increase the Truck Planning Efficiency of Landside Air Cargo Operations

Ting Wei Wu,* Alessandro Bombelli,† Lóránt Tavasszy,‡

Delft University of Technology, Delft, The Netherlands

Abstract

Congestion in the landside air cargo supply chain occurs for different concurring reasons. Lack of coordination between freight forwarders, as example, might create truck congestion on the ground handler side. Horizontal collaboration between forwarders can be introduced and modeled via mathematical programming to mitigate congestion. Resulting models, which are generally variations of Pickup and Delivery Problem with Time Windows (PDPTW), can be solved to optimality only for small-size instances, and the computation is generally time consuming. We therefore propose a simulated annealing (SA)-embedded adaptive large neighborhood search (ALNS) heuristic to address truck route planning in the landside air cargo supply chain. In this work, we allow the search to visit infeasible time-dependent solutions. Accordingly, the objective function minimizes the feasible solution, where total travel distance cost, total travel time cost and unassigned shipments cost, and the time-dependent violation costs. Computational results are reported for 10 instances that were also solved with a mathematical programming approach. Results shows that the meta-heuristic method performs equally or better than the mathematical model given a computational limit for the latter of 2 hours. In addition, the meta-heuristic method was able to find a feasible solution for those cases where the exact model failed to identify a feasible solution.

Keywords: Landside airport operations, Vehicle routing, Simulated annealing, Time slack

1 Introduction

The air cargo supply chain is a complex process that involves several stakeholders cooperating together to carry out door-to-door transportation of a shipment from a shipper to a consignee (ICAO, 2016). Different bottlenecks and inefficiencies might harm the soundness of the overall transportation process causing congestion issues. The expected growth of the air freight raises challenges to the air cargo chain to different levels. Some factors have been identified to play a crucial role in contributing to congestion. First, there is generally an imbalance in numbers between ground handlers and freight forwarders around an airport. Only ground handlers are located on the premises of an airport, which characterized them as a bridge between freight forwarders and airlines (Max, 2018). Therefore, a handful of ground handlers must process shipments coming from all the forwarders operating with the airport. The second reason lies in the lack of coordination between freight forwarders when designing their truck tours. Trucks coming from different freight forwarders might be headed simultaneously to the same ground handler, creating an imbalance between supply (docks available) and demand (trucks), and creating delays in the delivery process.

Previous research has shown that horizontal collaboration, i.e., the collaboration between stakeholders in the same level of the supply chain, is beneficial in reducing the transportation cost (Ankersmit et al., 2014; Buso, 2017; Bombelli and Tavasszy, 2018). To solve the truck congestion problem in an airport, Bombelli and Tavasszy (2018) designed a Landside Air Cargo Supply Chain Pickup and Delivery Problem with Time Window (LACSC-PDPTW) model to plan optimal truck routes where horizontal collaboration between forwarders was allowed. The model is based on a Mixed Integer Linear Programming (MILP) problem and solved exactly for small instances. In many cases, the process to obtain the optimal solution is computationally intractable, which poses a challenge for real-time or quasi-real-time applications. The goal of this research is to develop a meta-heuristic method to reach good-quality solutions for LACSC-PDPTW instances within a limited computational time.

The rest of this paper is structured as follows. The relevant academic literature is reviewed in Sec. 2. The mathematical formulation of the LACSC-PDPTW is presented in Sec. 3, while the proposed methodology is described in Sec. 4. Section 5 presents the computational results. Finally, conclusions are drawn in Sec. 6.

*MSc student, Transport Intrastructure & Logistics, Faculty of Civil Engineering and Geosciences, Delft University of Technology

†Postdoctoral researcher, Transport & Planning, Faculty of Civil Engineering and Geosciences, and Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

‡Professor, Engineering Systems and Services, Faculty of Technology, Policy and Management, and Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology

2 Literature Review

As stated before, the main goal of the research is to develop a meta-heuristic method to plan truck routing in airport aprons. Every meta-heuristic method must satisfy the same assumptions and requirements as the associated mathematical method. The first part literature review is about the mathematical models addressing vehicle routing in logistics. Then, an overview of meta-heuristic methods is provided.

The Vehicle Routing Problem (VRP) is a seminal model that solves the routing problem for a fleet of vehicles to collect pickup or delivery goods at their corresponding request nodes (Savelsbergh, 1985). In general, vehicles leave the depot and deliver shipments to their demand nodes or pickup shipments from their supply nodes and return to the depot. In a similar fashion, in the landside air cargo supply chain, trucks perform a pickup tour among freight forwarders, and then a delivery tour among ground handlers. A major difference is that, due to the horizontal collaboration, both tours are built progressively and not carried out in a single distribution center. Some general requirements of the VRP, and of every model inspired by the VRP, are: (1) each vehicle starts and ends at the same depot, (2) each customer is serviced exactly once by a vehicle, (3) the total capacity of a vehicle should not exceed its capacity all the time.

The Vehicle Routing Problem with Time Windows (VRPTW) was introduced as a variant of the VRP when bounds on pickup and delivery time are present. Vehicles service each customer within an associated time window and the vehicle must remain at the customer location during the service.

The Pickup and Delivery Problem with Time Windows (PDPTW) is another important extension of the VRP, which is frequently used in logistics and the transportation industry (Li and Lim, 2003). The PDPTW is similar to the VRPTW because of capacity limitations on vehicles and on the fact that each truck should leave and go back to the same depot. However, the biggest difference is that requests are paired in the PDPTW, but not in the VRPTW and the VRP (Hosny, 2011). In other words, the VRPTW is used when all request nodes are formed by the same characteristic requests, which is either delivering shipments from the deport or picking up shipments to be returned to the depot. The PDPTW is used to assign an empty truck from the depot to pickup a shipment at the pickup node and then deliver to its corresponding delivery node then return an empty truck to the depot. Due to the paired request characteristics of the PDPTW, there is a precedence constraint. Pickup nodes must be visited before the corresponding delivery nodes (Li and Lim, 2003).

Typically, there are two approaches to solve such mathematical models. One is an exact method that obtains an optimal solutions, and another is a meta-heuristic method that searches a near-optimal solution in a (generally) shorter time. When the size of the problem grows, every exact method poses a computational challenge. Meta-heuristic methods are a viable solution to obtain sub-optimal, or even optimal solutions for models and instances where computing the exact solution of the MILP problem is computationally intractable.

The most common meta-heuristic methods are Tabu Search (TS), Simulated Annealing (SA), Adaptive Large Neighborhood Search (ALNS) and Genetic Algorithm (GA). Li and Lim (2003) apply tabu-embedded SA algorithm, which restarts the current best solution when the iterations do not improve within the SA structure, to solve the PDPTW problem. Ropke and Pisinger (2006) introduce ALNS, which is an extension from the Large Neighborhood Search (LNS), to effectively solve the PDPTW problem. ALNS apply different removal and insertion approaches in each iteration, and an adaptive weight is applied to select one of the approaches. In addition, SA is used to be a stopping criteria in the ALNS. Baños et al. (2013) use SA to solve the multi-objective VRPTW problem. Li et al. (2016a) apply ALNS to solve transport routing planning when people and parcels are sharing vehicles. Two solution evaluation approaches are used in the objective function in Li et al. (2016a) research, one only allows visiting feasible solutions and other considers infeasible solutions by adding a penalty to the constraint violation. In addition, the penalty weights for infeasible solutions are modified according to the feasibility of the solution. Li et al. (2016b) use ALNS to address collaborative logistics routes, which is so called Pickup and Delivery Problems with Time Windows, Profit and Reservation Requests (PDPTWPR).

A general description of the different heuristic methods is provided. TS applies local search as its neighborhood search outline but forbids the non-improving moves in the next iterations. The forbidden movement is stored in a list and used to avoid cycling the visiting solution from continuously moving back and forth (Cordeau and Laporte, 2003). GA requires a population of candidate solutions to be evaluated; moreover, it is often used to cluster the requests but not is not aware of how they are routed (Hosny, 2011). ALNS applies the ruin and recreate principle but uses different neighborhood searches in different iterations to find a better solution (Ropke and Pisinger, 2006). SA allows visiting infeasible solutions to find a global best solution, it accepts or rejects the solution according to its relative cost (Ropke and Pisinger, 2006). A comparison for solving PDPTW between GA and SA methods is carried out in Hosny (2011), where it is shown that the SA method yields a better average result than GA method. In addition, SA is proven to be faster.

3 Problem Definition

This section provides a formal description of the LACSC-PDPTW model and the basic constraints that limit the solution space in the meta-heuristic method. In the LACSC-PDPTW, a central planner with information regarding weight, destination, time window of shipments, computes the optimal routing strategy that also accounts for dock capacity on the ground handler's side. Figure 1 displays an example of optimal route for trucks. The gray diamonds O_d and D_d represent the origin and destination depots respectively, which may be the same depot, but are different from a modeling perspective. The solid line shows pickup and delivery orders of shipments and the dashed line presents the truck route to the corresponding docks while picking up/delivering shipments. In addition, the blue line represents the truck 1 route, and the yellow line represents the truck 2 route.

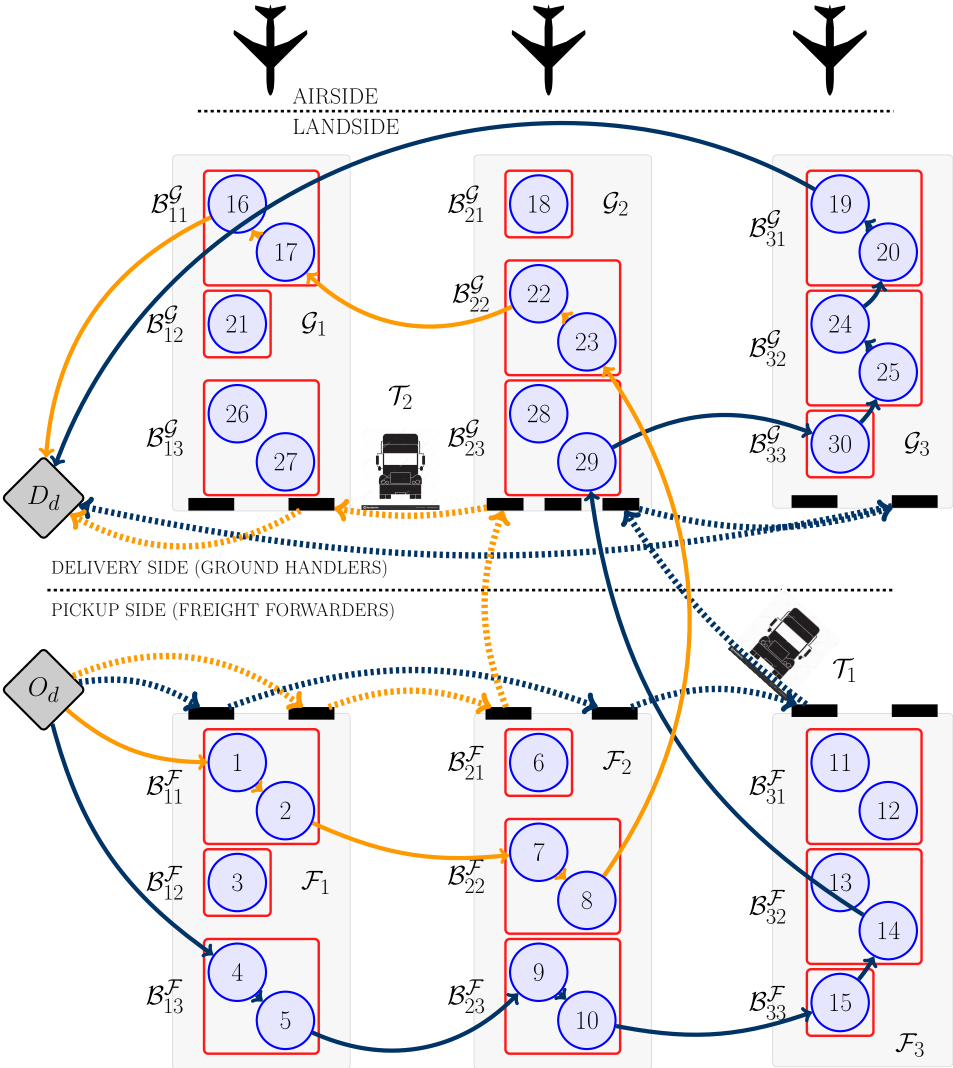

Figure 1: LACSC-PDPTW framework (reference: Bombelli and Tavasszy (2018))

The problem contains σ requests and k vehicles, and is based on a directed graph $G = (N, E)$ where $N = O_d \cup D_d \cup N_P \cup N_D$ is the set of nodes, and E is the set of edges. Subsets O_d and D_d represent the origin depot and the destination depot. N_P is the set of pickup nodes and N_D is the set of delivery nodes. As σ is a number of shipments to be delivered, the total number of nodes is $2\sigma + 2$. O_d is node 0, pickup nodes $N_P = \{1, ..., \sigma\}$, delivery nodes $N_D = \{\sigma + 1, ..., 2\sigma\}$ and the destination node D_d is node $2\sigma + 1$. K is a set of vehicles, which $k \in K$.

A set of freight forwarders N_F and ground handlers N_G are considered in the model, which are grouped into blocks (Figure 1). The notation of freight forwarders is $F = F_1, F_2, ..., F_{N_F}$ while for ground handlers is $G = G_1, G_2, ..., G_{N_G}$ in the model. Each shipment in a freight forwarder is grouped in a block B^F_{ij}, which contains the information about the shipment that has to be picked up from the freight forwarder i and then delivered to the ground handler j. Similarly, each shipment at ground handler N_G is grouped in block a B^G_{ij}, meaning that the shipment is delivered from the freight forwarder j to the ground handler i.

For each $(i,j) \in E$, it is assigned a distance $d_{ij} \geq 0$ and a travel time $t_{ij} \geq 0$. In addition, the travel time between node i and j is defined as $t_{ij} = t'_{ij} + T_c$. t'_{ij} is the travel time between node i and j, while T_c is a fixed additional time if node i and j belong to different warehouses. This is an additional check-in time that truck drivers need to spend on document checking procedures if visiting different warehouses. Each shipment request i has its own service time s_i and a time window $[e_i, l_i]$ for both pickup and delivery procedure. s_i indicates the time for loading or unloading request i in the warehouse, and the time window represents the earliest and the latest time for request i to be picked up or delivered at the warehouse. For instance, the truck driver is allowed to arrive at the warehouse earlier than e_i but has to wait until e_i for picking up the shipment i, and he is not allowed to arrive at the warehouse later than l_i. The capacity of truck $k \in K$ is set by a maximum weight capacity Q_k.

The goal of the LACSC-PDPTW model is to maximize the number of shipments delivered while minimizing transportation costs at the horizontal collaboration landside shipping. The objective function and the set of variables and parameters, as given by Bombelli and Tavasszy (2018), are as follow:

- X_{ij}^k: binary variable is 1 if truck k goes from node i to node j, 0 otherwise
- Z_i: binary variable is 1 if shipment i is not picked up by any truck, 0 otherwise
- $\tau_{D_d}^k$: a continuous time variable when truck k leaves the origin depot
- $\tau_{O_d}^k$: a continuous time variable when truck k goes back to the destination depot
- d_{ij}: distance from node i to node j
- C_d: the transport cost per distance unit
- C_τ: transport cost per time unit
- α, β, γ: the parameters that control the relative importance of different terms

$$\textit{Min: } \alpha \sum_{i \in N} Z_i + \beta C_t \sum_{k \in K} \sum_{(i,j) \in E} d_{ij} X_{ij}^k + \gamma C_\tau \sum_{k \in K} (\tau_{D_d}^k - \tau_{O_d}^k) \tag{1}$$

Some operational constraints are added to limit the solution space the model can explore, and are consistent with logistics requirements. For instance, a Last-In-First-Out (LIFO) approach is used for the delivery tour to avoid the potential unnecessary unloading and reloading of shipments. This is particularly relevant if some shipments have already consolidated into the ULD at freight forwarder's side. LIFO approach is the only viable solution to avoid unnecessary operations if the lateral occupancy of shipments (e.g., ULDs) makes it difficult to reach shipments loaded further down inside the trailer. All five constraints are shown in the following:

1. **Time window constraint:** shipments are only allowed to be picked up or delivered within a certain period.
2. **Total weight constraint:** the total maximum weight of all shipments that truck can load in the assigned route.
3. **Sequence constraint 1:** trucks must follow the sequence of visiting pickup points and then the corresponding delivery points in the assigned route.
4. **Sequence constraint 2:** trucks are only allowed to visit the freight forwarder and the ground handler once in their routes. This sequence constraint decreases the unnecessary movement for truck driving back and forward to the same warehouse.
5. **Sequence constraint 3:** trucks apply LIFO approach to pickup and deliver shipments.

Apart from small instances, an optimal solution cannot be computed by the LACSC-PDPTW within a reasonable computational time. Ideally, the central planner would process each morning (or every few hours) shipment information and would output the optimal routing schedule. Therefore, a heuristic method is applied to search a near-optimal solution in a reasonable computational time. To compute the same case study in the meta-heuristic method, the same constraints of the LACSC-PDPTW must be satisfied. In our meta-heuristic, time window constraints and dock-capacity constraints are seen as the time-dependent (meaning that an eventual violation might be recovered with changes in the time schedule), while the total weight constraint and the sequence constraints 1 to 3 are seen as the time-independent constraints (meaning that an eventual violation cannot be recovered).

4 Methodology

The core of the chosen meta-heuristic method is SA and the neighborhood search is based on the ALNS method described by Ropke and Pisinger (2006). Figure 2 presents the solution approach for the meta-heuristic method. An initial temperature T and a maximum iteration L are set at the beginning, and then $s_{current}$ is produced by an initial solution.

The dock capacity constraint is defined as a violation when more trucks arrive at a ground handler than the available docks at the same time. To provide more flexibility of the solution, a penalty is given in the objective function when a time-dependent constraint is violated. The objective function accounts for a feasible component, and the infeasible component deriving from eventual time window or dock capacity violations. Hence, the objective function of the meta-heuristic method is defined as follows:

$$J^* = J + \gamma_1 \sum_{i \in N} TWV_i + \gamma_2 DCV \tag{2}$$

Term J represents the feasible component of the objective function (see equation 3), which is the same cost function as the MILP model. On the other hand, two additional terms represent time window and dock capacity violations, respectively. γ_1 is a violation cost per time unit if shipment i is not picked up or delivered before its latest arrival time. TWV_i is total time units that shipment i violates its time window interval. γ_2 is the violation cost per time unit while truck k waits for an available dock at the warehouse and DCV is the overall time units where more trucks than the available docks are assigned to each ground handler.

The feasible solution J includes three different costs: (1) travel time cost of each truck (2) travel distance cost of each truck (3) unloaded shipments cost.

$$J = \sum_{k \in K} \sum_{(i,j) \in E} (Cd_{ij} D^k_{ij}) + \sum_{k \in K} \sum_{(i,j) \in E} (Ct_{ij} T^k_{ij}) + \alpha \sum_{i \in N} Z_i \tag{3}$$

The set of parameters in the objective function is described below:

- Cd_{ij} : travel distance cost from node i to node j
- Ct_{ij} : travel time cost from node i to node j
- α : cost that shipment i is not picked up by any truck

The set of notation in the objective function is provided as:

- D^k_{ij} : the distance that truck k travels from node i to node j
- T^k_{ij} : the travel time that truck k spends from node i to node j
- Z_i : a binary variable is 1 means shipment i is not picked up by any truck, 0 otherwise

4.1 Initial Solution Computation

An initial solution is constructed with the *basic greedy insertion*, which is detailed described in Sec. 4.2. The minimum objective in an initial solution is J^*. All shipments are initially in an unassigned shipment list. The solution is built by iteratively inserting nodes from the unassigned shipment list into the route. When no more shipments can be added in the route, because one of the time-independent constraints would be violated, a new route is created.

4.2 Methodology of the ALNS

The route planning in this research is computed with paired pickup-delivery customers. The neighborhood search is based on the ALNS described by Ropke and Pisinger (2006) and Azi et al. (2014). Four different removal and insertion approaches are described in the follows. The number of removal requests is determined by the parameter $r \in \{0, ..., N\}$. The number of r decreases during the insertion process. When r is equal to zero, it means that there is no request waiting for the insertion and no search will be performed. Shaw removal, random removal, worst removal and shortest route removal are introduced in the removal approach. Basic greed insertion, regret insertion, regret-k insertion and new route insertion are discussed in the insertion approach.

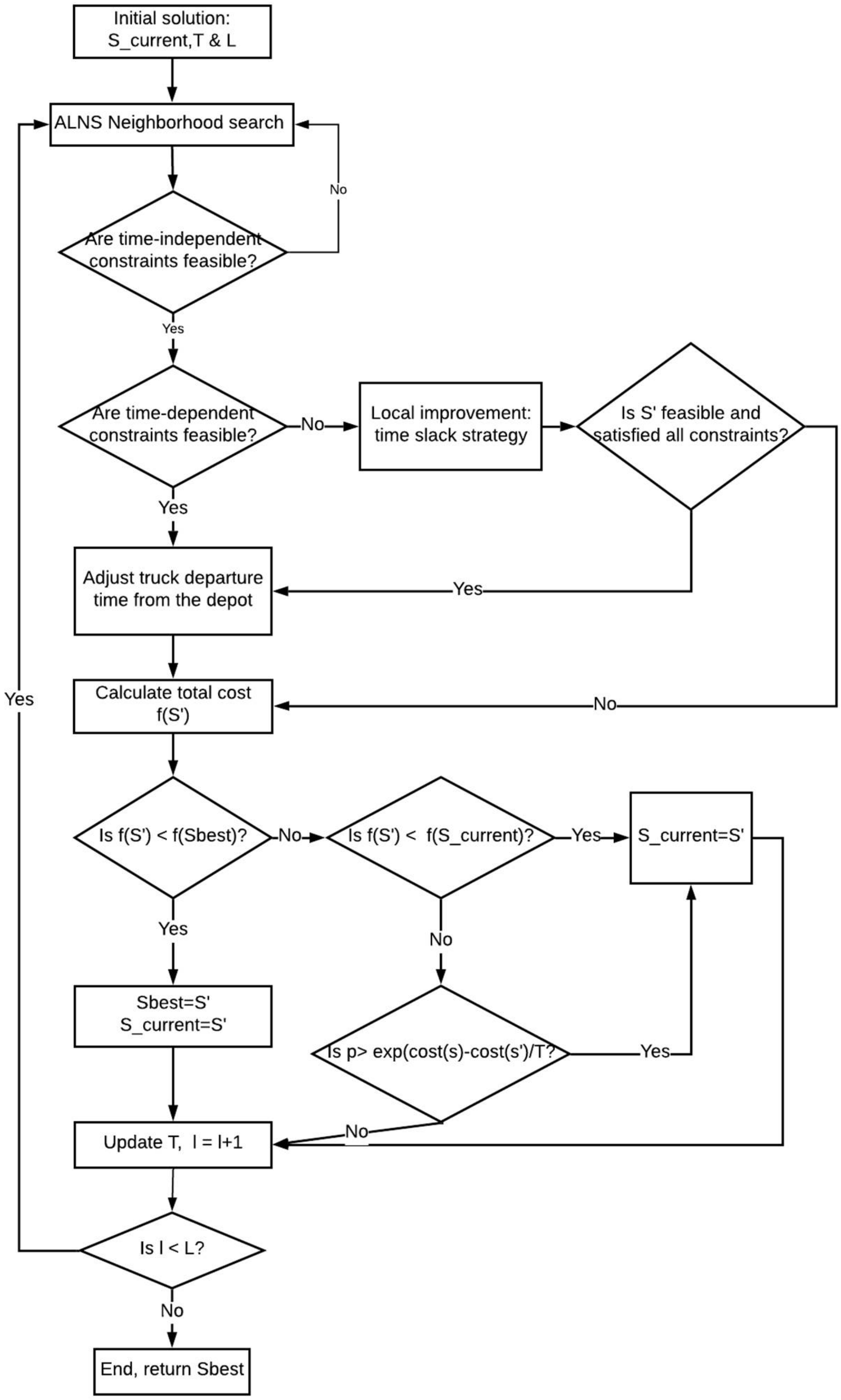

Figure 2: Solution approach flow diagram

4.2.1 Removal Approach

- **Shaw removal:** The Shaw removal randomly chooses a paired request and then requests with the similar characteristics, such as distance, time window or other factors. It is easier to shuffle the similar requests around and creates a new or even a better solution. If the chosen requests are very different from each other, it is possible to gain nothing while reinserting the requests.

  A relatedness measure $R(i, j)$ is used to search the similarity between the chosen request i and request j. Four parameters are included in the relatedness measurement calculation:

  1. pickup and delivery distance difference between the chosen request i and the request j
  2. time window difference between the chosen request i and the request j
  3. the available weight to serve the chosen request i and the request j
  4. the time slack difference that the chosen request i and the request j have at the warehouse

  The weights of different parameters are α_d, β_t, γ_w and δ_{st}, respectively. The algorithm of relatedness measure is given as follows:

$$\begin{aligned} R(i,j) = \alpha_d(|d_{P(i)} - d_{P(i)}| + |d_{D(i)} - d_{D(j)}|) + \beta_t(|Te_{P(i)} - Te_{P(j)}| + \\ |Ta_{P(i)} - Ta_{P(j)}| + |Te_{D(i)} - Te_{D(j)}| + |Ta_{D(i)} - Ta_{D(j)}|) + \\ \gamma_w|W_i - W_j| + \delta_{st}(|ST_{P(i)} - ST_{P(i)}| + |ST_{P(i)} - ST_{P(i)}|) \end{aligned} \tag{4}$$

  $P(i)$ and $D(i)$ indicate pickup and delivery locations of request i, Te denotes the earliest arrival time and Ta is the last arrival time. Hence, $Te_{P(i)}$ is the earliest arrival time of node i at its pickup warehouse, while $Ta_{P(i)}$ is the last arrival time of node i at its pickup warehouse. W_i is the weight of request i and $ST_{P(i)}$ is the time slack of request i at its pickup warehouse. $R(i, j)$ is firstly sorted sequential in an array L, and then employed a random selection to choose the number of related request N_j by the following method (Ropke and Pisinger, 2006).

$$N_j = N_j \cup L[y^p|L|] \tag{5}$$

  A user-defined parameter $p \geq 1$ is used to randomly select the related requests in the list $R(i, j)$. When p has a low value, it introduces more randomness in the selection. On the other hand, for a high value of p, the most related shipment with respect to i is chosen.

- **Random removal:** The random removal algorithm is a simple method that randomly selects r requests and removes them from the current solution of s to a better position. The steps follow by the *Shaw removal* but p is set equal to 1 to achieve a totally random selection. This method is faster than other methods and speeds up the removal of requests process.

- **Worst removal:** The worst removal is used to search the highest decrease while the request i is not served in the solution s. The cost without request i is calculated as $cost(i, s) = f(s) - f_{-i}(s)$. $f_{-i}(s)$ is the cost while the solution without request i. The selected requests and the costs $\{i, c\}$ are placed by an increasing order. It seems reasonable to remove the request that locates the position with the highest cost to other lower cost position, so the highest cost request i is selected.

- **Shortest route removal:** The shortest route removal, as the name suggests, removes the shortest route (intended as the route with fewer shipments) from the current solution to the unassigned shipment list. This approach is added to verify if a better solution can be achieved that uses fewer trucks.

4.2.2 Insertion Approach

- **Basic greedy insertion:** The basic greedy insertion does at most n iteration when inserting the request i into different positions k of a route. The inserted position is chosen at the smallest cost increase $\Delta f_{i,k}$ in the route. The formula is given as follow,

$$\Delta f_{i,k} = f(s') - f_{-i}(s') \tag{6}$$

  If the request i can not be inserted into position, it will stay in the unassigned shipment list. The process stops until all unassigned shipments have been inserted or no more requests can be inserted into any route.

- **Regret insertion:** The regret insertion is used to improve the basic greedy insertion because it only considers the least change after request i is inserted in the position k. The regret insertion looks further information by computing a regret value. The regret value calculates the difference for a request insertion cost between its best position and its second best position. From the basic greedy insertion knowing every $\Delta f_{i,k}$ and sorting them in order, that is, $\Delta f_{i,k} < \Delta f_{i,k'}$ where $k < k'$. The regret value R_i is the request i that inserts at the position k where has the maximum regret value but minimum cost of $\Delta f_{i,k}$. The algorithm 7 shows the formula of the regret value.

$$R_i = \{\Delta f_{i,k'} - \Delta f_{i,k}\} \tag{7}$$

- **Regret-k insertion:**Regret-k insertion is an extension approach from the regret insertion approach that calculates the regret value among other. Instead of calculating the regret value between the best position and the second best position of the request i, it calculates the regret value R_i^* between the solution with the least cost and the $k-1$ best solutions (equation 8).

$$\max_{i \in N}\{\sum_{k=2}^{k} \Delta f_{i,k} - \Delta f_{i,1}\} \tag{8}$$

  The request i is inserted at the minimum cost position while the maximum regret value is found in either regret insertion or regret-k insertion. Regret insertion is included in regret-k insertion while $k = 2$ because it only compares the regret value between the best and the second best solution. While $k > 2$, the regret-k insertion approach investigates the request value between the best and the $k-1$ best routes, it discovers the limitation of a insertion position of a request i earlier than regret insertion. The lower bound of k is 3 and a upper bound of k is given by the number of the routes in the current solution. The maximum regret value is hence chosen to be inserted above all unassigned shipments.

- **New route insertion:** creates a new truck route that accommodates as many shipments as possible among the ones present in the unassigned shipment list.

4.3 Removal and Insertion Approaches Selection

There are different removal and insertion approaches used in the neighborhood search, therefore, two methods can be used in choosing the approach. The first method applies equal probability to select four removal and insertion approaches in every neighborhood search. The second method is a dynamic probability of the selection of the removal and the insertion approaches which is inspired by Ropke and Pisinger (2006). Instead of a equal random selection of the removal and the insertion approach pattern, the weight of different heuristics is based on the previous iteration result and would be adjusted dynamically.

The basic idea of the dynamic probability is to monitor the performance of the removal and the insertion approach. The measurement of the weight is calculated by different result performances in the meta-heuristic method, which means a score is given to the corresponding result. To avoid a bias selection from happening, a number of segments is defined to reset the score. The probability of different neighborhood searches is determined by the performance of the previous segment, and the score would be updated at the beginning of every segment.

Three different scores are given while different performances are achieved. The first criteria of the score σ_1 is defined when the selected removal and insertion approaches are able to find a new overall solution through the neighborhood search. This is the highest score among three different criteria because it brings the result ahead to obtain a near optimal result. The second criteria of the score σ_2 is when a local improvement is achieved via the selected removal and insertion approaches, meaning a new solution is better than the current solution but not better than the best solution. As it is interesting to diverse the search, an acceptance of the worse solution is taken into account as well. The third criteria of the score σ_3 represents an encouragement of a new solution development, which is the least score among three criteria.

The weight of the heuristic performance is calculated at the end of the segment. The mathematical formula of the dynamic probability shows as follows:

$$W_{i,j+1} = W_{i,j}(1-r) + r(\frac{\pi_{ij}}{\sum_i \pi_{ij}}) \tag{9}$$

In this formula, $W_{i,j+1}$ represents the weight of heuristic i in the segment $j+1$, π_{ij} is the total score that heuristic i obtains in the last segment j and r represents a dummy variable. If $r = 0$, meaning heuristic i applies the weight that calculates from the last segment j and $r = 1$ means heuristic i begins the new segment and applies the new weight that obtains from the last segment j.

Apply the dynamic probability in the randomly choose would select the better performed heuristic in order to obtain the solution faster; however, this could not tell whether the selected insertion heuristic or the removal heuristic is more success in the model.

4.4 A Local Improvement: Time Slack Strategy

A time slack strategy is used to adjust while dock capacity constraint is violated. To eliminate its infeasibility, the time slack strategy is employed to reschedule truck arrival time. The basic concept of the time slack strategy is to guarantee every shipment can satisfy its time window constraint by shifting truck arrival time at the pickup or delivery nodes.

The time slack strategy is inspired by Li et al. (2016a), and used the time slack concept to decide which truck to delay in case of a dock capacity violation. This approach uses the information of arrival time A_i^k, beginning of the service time B_i^k, departure time D_i^k, time window $[e_i, l_i]$ and time slack of every shipment. The following Table 1 and Fig. 3 illustrate an example while dock capacity constraint is violated for a period of time. Two docks are available but three trucks are visiting at a freight forwarder's warehouse at different time schedule.

Table 1: Example of time slack strategy while dock capacity constraint is violated

Truck	**A**	**B**	**C**
Arrival time	10	30	40
Departure time	50	80	90
Time window	[5, 20]	[30, 60]	[20, 45]
Time slack	10	30	5

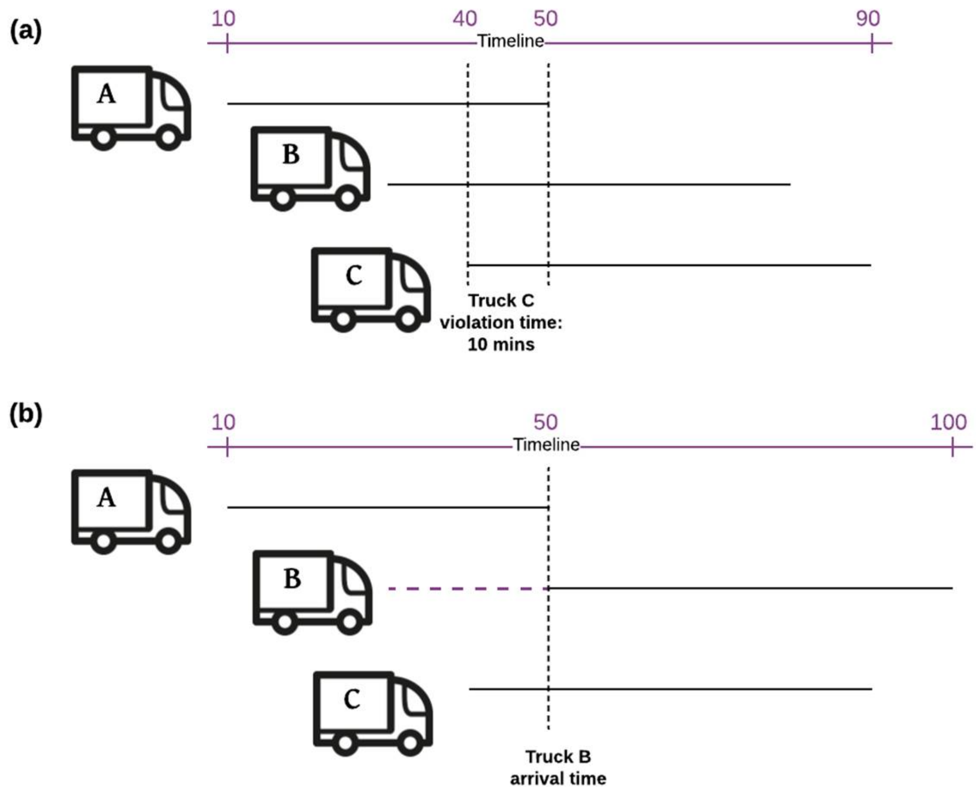

Figure 3: Dock capacity violation resolution by applying the time slack strategy (a) Trucks queuing at the warehouse (b) After the time slack strategy is applied

Table 1 provides the detailed information about arrival time, departure time and time window of each truck to pickup their corresponding shipments. It shows that while both truck A and truck B are still loading the intended shipments, truck C arrives at the warehouse. Therefore, truck C has to wait another 10 minutes for its turn. Due to the shipment pickup time window constraint, a violation would occur if truck C is pushed back 50 minute to pickup the shipment. Realizing that truck B would be characterized by a greater time slack to pickup the shipment, it is more reasonable to shift the arrival time of truck B by 50 minutes. Hence, all trucks can pickup their shipments and satisfy the time window constraints.

While truck B is shifted to arrive later at the warehouse, the downstream schedule of every truck may be affected. Therefore, both time window and dock capacity constraints have to be examined again. However, this strategy may worsen the schedule of all scheduled trucks. A stopping criteria is set to avoid a worse solution from being created.

4.5 Departure Time Adjustment

In general, all trucks are initially assigned to depart the depot as soon as possible. Cordeau and Laporte (2003) showed an approach to delay truck departure time to decrease the total travel time while satisfying time windows and ride time constraints. Given that $B_i^k = max\{A_i^k, e_i\}$ is the earliest possible service time for node i, how much the service at each node can be delayed (i.e., the slack time of node i) depends on the time characteristics of nodes downstream.

The forward time slack F_i^k quantifies the additional time that truck k can delay to start servicing request i before violating a time window or ride time constraint. The formula to determine such value is shown in Eq. 10, where q is the index of the last node in the current route, i.e., the destination depot. The formula contains the minimum of the total waiting time until shipment i, plus the difference between latest arrival time and the beginning of service time of shipment i (Cordeau and Laporte, 2003).

$$F_i^k = \min_{i \leq j \leq q} \{ \sum_{i < p \leq j} W_p + (l_i - B_j^k) \} \tag{10}$$

Devising a strategy that is based on F_i^k improves the quality of the solution by reducing total transport costs. Because the local improvement might worsen a feasible solution by introducing a time window violation or a new dock capacity violation downstream, the procedure to resolve dock capacity violations is stopped if the potential resolution of the first dock capacity violation would introduce an additional infeasibility. Note that, for every dock capacity violation that was resolved, the delayed truck would incur in an additional delay $Delay_i^k = B_i^k - (A_i^k + W_i^k)$. A last step is carried out to verify if the associated truck can have its departure pushed back by the same quantity, in order to avoid the dock capacity violation without increasing its travel time. The whole procedure is shown as follows:

1. Check if $Delay_i^k > 0$
2. Determine the position of $Delay_i^k$ in the visiting warehouse
3. Set $D_0 = e_0$
4. Compute A_i^k, W_i^k, B_i^k and D_i^k for every shipment i in truck k
5. Compute F_0^k for truck k
6. Determine the sequence of the visiting warehouse
7. Set $D_0^k = \min\{F_0^k, Delay_i^k, Delay_{i-1}^k\}$
8. Update A_i^k, W_i^k, B_i^k and D_i^k for every shipment i in truck k
9. Compute the feasibility of the solution after the change

The additional delay $Delay_i^k$ is checked in step (1) and then the sequence of the shipment i at the visiting warehouse is check in step (2). Note that, since we are delaying the docking operations of a truck, the shipment where the delay is actually applied is the first one in the intended ground handler.

Steps (3) to (5) determine the time stamps and the minimum forward time slack of the route until shipment i. Step (6) checks the sequence of the visiting ground handlers to determine if a departure delay of the current truck might create new dock capacity violations upstream, which is clearly an undesired effect.

Step (7) computes the minimum delay departure time that can applied without increasing a time window violation or creating a new dock capacity violation upstream. Note that a new dock capacity violation can be generated, only if the current truck is delayed at a ground handler that is not the first one in the delivery sequence. If previous ground handlers are visited, the maximum departure push back time for the current track is the minimum among (i) the slack time of all nodes preceding shipment i, and (ii) the maximum delay the truck can incur in all previous ground handler before generating an unplanned dock capacity violation. As mentioned before, if the dock capacity that was resolved occurs in the first ground handler visited, only option (i) is considered.

5 Computational Experiments

To provide a more specific and tangible assessment of the truck congestion in landside operations, Amsterdam Schiphol Airport is used as a reference. The layout of Schiphol Airport and the location of five freight forwarders and five ground handlers is presented in Fig. 4. All instances presented here are based on this layout and on subsets of the presented warehouses. The locations of freight forwarders and ground handlers are based on the real company locations, the orange rectangles define the location of ground handlers and the blue rectangles define the location of freight forwarders. The location of the central depot is chosen at the red rectangle. It is sufficiently close to all warehouses and a building-free area. Note that how to assess what is the best location of the central depot, which would be an interesting research topic, but it is beyond the scope of this work. In addition, the central depot and the fleet of trucks are assumed to be owned by the airport. Therefore, it is assumed that the airport is responsible for planning the truck routes.

The number of docks does not represent the overall number of docks available at each ground handler's side, it represents the number of docks that each ground handler reserves for trucks serving the consortium of freight forwarders. The available dock capacities are different in each ground handler. The number of the available docks is [1, 2, 1, 1, 2] from ground handler 1 to ground handler 5, respectively.

Figure 4: Location of the five freight forwarders, of the five ground handlers, and of the central depot for the case study (background figure from Google Maps).

The distance and time related transport costs are based on Schonewille (2015) research, where the travel distance unit $Cd_{ij} = 0.66$ and the travel time unit $Ct_{ij} = 0.6$. Other parameters that are fixed in the model is: the initial temperature $T = 50$, the number of removal request $r = max\{5, \textit{number of unassigned shipments}\}$

and the cost that shipment i is not picked up by any truck $\alpha = 300$.

In this section, computational results will be shown. First, a brief sensitivity analysis that addresses the weight of the infeasible contribution of the cost function is carried out. Then, a comparison between an equal probability and a dynamic probability in the selection process of the removal/insertion moves is shown. Finally, a comparison between the solution of the MILP model and the meta-heuristic method is provided for a set of 10 instances.

5.1 Parameters Sensitivity

The infeasibility of a time-dependent constraint is allowed in our meta-heuristic method to explore and enrich the search space. In fact, exploring a neighborhood of such a solution might identify a feasible new best solution. While focusing on solutions characterized by a time-dependent infeasibility is allowed during the search, but the final output of the meta-heuristic method should be a feasible solution, if such a solution exists. As a consequence, for a specific instance it was assessed how the weight of the time-dependent infeasibility affected the final outcome.

We focused on a case where 50 shipments must be delivered from freight forwarders 1 through 5, to ground handlers 3 and 4. We tested 3 different weights for the time-dependent infeasibility, i.e., 1, 10, and 100. Results in terms of best solution cost, infeasibility of the best solution, and computational time are reported in Table 2.

Table 2: Result of different weights of time-dependent constraints parameter

parameters of time-dependent constraints	**total cost**	**travel time [mins]**	**infeasibility of time-dependent constraints [mins]**
1	814.6	1156.0	69.8
10	841.3	1314.2	0
100	841.1	1293.6	0

As highlighted in Table 2, when the weight is set equal to 1, the best solution is the lowest among the three, but it is not feasible. This is an undesired result we want to avoid. On the other hand, when the weight is set to 10 or 100, the best solution is the same, the time-dependent constraints are infeasible while explored during the search, but is not present in the best solution. While for this particular case, a weight of 10 was already sufficient to avoid infeasibility, we decided to use a weight equal to 100 in all computational analyses.

5.2 Comparison between Equal Probability and Dynamic Probability

Section 4.3 discussed two different methods to choose removal and insertion approaches in a neighborhood search: (1) select a removal approach and an insertion approach with an equal probability (2) select a removal approach and an insertion approach with a dynamic probability based on the previous performance. A comparison between the equal probability (non-adaptive strategy) and the dynamic probability (adaptive strategy) is reported in Fig. 5. The input data is as same as the previous case study, the parameter sensitivity experiment. In addition, parameters of time-dependent violations are set equal to 100.

The blue line indicates the result of applying the dynamic probability in the case study, and the red line indicates the result of the probability remaining unchanged. Although the two approaches eventually converge, it is clearly shown how the dynamic probability outperforms the equal probability and reaches the local minimum faster. Although we showed the comparison for a specific case and did not perform an exhaustive analysis, we decided to use the dynamic probability approach in all the computational results shown next.

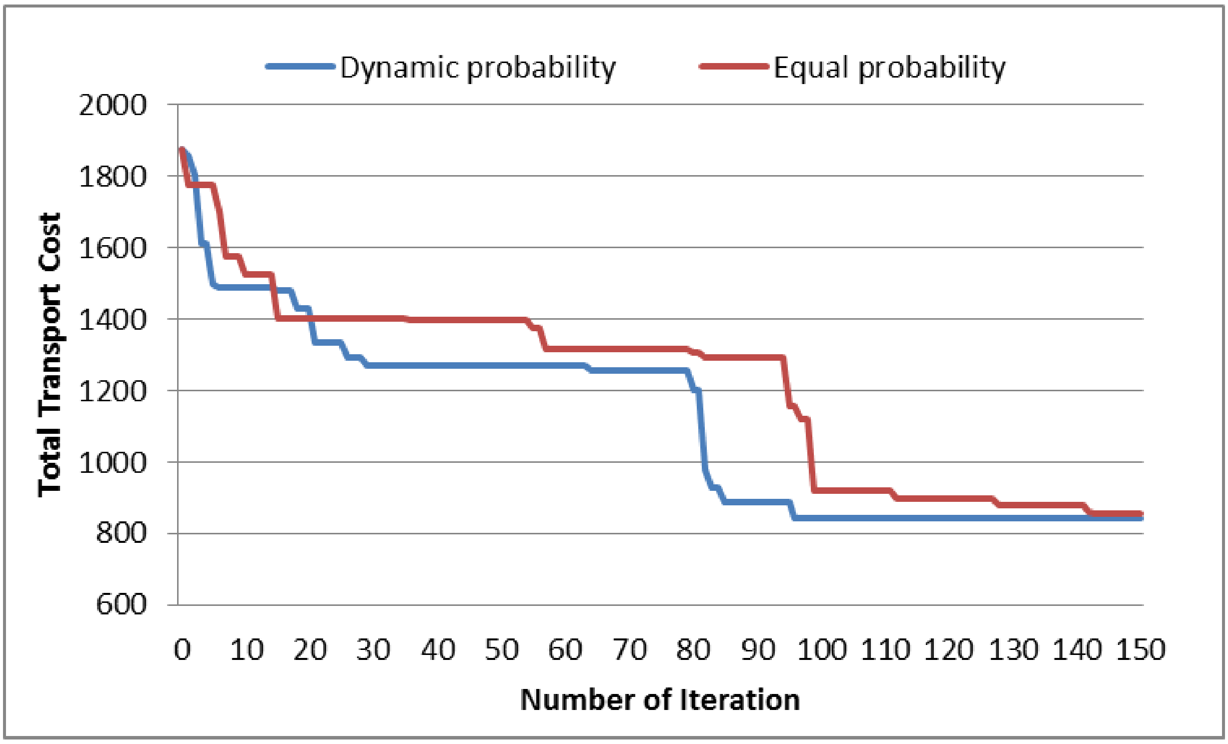

Figure 5: Comparison between equal probability and dynamic probability

5.3 Comparison between the Exact Model and the Meta-heuristic

In order to provide a benchmark time in our comparison, a time limit of 120 minutes (2 hours) is set when computing the solution of the MILP model. Three outcomes are possible for each MILP instance. (1) no solution was found within the time-limit, (2) an incumbent solution was found, with a non-zero gap optimality, and (3) the optimal solution was found (i.e., gap optimality is 0%).

The parameters for the time-dependent constraints are set equal to 100, and the dynamic probability approach is applied in the neighborhood search. In the FF column we report the freight forwarders considered in each instance, while in the GH column we report the associated ground handlers. The locations of the warehouses can be retrieved in Fig. 4.

Table 3: Computational result of 10 instances in Schiphol Airport

Instances	Total shipments	FF	GH	Total cost of heuristics method	Computing time [mins]	Total cost of exact model	Computing time [mins]	Gap [%]
1	15	F1 & F2	G1 & G5	236.8	2.3	240.6	120	-1.58
2	15	F1 & F2	G3 to G5	203.5	1.5	203.9	120	-0.17
3	20	F3 & F4	G1 to G4	315.0	3.3	309.1	120	1.91
4	25	F4 & F5	G1 to G5	397.7	6.2	386.7	120	2.83
5	20	F3 to F5	G4 & G5	279.0	2.9	279.9	120	-0.35
6	30	F3 to F5	G3 to G5	428.4	10.6	432.3	120	-0.91
7	35	F1 to F3	G2 to G4	564.8	15.2	533.6	120	5.86
8	40	F1 to F3	G1 to G4	668.5	20.5	N/A	480	-
9	50	F1 to F5	G1 to G5	865.2	44.8	N/A	480	-
10	55	F1 to F5	G1 to G5	927.4	54.9	N/A	480	-

Results for the 10 instances are shown in Table 3. It is evident how the meta-heuristic method is effective in producing comparable, or even better solutions than the exact model in a shorter time. In particular, we believe that this solution is quite close to the best solution for all situations in which an existing solution is found. Unfortunately, the optimality gap is still very high because a very slow growth of the lower bound was experienced.

For 7 of the 10 instances presented, an incumbent solution was found, while for 3 instances no feasible integer solution was found. We highlighted the no feasible integer solutions within the specified time-limit by reporting N/A (not available) in the associated column. Instances 1, 2, 5, and 6 show a better solution in the heuristic model, while for instances 3, 4, and 7 the incumbent solution of the exact model is lower than the outcome of the meta-heuristic method. The maximum gap, intended here as percentile difference between the MILP and heuristic solutions, is 5.86% for instance 7, which still supports the good performance of the devised algorithm.

6 Conclusions

This paper has proposed a SA-embedded ALNS algorithm to achieve a near-optimal solution for the truck route design in the landside air cargo supply chain. Constraints are main assumptions were inherited from a mathematical formulation that addressed the same problem. Within the SA framework, the choice of the best removal and insertion moves was based on a dynamic probability selection process, which proved to be more efficient than a static approach. In addition, we allowed infeasible solutions (and their neighborhoods) to be explored, but showed that infeasibility should be properly weighted to avoid the routine from outputting a final solution that is not entirely feasible. Given a set of test instances, the solutions generated with the meta-heuristic method are equal or superior than the ones produced by the exact model in most cases. Even if the solution provided by the exact model was not the optimal (which means that the comparison could favor the exact model if run with a longer maximum runtime), the meta-heuristic method obtained better or only slightly worse solutions in a shorter time. In addition, the SA-embedded ALNS algorithm was able to find a feasible solution for those cases where the exact model fails to identify a feasible solution.

References

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.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167 – 173.

Baños, R., Ortega, J., Gil, C., Fernández, A., and de Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, 40(5):1696 – 1707.

Bombelli, A. and Tavasszy, L. (2018). A pickup and delivery formulation for landside air cargo supply chain optimization. (unpublished).

Buso, F. (2017). Analyzing the "milkrun" service process for inbound freight at schiphol airport with queuing theory and simulation. Master's thesis, University of Mannheim.

Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579 – 594.

Hosny, M. (2011). Comparing genetic algorithms and simulated annealing for solving the pickup and delivery problem with time windows.

ICAO (2016). Moving air cargo globally. `https://www.icao.int/Security/aircargo/Moving%20Air%20Cargo%20Globally/ICAO_WCO_Moving_Air_Cargo_en.pdf`. Accessed: April 2019.

Li, B., Krushinsky, D., Woensel, T. V., and Reijers, H. A. (2016a). An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research*, 66:170 – 180.

Li, H. and Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186.

Li, Y., Chen, H., and Prins, C. (2016b). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1):27 – 38.

Max, v. I. (2018). Amsterdam airport schiphol doing a case study about creating key performance indicators for measuring the processes of the ground handlers. Master's thesis, Amsterdam University of Applied Science. `https://acn.nl/wp-content/uploads/2019/01/Max_van_Iersel_KPIs_Ground_Handlers_Schiphol.pdf`.

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. cited By 555.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, 4(1):285–305.

Schonewille, G. (2015). Calculation of transport cost for freight carriers on the last mile. Master's thesis, Delft University of Technology, Master thesis.