Department of Behavioural, Management, and Social Sciences
Industrial Engineering & Management
Master's Thesis

Optimizing Route Efficiency In Dry Bulk Logistics: Reducing Empty Kilometers

Esmee Nijhof

First supervisor: Alessio Trivella
Second supervisor: Sebastian Piest
Company supervisor: Jordi Lenselink

June, 2025

UNIVERSITY OF TWENTE. NIJHOF WASSINK

This thesis is written as part of the graduation assignment for the master's program of Industrial Engineering and Management at the University of Twente. The graduation assignment is conducted at Nijhof-Wassink.

University of Twente	Nijhof-Wassink
Industrial Engineering and Management	Application management
Drienerlolaan 5	Galvanistraat 18
7522NB Enschede	7461 JC Rijssen
+31 53 489911	+31 548 538 000

Publication date: June, 2025
Number of pages excluding appendices: 54
Number of pages including appendices: 77
Number of appendices: 9

Author:
E.Y. Nijhof,
Master student Industrial Engineering and Management,
University of Twente

First supervisor University of Twente:
Dr. A. Trivella
Assistant Professor of Operations Research

Second supervisor University of Twente:
Dr. J.S.P Piest
Assistant Professor University of Twente

Supervisor Nijhof-Wassink:
J. Lenselink
Teamlead Application Management

Preface

You are about to read my master's thesis: "Optimizing Route Efficiency in Dry Bulk Logistics". This thesis marks the conclusion of my master's in Industrial Engineering and Management, as well as my time as a student. Spending the last seven years at the University of Twente has been a great experience.

I want to thank my first supervisor, Alessio Trivella. He was always open to brainstorming and always helped me see the bigger picture. Of course, I would also like to thank my second supervisor, Sebastian Piest, for his valuable feedback and fresh perspective.

Additionally, special thanks to Jordi Lenselink from Application Management at Nijhof Wassink. Every week, we had valuable discussions about the goal and scope of this research. Next, I thank everyone at Nijhof Wassink who contributed to this research. In particular, when Jordi was on leave, Luca Overmars guided me through the last phase and the entire Application management team in making the office a welcoming place, especially during our lunch walks.

Last but not least, I would like to thank my friends and family for their support during this time.

Esmee Nijhof
Enschede, June 2025

Management Summary

This research was conducted at Nijhof Wassink, specifically within the Dry Bulk Logistics (DBL) department. They specialize in the transportation of silo-based goods, such as plastic granulates. The department operates with approximately 100 vehicles and executes around 400 requests per week. A source of operational costs is the distance driven without a load, known as empty kilometers. This research aims to optimize routing efficiency by reducing empty kilometers, leading to the following main research question: "How can the Dry Bulk Logistics planning at Nijhof Wassink be optimized to reduce empty kilometers?".

Problem Context
The current planning process relies heavily on manual planning decisions, which are time-consuming and prone to errors. No automated routing optimization tools are currently in use. Therefore, managing the complexity of requests and vehicle constraints can be a challenge. As a result, the existing approach is prone to a high number of empty kilometers and other routing inefficiencies. A context analysis highlighted the desire for a standardized and scalable planning approach.

Solution Design
To address these challenges, a vehicle routing algorithm was developed to generate weekly routing plans that respect practical constraints, including time windows, transfers, and driver regulations. The routing problem is formalized as a Multi-Depot, Pick-up, and Delivery Problem, with Time Windows and Transfers (MD-PDPTW-T). Due to its complexity, a two-phase heuristic approach was implemented. The first phase uses a fast constructive heuristic to create an initial, feasible routing solution. The second phase applies an Adaptive Large Neighborhood Search (ALNS) algorithm to iteratively improve a routing solution. The model balances multiple objectives. While minimizing empty kilometers is the key priority, other components, such as total travel distance, time window violation, unserved requests, and vehicle utilization, were added to avoid impractical and unrealistic routing solutions.

Results
The heuristic solutions were compared to historical planning data and the initial, constructive routing solution. Scenario analyses further explored the effect of input parameters on routing performance. The key findings are as follows:

- 10% reduction in overall planning cost compared to historical routing plans.
- 26% reduction in the overall planning cost compared to the constructive, initial solution.
- Improvements were mainly driven by better vehicle utilization, not reduced empty kilometers.
- Solution improvements were achieved by an optimized vehicle utilization.
- A trade-off was identified between reducing empty and total distance and an increase in unserved requests.
- A clear relationship was found between reducing the number of vehicles and an increase in unserved requests.
- Flexible time windows significantly improved routing flexibility.
- No consistent correlation was found between late received orders and routing efficiency.

Practical Contribution

This study provides several practical contributions. First, the algorithm shows that it effectively improves routing efficiency, which offers the potential for reduced operational costs. Although minimizing empty kilometers was the primary objective, results showed limited improvement for this Key Performance Indicator (KPI). Instead, the main improvement was achieved through better vehicle utilization. Therefore, it is recommended to shift the focus toward vehicle utilization to save operational costs. Second, the algorithm demonstrates its ability to create concept routing plans with improved or similar performance to historical routing plans. While the model is not yet suitable for direct integration into the daily planning process, it can already support strategic and tactical decisions in tender evaluation and post-analysis. Third, this research provides a foundation for developing a more standardized, data-driven, and less resource-intensive planning process. For full practical implementation, further development is needed to incorporate additional real-world constraints.

Table of Contents

1| Introduction

This thesis is conducted at Nijhof Wassink and aims to reduce empty kilometers in the Dry Bulk Logistics (DBL) planning. This chapter will introduce the research. Section 1.1 provides the background of Nijhof Wassink and the research motivation. Furthermore, Section 1.2 describes the problem identification, followed by Section 1.3, which outlines the research design and research questions. Lastly, Section 1.4 defines the scope of the research.

1.1 Background

1.1.1 Company Background

A brief introduction to the company for which this thesis is conducted is provided for a broader context. Nijhof Wassink specializes in bulk transport across multiple modalities, such as road, rail, and water [44]. The company focuses on end-to-end logistics, meaning that it coordinates the entire logistic process to ensure the transport of loads. The company employs around 1000 employees and operates mainly in the Netherlands, Germany, Belgium, Poland, and Hungary. The main office is located in Rijssen, the Netherlands. The company's expertise lies in animal feed logistics, chemical logistics, and warehousing. To manage its various operations, Nijhof Wassink is organized into four business units: Animal Feed Distribution, Dry Bulk Logistics, Liquid Bulk Logistics, and Fuel Distribution.

This research specifically focuses on Dry Bulk logistics. This business unit specializes in silo transport of dry bulk, such as clean granulates and powders. The most common are plastics used in everyday objects like polyvinyl (PVC), polyethylene (PE), polypropylene (PP), and polyethylene terephthalate (PET). The company has an international reach, with a main focus on transporting loads in Germany, Poland, and the Netherlands. Furthermore, customers request Nijhof Wassink to transport loads from location A to B as a one-stop shop. The planners are responsible for planning as many requests as possible per vehicle while respecting the driver, product, and customer constraints. The DBL department relies on manually constructed routing plans and aims to explore possibilities to enhance this process. The department desires to provide planners with tools to make the best possible, data-driven decisions.

1.1.2 Research Motivation

The goal of the research is to minimize empty kilometers. This is the distance that trucks travel without carrying any load. Reducing empty kilometers will generally reduce Nijhof Wassink's operational costs, and therefore, it can offer more competitive transport rates for customers. This can lead to more business opportunities as (potential) customers are attracted to lower prices. Additionally, as the number of requests and customers increases, the probability that the customers' locations lie closer together increases. In this way, it is easier to plan the next request in the vicinity, reducing empty kilometers even further. This positive feedback loop is in line with Nijhof Wassink's goal of maximizing profitability and expanding its customer base. This ultimately strengthens its market position.

Next to the commercial benefits, the importance of reducing the environmental impact has increased in recent years. Nijhof Wassink aims to reduce its CO_2 footprint by 10% in 2025 and by 30% in 2030. Therefore, reducing empty kilometers directly supports these sustainability goals. Furthermore, under the Corporate Sustainability Reporting Directive (CSRD), larger companies in the European Union are required to report their greenhouse gas emissions, of which CO_2 is the primary contributor. This includes both direct and indirect emissions through suppliers and distributors. As a result, Nijhof Wassinks (potential) clients value working with transport companies that minimize emissions, next to offering competitive rates. By reducing empty kilometers, Nijhof Wassink not only cuts its emissions but it also helps the company to provide an attractive sustainability report to its clients. To conclude, Nijhof Wassink is interested in reducing its empty kilometers as it may create more business opportunities due to more competitive rates and more attractive sustainability reports.

1.2 Problem Identification

1.2.1 Problem Cluster and Core Problem

Figure 1.1 provides a schematic overview of the problem cluster, outlining the action them. The flowchart helps to identify the underlying problems and illustrates the relationships between the problems. Below, some of the underlying problems will be discussed, and a motivation for the chosen core problems is provided.

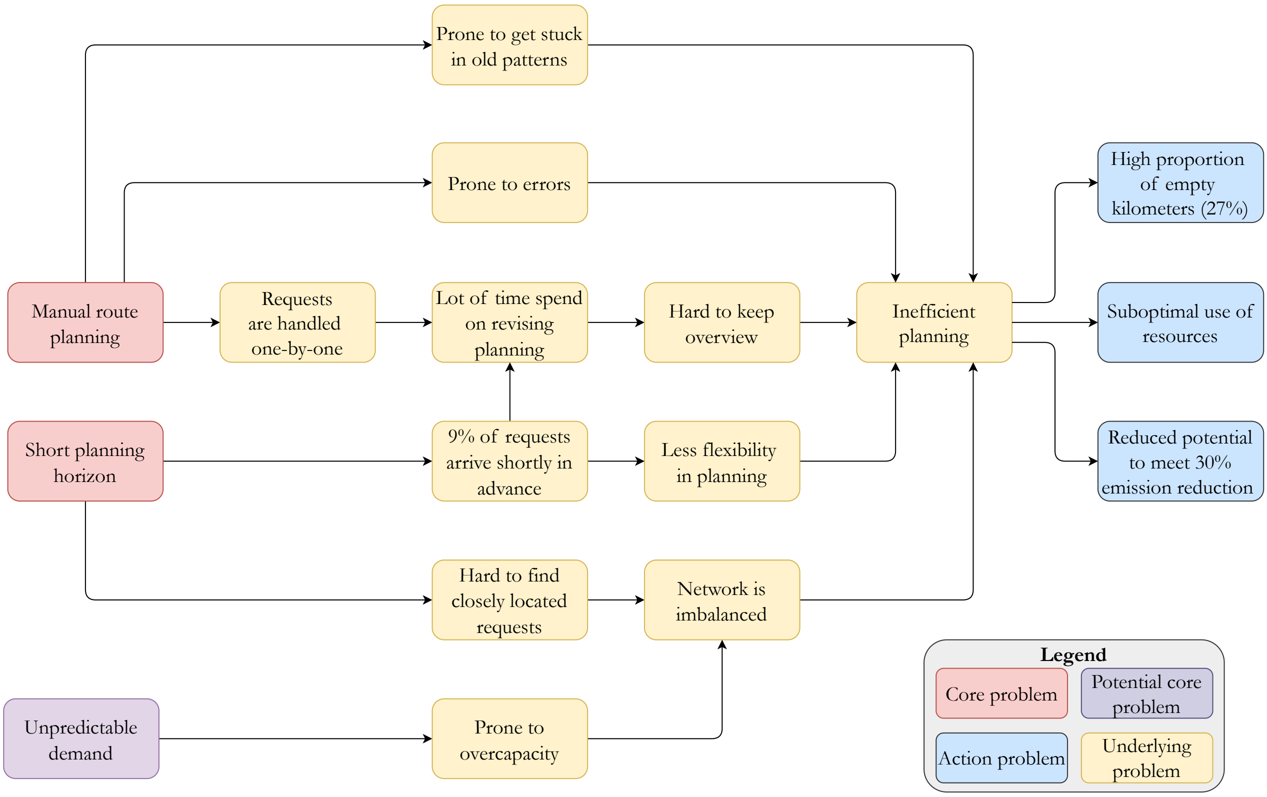


Figure 1.1: Problem cluster.

Many empty kilometers
According to the post-analysis by Nijhof Wassink, it has become evident that the load factor in 2023 was 73%, meaning that 73% of the total kilometers were driven with a load. This is equivalent to stating that 27% of the driven kilometers trailers were driven empty. The company has a strong desire to reduce empty kilometers to lower the operational costs, which increases business opportunities, and decreases emissions. The underlying problem of empty kilometers stems from inefficiencies in the routing plan.

Suboptimal use of vehicles
Nijhof Wassink also suspects that its vehicles are being used suboptimally, which stems from inefficient planning. Ineffective routing can result in underutilized vehicles, meaning that more vehicles are required to complete the same volume of work. As a result, the company may miss business opportunities. Moreover, operational costs increase, ultimately limiting the ability to offer competitive pricing.

Emission reduction
Nijhof Wassink aims to reduce its CO_2 emissions by 30% by 2030. However, reaching this goal is at risk due to inefficient planning. Unnecessary kilometers and suboptimal route optimization increase emissions without adding operational value, ultimately undermining the company's progress toward sustainability goals. As a result, the company faces a reduced ability to create attractive sustainability reports for its customers.

Inefficient planning
Nijhof Wassink uses planning software from Ortec to track incoming requests and the routing plan for each truck. While Ortec offers automated planning functionalities, these are currently not implemented at Nijhof Wassink. Instead, planners have to manually assign each request to a vehicle and a driver. The company often operates with inefficient routing plans due to several reasons. Due to the large number of requests and trucks, it is hard to keep an overview of the routing plan as a whole, and planners are likely to fall into old planning patterns instead of trying to find an improved solution. Also, there is a risk of unnoticed human errors in the planning process, and no available tool exists to detect them. Furthermore, some requests arrive shortly before they need to be executed. This, combined with the numerous planning constraints, such as the availability of trucks and drivers, required cleanings, required product handling, and customer time windows, results in limited flexibility in the routing plan. These factors contribute to inefficiencies in the routing plan.

Revising the routing plan
Planners continuously review the list of unplanned requests and assign each request individually to a vehicle. When new requests arrive, planners attempt to assign them to an available vehicle, or they must reconsider the already planned vehicles to make an optimal routing choice. Additionally, some requests need to be planned at the last minute, often requiring the current routing plan to be reconsidered and adjusted to satisfy all constraints. A negative consequence of this continuous planning approach is the large amount of time spent revisiting and revising the routing plan.

Network imbalance
Requests have a start and end location and usually cover a large distance. To reduce empty kilometers, it is ideal to have a request with a starting location close to another request's end location. This is called a balanced network. If the network is imbalanced, pick-up and delivery locations have a widespread. This is partially caused by the short planning horizon, as there is limited time to identify pairs of requests that lie close together. Furthermore, unknown demand is prone to overcapacity of available resources. This causes underutilization of the resources and may lead to unnecessary traveled distance, as the planning department prioritizes using all available resources, even if that implies suboptimal routes.

Core problems
From this problem cluster, three core problems can be identified: manual route planning, a short planning horizon, and unpredictable demand. This thesis will focus on the first two, as the planning department is expected to have the most influence in these areas. Currently, the manual route planning approach relies solely on the planners' experience and intuition. This approach may lack efficiency and often results in more empty kilometers. Additionally, the current planning horizon is 1 or 2 days. Within this short time frame, there is limited time to optimize the routes and there is limited flexibility when selecting which requests to pair together. This can result in suboptimal routing decisions. To address these issues, automatically constructed routing plans with a longer planning horizon and with an aim to reduce empty kilometers can help to optimize truck routing.

As for the third core problem, unpredictable demand is caused by the agreements that Nijhof Wassink has made with its customers. These constraints allow customers to place a certain number of requests within a specified period. However, the agreements do not specify when the requests must be made or if they must be made periodically. Due to the loosely defined timing of requests, it is challenging to predict the number of requests without adjusting the approach of constructing contracts with customers. The planning department has little influence on this process and is therefore not chosen as a core problem.

1.3 Research Design

As Nijhof Wassink seeks to reduce empty kilometers to improve its market position, there is a need to improve the routing plan. Hence, the main research question of this thesis is formulated as follows: *How can the Dry Bulk Logistics planning at Nijhof Wassink be optimized to reduce empty kilometers?* To answer the main research question structurally, the different chapters of the thesis will explore a part of the main question. For this, each chapter has a sub-research question. To help answer each sub-research question, it is divided into a set of smaller questions. Together, they provide a framework for the research. Additionally, a schematic representation of the research questions and the input and output per chapter is shown in Appendix A.

Ch 2: Context Analysis

This chapter analyzes the current planning approach at Nijhof Wassink's DBL. This includes understanding what request and resource constraints the planners have to respect and what methodology the planners use to minimize empty kilometers. Also, a stakeholder analysis is done to illustrate the parties and their impact on the planning process. Furthermore, we need to formulate certain parameters that can indicate how good a routing plan is, the so-called key performance indicators (KPIs). Lastly, the challenges that the planners currently face when constructing a routing plan will be analyzed. The information will be obtained by interviewing the company's planning department and other experts. This yields the following sub-research questions and sub-questions.

RQ: How is the current planning process organized?

a) Who are the stakeholders involved in the planning process?
b) What are the characteristics of the current planning approach in DBL?
c) What are the limits of the current planning approach?
d) What are relevant KPIs to determine the quality of the routing plan?

Ch 3: Literature Review

After the analysis of the current situation, an extensive literature review is done to explore already existing research in the transport sector of Dry Bulk. Specifically, we are interested in classifying the problem in this context. The review aims to summarize relevant optimization models and solution approaches that align with this company's problem. This will yield a theoretical foundation for solving the problem in the next chapters.

RQ: What are the most suitable methods for developing an optimization model in the context of Nijhof Wassink's DBL planning?

a) Which theoretical transport problem resembles most with the DBL planning problem?
b) How can the characteristics of the DBL planning be modeled?
c) What are solution methods to solve the optimization model?

Ch 4: Problem Description and Mathematical Formulation

This chapter provides a formal description of the problem and presents a mathematical formulation of the Nijhof Wassinks DBL optimization problem. This formulation, combined with modeling assumptions, ensures a universal understanding of the problem at hand. Furthermore, the problem is modeled as a Mixed Integer Problem (MIP) and serves as an exact approach to solving the problem. However, the feasibility of the exact approach needs to be investigated, as large instances may be too computationally expensive.

RQ: How can the DBL planning problem be described and formulated mathematically?

a) What are the modeling assumptions?
b) What is the theoretical formulation of the routing problem?
c) What are the limitations of this exact method?

Ch 5: Heuristic Approach
Given the computational challenges of solving large instances with an exact approach, this chapter introduces a heuristic solution for the DBL planning problem. This chapter explains the design logic of the proposed heuristic and explains how the objective and constraint from the mathematical model can be identified. The goal of the heuristic is to provide a good-quality routing solution within a reasonable computation time.
RQ: How can a heuristic solution be designed to optimize the DBL planning?

a) How does the heuristic solution address the specific characteristics of the DBL planning?
b) How can this heuristic solution be applied to solve this problem?

Ch 6: Experiments and Results
After the solution design, we can start to experiment with the solutions. Firstly, the preprocessing steps of the input data are explained, and the experiment parameters are tuned. Different experiments result in routing solutions. Their quality is compared to the historical planning data. Nijhof Wassink's DBL planners are involved to help assess the routing solutions. Furthermore, the impact of the computational time and the solution quality will be compared.
RQ: What are the model outcomes, and what is the overall performance?

a) How should the input data be prepared for the optimization model?
b) What parameter settings have the best performance on the solution?
c) To what extent does the solution improve the current planning process?
d) What is the trade-off between computational time and solution quality?

Ch 7: Conclusions, Contributions and Recommendations
Based on the solutions found in the previous chapter, this chapter evaluates the research findings and the practical implications for Nijhof Wassink. It provides recommendations for improving their routing efficiency.
RQ: What are the conclusions and recommendations that can be given to Nijhof Wassink?

a) What conclusions can be drawn from the results?
b) What are the recommendations for Nijhof Wassink?
c) What are further research possibilities?
d) What are the practical and academic contributions of this paper?

1.3.1 Research Layout

Based on the framework above, the layout of the thesis will be as follows. Chapter 2 describes the analysis of the current situation of the DBL planning at Nijhof Wassink. This provides insights into the current planning methodology, constraints, parameters, and current issues. After understanding the current situation, Chapter 3 elaborates on the existing literature about DBL planning and provides a theoretical foundation for the remainder of the research. Subsequently, in Chapter 4, the problem is formally described and modeled as a MIP. Chapter 5 provides a heuristic solution to address large planning instances. Subsequently, experiments on the model are performed, and the quality of the solutions is determined in Chapter 6. Finally, Chapter 7 evaluates the research findings and the practical implications for Nijhof Wassink.

1.4 Scope

This thesis aims to provide a practical contribution to the context of Nijhof Wassink's DBL routing plan. The findings are only applicable to the Dry Bulk department of Nijhof Wassink and are solely focused on long-term planning. This means that it is focused on the routing plan that is made ahead of time rather than last-minute problem-solving in response to disruptive events. Furthermore, the findings are intended to demonstrate the improvement potential of using a model-based planning tool. The thesis, however, does not provide a functional planning tool that is ready to be implemented. Instead, it attempts to find evidence that such a tool can help to support the planning department to make the best possible routing decisions. This offers a starting point for further development and application within the company.

The key deliverables are the mathematical model based on Nijhof Wassink's context, a proof of concept demonstrating how a model-based approach can influence their planning efficiency and a set of recommendations regarding how Nijhof Wassink can use these results in their current context. Furthermore, this thesis is limited to 20 weeks, following after the approval of the project plan.

2| Context Analysis

This chapter analyses the current planning process of DBL at Nijhof Wassink to address the first research question: *"How is the current planning process organized?"*. Firstly, the parties involved in the planning process, and their influence will be explained in a stakeholder analysis in Section 2.1. Furthermore, the overall procedure of planning requests is explained in Section 2.2. This also includes a data analysis of the current resources and planning decisions. Lastly, the KPIs that determine the quality of the routing plan are discussed, as well as the performance of the current routing plan 2.3.

2.1 Stakeholder Analysis

The stakeholders involved in the planning process at Nijhof Wassink are the customers, the management, and the planning department. They have various interests in this transport process and the relations between them are explained below.

The Customers
The customers have a wide selection of transportation companies to choose from, making cost efficiency, service reliability, and sustainability key factors in their decision-making process. Customers desire to have affordable transport rates to increase their profitability, as well as reliable and timely pick-ups and deliveries to ensure a smooth supply chain. Additionally, transportation companies that emit minimal CO_2 emissions can reduce the customer's carbon footprint. Based on these factors, customers may choose Nijhof Wassink as their transportation company. This influences Nijhof Wassink's request demand.

The Management
Nijhof Wassink's management competes in a highly competitive market and aims to attract potential customers. This is done by offering the best affordable rates while maintaining a high service level. They aim to increase business opportunities and expand their market share. To achieve this, they are continuously looking for ways to reduce their operational costs and carbon footprint. The management's strategic choices have a high influence on the competitiveness of the transport market.

The Planning Department
The planning department works 6 days a week to ensure that the DBL planning process runs smoothly. Their goal is to plan all the customer's pick-ups and deliveries within the given time windows while respecting the request, product, and resource constraints. Additionally, they are responsible for planning optimal routes to minimize empty kilometers, which is a key factor in reducing operational costs and improving efficiency. Besides planning ahead, the planning department faces many ad hoc challenges with last-minute delays due to traffic congestion or loading delays. This can make balancing a proactive, ahead-of-time routing plan and last-minute problem-solving challenging. All in all, the planning department also has a high influence on the operational efficiency of the DBL planning process.

To conclude, Nijhof Wassink's planning process is influenced by the relationships between the customers, management, and the planning department. Customers can influence the demand for requests based on costs, reliability, and sustainability. The management plays an essential role in strategic decision-making to improve the company's performance, whereas the planning department is crucial for developing and executing operational planning strategies. The stakeholder's interactions impact the operational efficiency, customer satisfaction, and, ultimately, Nijhof Wassink's market position.

2.2 Current Planning Process

Nijhof Wassink's current planning process is complex due to its large scale and its many decisions. During workdays, around 100 vehicles are on the road, and approximately 100 requests, consisting of a pick-up and delivery, are fulfilled. To plan these vehicles, a planning department of 4 planners is at work for 5 days a week. The planners are focused on constructing a routing plan for the upcoming days, as well as real-time problem-solving when unexpected issues arise due to delays or other disruptive events. The planners work with a planning software, which lists all unplanned requests, the current routing plan, and real-time information about the location of the vehicles. Below, all components of constructing a routing plan will be discussed, and the process flow will be explained.

2.2.1 Customer Agreements

Before the requests are planned, it is essential to understand the underlying process of receiving requests. This process is managed by the Nijhof Wassink's sales department. The procurement of customers is conducted through a tender process. Potential customers send out their required number of requests from pick-up location A to delivery location B. Transport companies, among Nijhof Wassink, can then submit their proposals, and the potential customers choose their preferred transport company based on price, emissions, and service levels. The prices are determined by the transport costs from the start to end location, the expected empty kilometers after performing the trip, the product handling costs, and the desired profit rate. If Nijhof Wassink is the chosen transport company, an agreement with the customer is established. This agreement is made for a specified trip only, and therefore, it is common for customers to have several agreements with the transport company. This contains the following information:

- The start (pick-up) and end (delivery) location of a trip
- The product information
- The number of trips
- The time frame in which all trips need to be requested
- The price of a trip

In the past 6 months, Nijhof Wassink had a customer base of around 100 customers. The number of requests that each customer requests varies widely. Namely, 13% of the customers account for 80% of the total number of requests. Furthermore, notice that the customer agreement only states that the maximum number of requests should be requested within some time frame, but the requests do not have to be spread evenly among the time frame. For instance, if the time frame is a year and the total number of trips is 12, the customer can request one request each month but is also allowed to request all 12 trips in the first month. Additionally, the agreed number of trips is a maximum, and there is no consequence if the customer requests fewer trips. This combination causes an unknown request rate in the planning department. Also, there is no clear agreement on the time between placing a request and the scheduled pick-up date. Generally, requests are requested 9 days in advance. However, with a planning horizon of 2 days, around 9% of requests are placed after the routing plan for the scheduled pick-up date has been completed. These requests are, therefore, considered late. Subsection 2.2.6 will elaborate on this further.

2.2.2 Request Information

After the agreement is established, the customer is allowed to request requests at Nijhof Wassink's customer service. These requests contain more details than the customer agreement. Each request contains the following information:

- Pick-up location and delivery location
- Time window of pick-up and time window of delivery

- Product handling instructions
- Customer handling preferences

When a request is sent to the planning department, it should contain time windows in which the vehicle has to arrive at the pick-up and delivery location. The flexibility of the time windows varies by location. Based on experience, the planning department is aware of how each location handles deviations. Furthermore, the product handling instructions can include certain forbidden products to precede this request. These rules are established by law. For example, plastic granulates cannot be transported before a human food product is carried without cleaning the inside of the trailer. Additionally, certain customers have preferences on product handling. For instance, some customers always require a trailer cleaning, even when it is not required by law. These handling instructions should be respected by Nijhof Wassink. Furthermore, it is important to note that each request in DBL is a full truckload. Each trailer can carry at most one product for exactly one customer, as the trailer has one compartment and products cannot be mixed. Therefore, the request quantity is always one. Although not included in the request information, the service time for pick-up and delivery is a relevant characteristic of a request. Customers are not required to specify this, but Nijhof-Wassink estimates the service time based on historical data for similar requests.

A data analysis was conducted, showing that the pick-up and delivery windows are approximately 8 hours long, as can be seen in Figure 2.1. The length of the pick-up time window has a larger spread and has more extreme outliers compared to the delivery time windows. In 45% of the cases, the pick-up time window is set to 1 minute. This means that the deadline for the pick-up is at that time. Arriving earlier at the pick-up location is often also allowed. Delivery time windows have less variation, and only 8% of the time windows have a specified deadline. The time between the end of the pick-up window and the start of the delivery window is usually 1 or 2 days, disregarding the outliers (see Figure 2.2. Therefore, most pick-ups have a next day or day after delivery. Furthermore, the workload on each day is approximately the same on weekdays. Monday and Tuesday tend to be busier with deliveries, and only a few requests are due during the weekend. This distribution of the workload can be found in Figure 2.3.

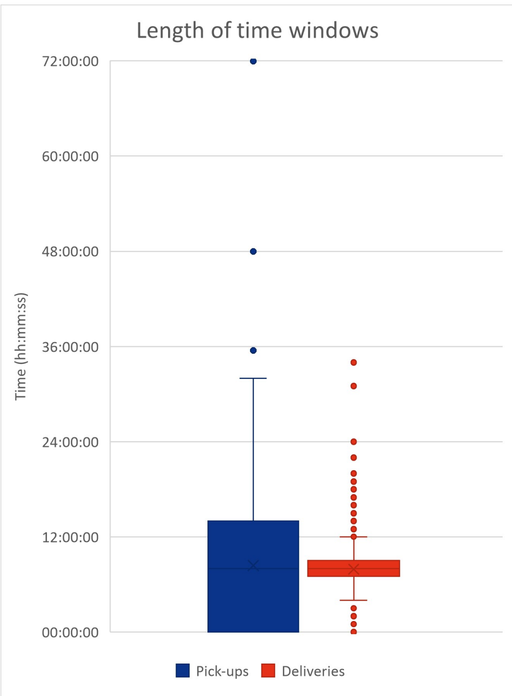


Figure 2.1: Box plot of the length of the pick-up and delivery time windows.

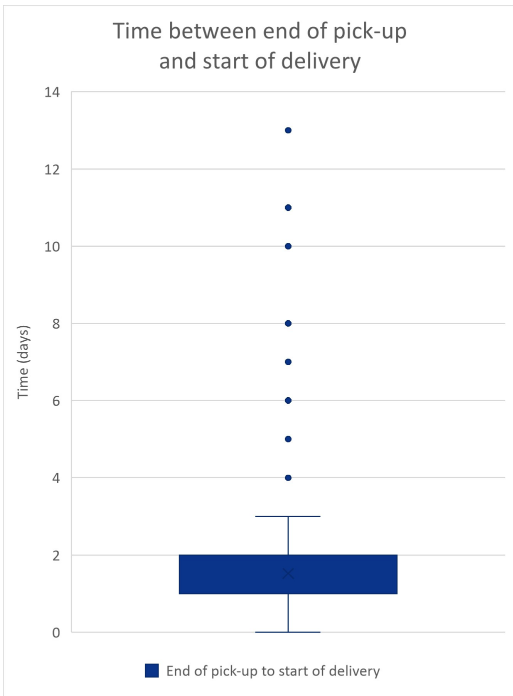


Figure 2.2: Box plot of the time between the end of the pick-up and the start of the delivery window.

Furthermore, the data show that after 67% of the requests, a cleaning is planned. On average, the cleaning takes place 80.43 km from the delivery location. However, from the histogram in Figure 2.4, there are outliers with long distances. These have a large impact on the average. By only excluding 5% of the highest values, the average already decreases to 56.9 km.

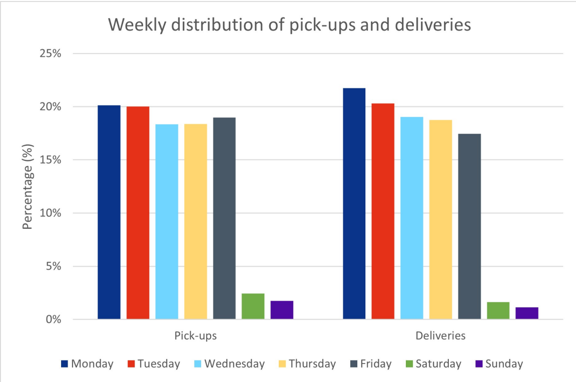


Figure 2.3: Weekly distribution of pick-ups and deliveries.

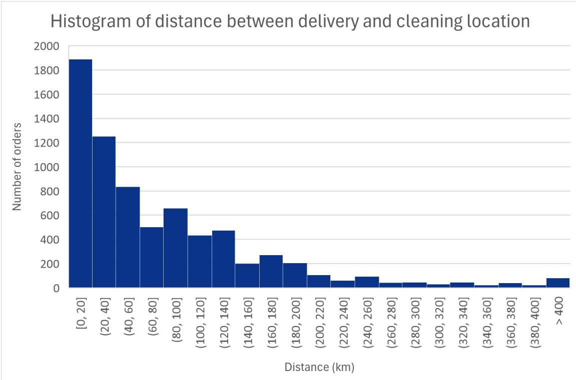


Figure 2.4: Histogram of the distance between the delivery and cleaning.

2.2.3 Vehicle Resources

To plan the requests, the planning department needs to consider its available vehicles. Nijhof Wassink deploys truck and trailer combinations, together referred to as vehicles. The truck is the front part containing the driver's cabin and cannot transport any load. The trailer is coupled to the truck and transports the load. As said before, the trailer always holds a full truckload and serves at most one customer at a time. Nijhof Wassink's DBL has 97 trucks and 136 trailers in total. Generally, all trucks and trailers are in use every week.

Moreover, the vehicles are stored at trailer parks located in Rijssen, Coevorden, Rotterdam, Bleskensgraaf, Antwerpen(BE), Zelzate(BE), Kallo(BE) and Barleben(DE). These trailer parks are also referred to as transfer locations, and are often the start and end locations of a driver's route. It also holds space for spare trailers. This makes it possible for drivers to decouple their trailer, either empty or loaded, and exchange it for another trailer. Transferring trailers can be convenient for driver changes or meeting tight time windows. It is also common for vehicles to end their shifts at the end of the week with a loaded trailer. The trailer is then stored over the weekend without impacting the operational capacity, and the request is delivered the following week. By exception, drivers are allowed to store their vehicles at another location. These locations are solely used as start and end locations, so transfers are not permitted there.

To give an overview of the distribution of the pick-up and delivery locations and transfer locations, Figure 2.5 helps to identify geographical clusters. The color of the circles identifies the type of location, and the size of the circles indicates the number of actions recorded over the past 6 months. It is evident that the pick-up locations are more concentrated, and the delivery locations are more dispersed. This is logical as pick-ups typically occur near harbors or other central hubs where the load can enter a country. Nijhof Wassink's deliveries serve as a means to distribute those loads. Furthermore, the transfer locations, indicated in green, are located near high-traffic areas, such as the harbor of Rotterdam and Antwerp. These transfer locations can easily facilitate the storage of incoming pick-ups until the delivery is scheduled.

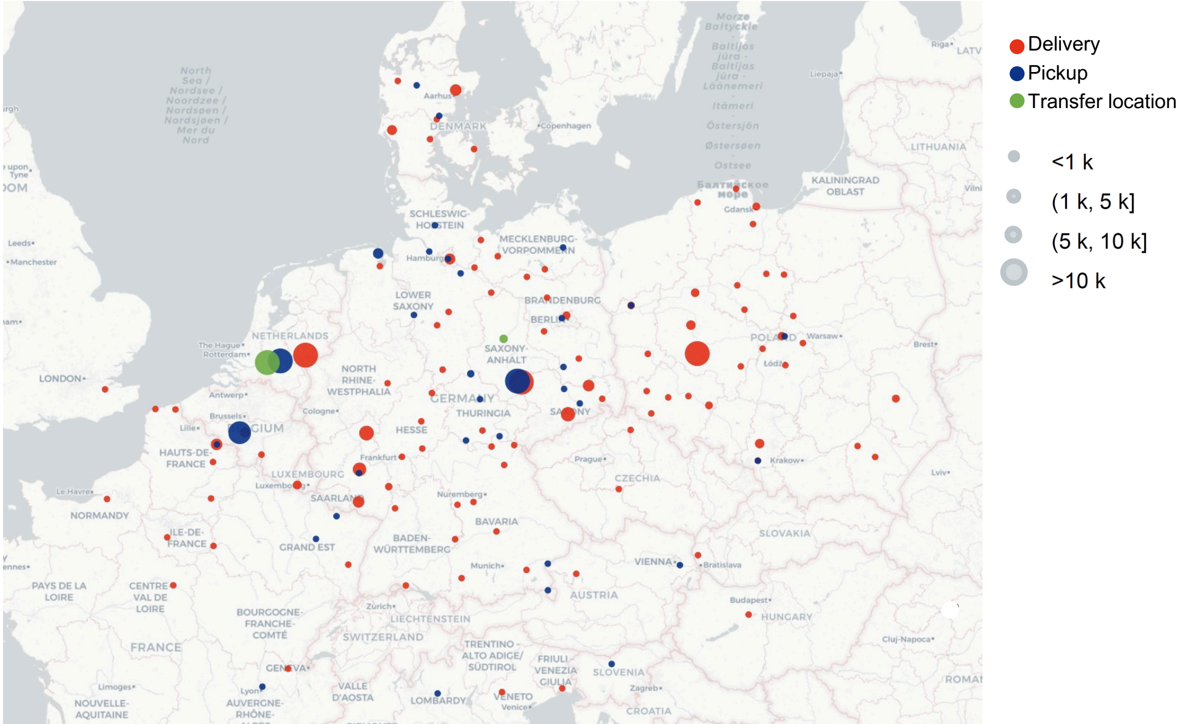


Figure 2.5: Map with locations of pick-ups, deliveries, and transfer locations and their quantities.

2.2.4 Driver Resources

Additionally, the truck drivers are an important resource. In total, Nijhof Wassink has a pool of 160 drivers. At the start of the year, a preliminary schedule outlining the available driver hours. So, the total number of staffed trucks is known in advance. All truck drivers are long-haul drivers, meaning they are on the road for multiple consecutive days and stay overnight at an on-route location. Nijhof-Wassink chooses a standard shift length of 5 days. Each shift starts and ends at one of Nijhof Wassink's transfer locations. The planning department is responsible for complying with the Dutch "rij- en rusttijden bij wegvervoer" law [52]. These regulations define the driver's maximum allowable working hours and minimum required rest periods. The law distinguishes between different types of time allocation: rest time in which no work is performed, driving time spent on actively driving a vehicle, and work time, which includes driving time and waiting time at customers. The law states the following:

- Drivers must take a daily rest of at least 11 consecutive hours.
- Between two full daily rest periods, the rest period may be reduced to 9 hours consecutively without compensating for the missed rest period.
- A break of 45 minutes is required on work days exceeding 9 hours, while a break of 30 minutes is required on work days between 6 and 9 hours.
- The maximum allowed total driving time is 10 hours per day up to 2 times a week. On the other working days of the same week, the maximum is 9 hours per day.
- The weekly maximum driving time is 56 hours.

For Nijhof Wassink, the most restrictive factor in the routing planning is the daily rest requirement. Short breaks are typically taken during waiting times at a pick-up or delivery location, minimizing their impact on the routing plan. Additionally, the law permits drivers to work every day of the week, whereas Nijhof-Wassink chooses a 5-day work week. As a result, the last condition of the law only applies in cases of overtime.

2.2.5 Outsourcing Requests

Given the driver and vehicle resources, Nijhof Wassink can sometimes have insufficient resources to fulfill all incoming requests within the required time windows. Additionally, outsourcing requests can boost planning efficiency, for instance, if the pick-up or delivery location is inconvenient in combination with the current routing plan. This may induce many empty kilometers, reducing the request's profitability or even a financial loss. So even though charters must be paid to execute an outsourced request, fulfilling the request with Nijhof Wassink's vehicles may be more costly. The planning department can choose to outsource requests to other transporters, known as charters. Nijhof Wassink collaborates with multiple trusted charters, and 34% of requests of the last half year were outsourced. The charter rates are already known, so the planners can estimate if the charter costs outweigh the costs of keeping the request. These outsourcing decisions reduce the planning workload as the requests do not have to be planned anymore.

2.2.6 Planning Horizon

The current planning horizon is 1 to 2 days, meaning there is a strong focus on a short-term routing plan. Even though the planning horizon is short, historical data show that requests are received on average 9 days before the scheduled pick-up date. This suggests that there is an opportunity for extending the planning horizon. As said before, 9% of the requests arrive late, meaning that planners are actively working on the routing plan of the specified pick-up date when the late request arrives. Regardless of the short notice, almost all requests are accepted. This is partly due to an act of service but also because of a lack of overview in the overall routing plan. Due to the latter, it is currently unknown if a late-arriving request has a positive or negative effect on the overall routing plan. This means that there is little reason to reject requests. Extending the planning horizon potentially increases the number of late requests, which may require Nijhof Wassink to reconsider its planning strategy to reduce the late requests.

2.2.7 Current Planning Process

Combining the information about the resources and requests, the process flow of constructing a routing plan can be explained. The flow chart is also shown in Appendix B. When a request is received, customer services first checks if the request is in line with the customer agreement. If the request details are incorrect, the customer is redirected to the sales department. Otherwise, the request is added to the list of unplanned requests and is forwarded to the planning department.

The planning department plans each request one by one. If the request is urgent, it has to be planned as soon as possible and is assigned to the vehicle that is best suited. The driver is probably already en route, so the planners have to send updated route information to the driver.

If the request is not urgent, there is more planning flexibility. Firstly, the planner checks if there is a vehicle that is allowed to carry this request's product after its previous load. Moreover, the vehicle must be able to execute the pick-up and delivery within the required time windows. After considering these constraints, the planners aim to minimize the empty kilometers by assigning the request to a vehicle that is sufficiently close to the pick-up location. After that, the planners check if a cleaning is required. The vehicle is assigned to the closest cleaning station, and updated route information is sent to the vehicle. If no vehicle can satisfy the constraints or is not close enough to the location of the request, planners attempt to alter the routing plan of the already planned vehicles or outsource the request. If this is unsuccessful, the planning department can ask customer service to contact the customer and discuss how to proceed. This is a continuous process until all requests are planned.

2.3 Key Performance Indicators

After the construction of the routing plan, the quality of the routing solution is evaluated by different key performance indicators (KPIs), such as the load factor, the total travel distance, and the CO_2 footprint. This section will explain each KPI and assess the planning performance from the last half year. In this period, 9445 requests have been planned and executed by Nijhof Wassink's DBL. An overview of these KPIs is provided in Table 2.2.

2.3.1 Load Factor

Empty kilometers are travel distances without a load. This occurs when the trailer is empty, for instance, between the delivery of one request and the pick-up of another request. It also includes the travel distance from and to a cleaning station. To evaluate the number of empty kilometers, we use a percentage of empty kilometers relative to the total travel distance. The load factor (LF) is related to the empty kilometers, as it is the percentage of travel distance with a load. Therefore, the load factor complements the percentage of empty kilometers and can be calculated as 1 minus this percentage (see Equation 2.1). Note that this formula assumes full truckload operations, where the truck is either empty or full. In less-than-truckload settings, the load factor is typically measured relative to the trailer's weight utilization over distance. To increase Nijhof Wassink's efficiency, a high load factor is favorable.

$$\mathrm{LF} = 1 - \frac{\mathrm{EmptyKilometers}}{\mathrm{TotalDistance}} \tag{2.1}$$

In the historical data from the last half year, the average empty distance per request was 143.5 km. Relative to the total distance per request, the percentage of empty kilometers is 27.20%, and the load factor is 72.80%. The distribution of empty kilometers as a function of total request travel distance is shown in Table 2.1. Requests with a low and high total distance tend to have more empty kilometers. For short trips, the percentage of empty kilometers is easily influenced when the total travel distance is small. The longest trips have a rather high percentage of empty kilometers due to the request's location. Some requests lie far away from the areas with a high demand, which makes it harder to find a return load in this vicinity. This can lead to long travel distances without a load.

Travel distance interval (km)	% Empty kilometers	#Requests in interval
[0-200)	47.5%	662
[200-400)	26.8%	1272
[400-600)	23.9%	2751
[600-800)	26.8%	2185
[800-1000)	24.3%	1710
[1000-1200)	28.4%	606
[1200-1400)	36.4%	157
>1400	41.4%	102

Table 2.1: Percentage of empty kilometers for requests grouped by total travel distance and the number of requests per total travel distance interval.

2.3.2 Total Travel Distance

The total travel distance is another KPI. This is the total number of kilometers traveled by all trailers. Note that Nijhof Wassink tracks the trailers instead of the vehicles or trucks. This is because their customers are charged based on the trailer's route. Each request's route is determined from the moment that the truck is coupled to the designated trailer used for the request until the trailer is decoupled. For high efficiency and low operational costs, a lower total travel distance is desired. Also, a low travel distance can indicate that Nijhof Wassink has room for taking on more business opportunities. The data analysis shows that the total travel distance per request is, on average, 527.8 km.

2.3.3 Total Emissions

Lastly, as said in Chapter 1, Nijhof Wassink aims to reduce its emissions, in particular its CO_2 footprint. The planning performance can be assessed based on the CO_2 emissions, which reflects the environmental impact of this planning solution. The calculation is done based on the GLEC Framework. This is a well-known framework in the logistics sector used to measure emissions [19]. The CO_2 footprint is expressed as the average Wheel To Wheel CO_2 efficiency in grams per kilometer. Wheel to Wheel refers to the entire fuel lifecycle, so from the production to the consumption. The formula is shown below in Equation 2.2. The lower the CO_2 emissions, the more efficient and environmentally friendly the routing plan is.

$$\mathrm{CO_2}\text{ footprint [kg/km]} = \frac{\text{Emission factor [kg/L]} * \text{Fuel usuage[L]}}{\text{Travel distance [km]}} \tag{2.2}$$

Based on the historical data, the average CO_2 footprint per request is 0.93 kg/km. The emission factor of Nijhof Wassink's vehicles is 3.256 kg/L, based on B7 biodiesel. Moreover, the average amount of fuel used on an average length route is 150.55 L.

KPI	Value
Average travel distance per request	527.8 km
Average empty kilometers per request	143.5 km
Average % of empty kilometers per request	27.2%
% Load factor per request	72.8%
Average CO_2 footprint per request	0.93 kg/km
Number of requests	9446

Table 2.2: Summary of the KPIs.

2.4 Conclusion

This chapter answered the research question: *"How is the current planning process organized?"*. Firstly, the stakeholders are introduced, and their requirements and wishes are determined. Then, the sub-processes of the current planning process are introduced and analyzed. It discusses all aspects concerning the customers, requests, drivers, and vehicles. This section concludes by discussing the connections between the sub-processes. Lastly, the key performance indicators are explained, and the performance of the routing plan from the past half year is assessed. The relevant KPIs are summarized in Table 2.2. This understanding of the context is essential to understanding what optimization model is the best fit for this company's context. Chapter 3 therefore explores methods to model a routing solution that minimizes empty kilometers.

3| Literature Review

This chapter explores the available theories and methods in the literature to answer the research questions mentioned in Section 1.3. The purpose of this literature review is to acquire the necessary knowledge to develop an optimization model that can solve Nijhof Wassink's routing problem. Based on the context understanding from Chapter 2, Section 3.1 classifies this company's transport challenge within recognized logistics and transportation problem classes. Section 3.2 discusses different variants of vehicle routing solutions and discusses the best methods that can be applied and adapted to this company's context. Section 3.3 discusses relevant solution methods that can be combined with discussed optimization models.

3.1 Classifying the Problem Context

The transportation challenge of minimizing empty kilometers can be classified within different recognized logistics and transportation problem classes. This section will analyze which concept is the best fit for this company's context.

The structure and aim of Nijhof Wassink's DBL planning and its transportation challenges strongly align with a Vehicle Routing Problem (VRP). This is a combinatorial optimization problem that is NP-hard. The concept of a VRP is defined as "the problem of designing least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints." [34]. The resemblance is the design of a routing solution. Moreover, cost is, in this context, defined as minimal empty kilometers. The side constraints refer to this problem's request and resource constraints. Additionally, the VRP has been widely studied and has evolved to include real-life complexities, such as time windows, vehicle limitations, and multi-depots [5]. The adaptability of the VRP can be used to customize to Nijhof Wassink's problem context.

Opposed to the VRP, the transport problem at hand has minor resemblances with the Generalized Assignment Problem (GAP). This problem "examines the minimum cost assignment of n jobs to m agents, such that each job is assigned to exactly one agent and subject to capacity restrictions on the agents" [7]. Particularly, each request is considered as a job that needs to be assigned to a vehicle. The GAP can also incorporate real-life complexities but is usually applied to small problems and focuses mainly on assigning agents to jobs. However, this case deals with homogeneous vehicles and focuses on route optimization [4]. Furthermore, Network Flow Problems (NFP) are used to optimize the flow of goods between a set of nodes [56]. They are commonly used in transport networks and focus on aggregate product flows between those nodes. They often do not explicitly model individual vehicle routing or customer-specific constraints. For this problem, however, we aim to route each vehicle instead of the total product flows, and customers pose limitations on the solution. Moreover, scheduling problems are also common in combinatorial optimization. They aim to find a schedule that determines the sequence of jobs at different agents [63]. This solution heavily depends on the precedences between jobs. In this research's problem context, request pick-ups should precede their deliveries but there are no dependencies between the requests. This problem class is, therefore, not suitable for Nijhof Wassink's problem.

This section compares commonly used problem classes in transport. It concludes that the VRP is the best fit for Nijhof Wassink's problem context due to its ability to optimize routes between geographical locations and its flexibility concerning different constraints. To introduce and identify the most appropriate variant of the VRP for this context, Section 3.2 provides a comparison between commonly used variants.

3.2 Variants of the Vehicle Routing Problem

The classical VRP, first introduced by Dantzig and Ramser, is concerned with designing optimal delivery routes, where each vehicle travels at most one route [15]. Also, each customer has a demand and vehicles have some capacity that cannot be exceeded. It originated as an extension of the Traveling Salesman Problem (TSP). This problem is concerned with visiting all nodes in one route while visiting each node exactly once and minimizing the cost. The VRP can utilize more than one vehicle to visit all nodes. Conversely, if we assume that the VRP has one vehicle with infinite capacity, the VRP reduces to the TSP [51]. The TSP and VRP are both combinatorial optimization problems that are both NP-hard. Over time, numerous variants have been derived from the VRP to incorporate real-life complexities into the VRP. For example, models can deal with multiple depot locations, time windows for customers, and customer pick-ups and deliveries. Since the VRP is an NP-hard problem, all variants of the VRP are also NP-hard [33]. This section provides an overview of key variants of the VRP. These variants are also displayed in Figure 3.1.

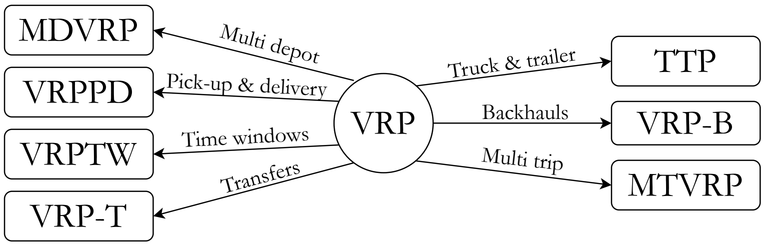


Figure 3.1: Variants of the VRP explained in Section 3.2.

3.2.1 VRP with Time Windows

A commonly occurring variant of the VRP is the routing problem with time windows (VRPTW). This variant adds a time constraint for each customer. Customers must be served between the start and end of their time window. The vehicle can arrive before the start of the time window but the customer will not be served before [20]. In practice, time windows are a useful addition as on-time deliveries have been linked to the accuracy of the transport and customer satisfaction. Typically, the time windows can be soft or hard constraints. Violating a hard constraint results in infeasible solutions, whereas the violation of a soft constraint is a feasible solution but induces a cost penalty in the objective function [64]. Hard constraints have a large impact on the solution space of the problem. Instead, softening the constraints can lead to more feasible solutions and is more likely to provide an executable vehicle routing solution [59]. Soft time windows can be modeled in various ways depending on the context [49]. Often, the penalty costs increase linearly with the degree of violation. Alternatively, penalties for late arrivals are significantly higher than for early arrivals. The latter occurs in situations where vehicles are allowed to wait without added costs. In other words, only the delay must be minimized. Another option allows a small amount of earliness and latency without significant added costs but at some specified deviation, the costs become extremely high. The extreme costs mimic the modeling of hard constraints. This is seen in inventory control, where slight deviations generally do not risk a stock-out but after a substantial deviation from the time window, it either increases the inventory holding costs excessively or risks stock-outs [9].

3.2.2 VRP with Pick-ups and Deliveries

Another well-researched variant is the VRP with pick-ups and deliveries (VRPPD). This case involves customer requests consisting of a pick-up and a delivery instead of deliveries to each customer from the depot in the general VRP. The routing solution should pair the pick-up and corresponding delivery. Practical examples of this are parcel delivery services that pick up packages from a sender to a delivery address. This is a one-to-one process, as each pick-up location corresponds to exactly one delivery location. It is also common in recycling and waste collection, which is a one-to-many process [3]. A single delivery location serves multiple delivery locations. Parragh et al. (2008) note that the VRPPD is often combined with time windows, commonly referred to as the VRPPDTW [45].

3.2.3 VRP with Multiple Depots

This variant of the VRP includes multiple depot locations for vehicles to start and end from and it is referred to as the Multi-Depot VRP (MDVRP) [42]. This is a more realistic situation in which the distribution of load is done from several depots to its customers. This can be particularly useful if the customers have a wide geographical spread. Practically, it also appears when raw materials are harvested from different locations and should be distributed amongst different plants. To deal with a multi-depot VRP, customers should be divided amongst the depots. Thus, this problem first deals with an assignment problem before the routing problem [25]. One way is to assign each customer to the closest depot. This problem can then be solved as a set of parallel general VRPs but in practice, this assignment is not always as clear [32]. This particularly is not an effective solution in collaborating companies that bundle their vehicles and customers to optimize their route solution [42, 37]. Therefore, generally, the multi-depot approach can serve each customer with an available vehicle from any depot. The vehicles should, however, start and end at the same depot. An alternative version allows the vehicles to start and end at a different depot. This allows vehicles to replenish at the closest depot, minimizing total travel distance [13].

3.2.4 Truck and Trailer Problem

The truck and trailer problem (TTP) is a less straightforward derivation from the VRP. In a TTP, a fleet of trucks and trailers serve a set of customers [32, 16]. A truck and trailer combination, i.e., a truck pulling a trailer, is similar to a vehicle as introduced before. Trucks, as well as trailers, can hold loads but only the truck, with or without a trailer, can drive. Unlike before, trucks can park their trailers to, for example, increase truck maneuverability. This divides the customers into two sets: only accessible by truck or accessible by complete vehicle (but also without trailer). The routing solution provides vehicle routes with two levels. The main level includes complete vehicle routes, whereas the second level includes only truck routes. These routes differentiate from the VRP as the TTP solution allows to visit a node more than once, resulting in a routing solution with subtours. This enables trucks to re-couple with their trailer [58]. Real-world applications have appeared in milk transport with customers in mountainous areas or waste collection in crowded cities [23, 26]. A set-partitioning formulation of the TTP can be seen in Villegas et al. (2013) [61].

3.2.5 Multi-trip VRP

A multi-trip VRP (MTVRP) allows vehicles to make multiple trips in a working period [64]. As opposed to the general VRP, this variant allows subtours in the routing solution. The additional challenge of this VRP is determining the set of trips and assigning each trip to a vehicle. In practice, this variant is useful if the total demand exceeds the total vehicle capacity [35]. Its advantage is that smaller capacity vehicles can accommodate the demands of a large area by utilizing intermediate stops at the depot. Practically, MTVRP is used if the depot is replenished continuously throughout the working period [6] and vehicles may need to return to the depot for an intermediate replenishment. It is also often suitable for city contexts, where distances between the depot and customers are short. Cattaruzza et al. (2016) proposed several mathematical formulations of the MTVRP [6].

3.2.6 VRP with Backhauls

The VRP with backhauls (VRP-B) is a routing problem with pick-ups and deliveries [30]. Customers are divided into linehaul customers, who require deliveries from the depot, and backhaul customers, who return goods to the depot. A key constraint is that all linehaul customers must be served before any backhauls are scheduled, and the vehicle's capacity cannot be exceeded. Furthermore, the linehaul and backhaul customers are not paired, and transported volumes are less than a truckload. This type of VRP is commonly used in contexts with two steams of goods, such as a recycling process or in the retail industry where suppliers are the linehaul and stores are the backhaul. The variant can be extended to include time windows, utilization of a mixed fleet, or spatial vehicle loading constraints. Objectives of a VRP-B often minimize total travel distance or cost [31]. Mingozzi (1999) presented one of the first mathematical formulations of this VRP variant [40].

3.2.7 VRP with Transfers

The VRP with transfers (VRP-T) allows the vehicles to transfer their load between vehicles at intermediate locations [21]. This is similar to the multi-trip VRP, except that the transfer locations are at a different location than the depot. This is convenient when a vehicle cannot complete an entire trip. Pemberthy et al. (2019) have studied a case where vehicles cannot cross country borders, and the load needs to be transferred [46], or Rais et al. (2014) use transfers to adjust the load or switch drivers [50]. Moreover, it is applied in a distribution chain, where the load is transported from a large central depot to local hubs and finally to the customer. The local hubs are, in this case, transfer locations. The use of transfers is often used in combination with pick-ups and deliveries [50, 38]. This allows vehicles to store the pick-ups at some transfer location. Then, other vehicles can finish the delivery, or the same vehicle can continue this request later. This can help to better utilize the vehicle availability and capacity and increase the solution space [3]. A formal definition of the VRPT can be found in the work of Rais et al. (2014) [50].

3.2.8 Overview of Variants

The variants of the VRP introduced above are all researched to some extent. Konstantakopoulos et al. (2022) researched that the general VRP with vehicle capacities has appeared in 82.9% of the papers published between 2012 and 2022 [32]. Second and third are the VRP with time windows and pick-ups and deliveries with respectively 46.4% and 20.2%. They also mention that these variants are rarely studied individually. Combining the different VRP variants to resemble real-life scenarios is more common. There is an extensive amount of literature found on VRPs with pick-ups, deliveries and, time windows, and is referred to as PDPTW [64]. For example, Kammarti et al. (2007) proposed a solution method, and Khoo and Bonab (2022) addressed this problem with multiple objectives [28, 29]. Moreover, the MTVRP, VRP-B, and TTP appear less than 5% and were combined with time windows in the following papers: [27, 35, 36, 17, 31]. Also, the TTP has been combined with transfers in [46]. Furthermore, variants with transfers are often paired with pick-ups and deliveries. The advancements of VRPPD with transfers have been summarized in [3]. A combination of transfers, pick-ups, deliveries, and time windows has also been researched but the number of papers is limited. Cortés et al. (2010) propose a formulation in a passenger transport system [11]. Their formulation also allows the use of several depots but does not include restrictions on driver hours or customer-specific constraints. Lye and Yu (2023) followed with a critical review of this problem and proposed a revised formulation, which also did not include these restrictions [38]. Additionally, Masson et al. (2013) cover a MDPDPTW-T [39]. Unlike this case, all delivery locations can be used as transfer locations, the number of vehicles is unlimited, and the total planning horizon is 10 hours. An applied case in crowd-shipping was also modeled as MDPDPTW-T [57]. However, this study has a planning horizon of at most 5 hours, allows vehicles to visit a transfer station only once, and includes no customer-specific constraints. A complete table with the relevant papers and their discussed VRP variants can be found in Appendix C.

3.2.9 Selecting a VRP Variant

Nijhof Wassink's problem context shares characteristics with several of the variants that were previously introduced. The variants that do not align with this problem's context are the MTVRP, TTP, and VRP-B. In a multi-trip context, vehicles must restock or unload at the depot as an intermediate stop to continue their routes. Contrarily, in Nijhof Wassink's context, depots merely park the vehicles, which is counter-productive when serving customers. The TTP is also not the best fit for this context, as the TTP utilizes trucks that can operate without trailers. For Nijhof Wassink, the trucks are only a means of moving the trailers and the trucks on their own cannot transport any load. The backhaul variant incorporates both pick-ups and deliveries but does not support a pairing of the pick-up to the delivery. It commonly involves less-than-a-truckload operations and has a focus on vehicle capacity management and loading constraints. Instead, Nijhof Wassink requires a paired pick-up and delivery, and truckloads are always full, reducing the relevance of vehicle capacity planning. On the other hand, a combination of the PDPTW with multiple depots and transfers seems to have the strongest resemblance. This combination will, for the remainder of this research, be referred to as MD-PDPTW-T. As mentioned, some papers have explored this combination of VRP variants before but there are still key differences to this paper's problem context. To the best of my knowledge, these papers have not considered an objective based on empty kilometers and only included limited driver and customer constraints.

3.3 Solution Methods for VRPs

Over time, many researchers have proposed different ways to solve variants of the VRP. These solution approaches can be divided into two categories, namely exact and heuristic approaches. Exact methods often involve mixed-integer programs (MIPs). However, the complexity of VRPs makes finding an optimal solution in a reasonable time almost impossible, particularly in larger instances. Therefore, heuristics are typically favored over exact methods due to their balance between solution quality and computation time. The following sections elaborate on the most relevant approaches for the MD-PDPTW-T. Additionally, a summary of the solution methods for each of the VRP variants can be found in Appendix C.

3.3.1 Exact Methods

Exact VRP solutions are often based on MIP models and guarantee an optimal solution if one exists. The most used methods for the PDPTW are branch-and-cut, branch-and-price, and branch-and-cut-and-price methods. The work of Ropke and Cordeau (2009) used the latter approach, and their results have been established as a benchmark for future research [53]. Baldacci et al. (2010) formulate VRPs as a set partition problem and provide an exact branch and price algorithm. The proposed algorithms were able to solve most of the benchmark instances on the PDPTW and MDVRP relatively fast [1, 18]. Thereafter, Pessoa et al. (2019) proposed a branch-and-cut-and-price method for a generic VRP model [47]. It achieved superior results on the VRPTW, notably better performance on the MDVRP, and mixed outcomes on the PDPTW, compared to existing literature. Recently, a revised MIP formulation for the PDPTW-T with a single depot was proposed by Lyu and Yu (2023) [38]. Their solvable scale has increased from 3 requests and 4 transfer stations to 5 requests and 4 transfer stations while reducing the average computing time by 40%. From the literature, it seems that solving a realistically sized problem exactly at this moment is not possible in a feasible time.

3.3.2 Heuristic Methods

The heuristics used in VRPs can be divided into constructive heuristics and meta-heuristics. The analysis by Konstantakopoulos (2022) concluded that construction algorithms are mainly used to build an initial solution [32]. The initial solution can then be used as input for meta-heuristics. Meta-heuristics are widely used for solving VRPs due to their ability to handle complexity and large-scale problems. These methods are capable of both exploring the often large solution space as well as exploiting a promising region of the solution space. Some of the most used meta-heuristics in VRPs will be discussed below.

Simulated Annealing
Simulated annealing is one of the first and most well-known meta-heuristics [55]. It requires an initial solution, which is usually chosen at random. Then, by making a small change to the current solution, so-called neighbor solutions are drawn up. If the solution is better, it is always accepted as the new, best current solution but if the solution is worse, it can still be accepted with a certain probability. The acceptance probability of worse solutions decreases as the temperature parameter

gradually decreases [20]. At high temperatures, the algorithm is more likely to accept worse solutions, enabling exploration of the solution space and chances of escaping local optima. As the temperature decreases, the acceptance of worse solutions is less frequent, allowing the algorithm to focus on thoroughly exploiting the solution space around the current solution. According to Zhang et al. (2022), SA is computationally efficient but works best in small-scale instances [64].

One of the first applications of SA on a VRP with time windows was done by Chiang and Russel (1996) [8]. They proposed a SA method with 2 methods of finding new neighbors and tested this on instances of 100 customers. In 2006, Bent and Van HentenRyck proposed a two-stage SA approach to minimize the number of vehicles, combined with a linear search to minimize distance[2]. Their approach was applied to VRPTWs. However, the authors noted that an extension with pick-ups and deliveries remains an issue. Furthermore, a parallel SA method is proposed to solve a VRP with simultaneous pick-ups and deliveries and time windows [62]. Future work suggests that different local search techniques can be used to further decrease travel distance. Cortes and Suzuki (2020) applied a two-stage SA to a VRP with transfers at customer locations, compared to a VRP without transfers [12]. It also allows customer demand to be satisfied by more than one vehicle. The study suggests that transfers decrease overall costs in instances of 400 customers.

Tabu Search

A tabu search (TS) is a local search method that aims to escape local optima by also allowing a worse solution than the best found so far [24, 22]. It also constrains the search in the solution space by prohibiting certain neighborhood moves. These moves are on the so-called 'tabu list' for a predefined time. Zhang et al. (2022) found that TS methods are closest to the optimal solution but have the longest computation time [64].

One of the first greedy solutions for the VRP with pick-ups, deliveries, and time windows, was the reactive tabu search developed by Nanry and Barnes (2000) [43]. The initial solution was created by an insertion algorithm, and three different neighbor moves were considered. For instances with 50 customers, near-optimal results were obtained. Soon after, Cordeau et al. introduced a unified TS method applicable to multiple variants, including the MDVRPTW [10]. Crevier et al. (2007) studied a case of MDVRP with inter-depot routes, allowing vehicles to replenish at certain depots. They found routing solutions with a tabu search, combined with adaptive memory and linear programming [13]. Note that the inter-depot replenishments are similar to transfers. Moreover, Montané and Galvao proposed a tabu search for VRPTW with simultaneous pick-ups and deliveries. Three types of movements and 4 different neighborhood moves are used and presented quality results for instances up to 400 customers [41]. Although various VRP variants have been solved successfully with a TS, the specific variant of interest addressed in this paper has not yet been explored.

Adaptive Large Neighborhood Search

The adaptive large neighborhood search (ALNS) is a local search framework that uses a set of heuristics. Destruction heuristics are in place to destroy a part of the solution, upon which repair heuristics rebuild a new solution. ALNS can track the performance of the destroy and repair heuristics and adjust the probabilities of selecting those heuristics. The choice of a set of operators enables the algorithm to explore large parts of the solution space in a structured way [48]. Furthermore, the heuristic requires a local search framework, which can be based on simulated annealing. This allows the algorithm to occasionally accept worse solutions to avoid getting trapped in local optima. This makes the ALNS more robust. The downside to the algorithm is the relatively large number of variables that are in need of parameter tuning, including the scores for each operator.

Pisinger and Ropke (2007) have studied the MDVRP and PDPTW with an ALNS [48] and see potential to extend their algorithm to other combinations of VRP variants. They also mention that, compared to other heuristics, the ALNS performs better in tightly constrained problems. This is due to its ability to make large changes to the solution to reach new feasible solutions. Masson et al.

(2013) then extended this research and added transfers to the PDPTW variant [39]. They presented an ALNS with 3 destruction and 3 repair heuristics, as well as heuristics for selecting an appropriate transfer location. The proposed algorithm is tested on PDPTW instances compared to PDPTW-T instances. The instances used have up to 4 transfer locations and can handle at most 200 requests. The computational time of instances, including transfers, is much longer but improves the solution by 9%. A similar study was done by Sampaio et al. (2020) [57]. They aimed to show the benefits of utilizing transfers in crowd shipping. To do so, their ALNS employs 5 destroy operators and 3 repair operators and showed significant improvements in comparison to the benchmark instances.

Genetic Algorithm

The genetic algorithm (GA) is a population-based meta-heuristic. Inspired by biological natural selection, this algorithm generates new 'offspring' solutions based on the fitness of the current 'parent' solutions. The higher the fitness, the higher the probability of selecting this parent solution. Pairs of parent solutions are crossed over to generate offspring. Additionally, random mutations are introduced to some offspring to maintain diversity and explore the solution space. As each generation passes down its strengths to the offspring, the population is refined over time, improving the solution quality.

A hybrid genetic algorithm was proposed by Vidal et al. (2013) [60]. This algorithm is applicable for large-scale instances of the VRPTW, combined with multiple depots and route duration constraints. The instances range from 50 - 1000 customers and up to 9 depot locations. The solution quality and computational efficiency are higher than the current methods. Moreover, Danloup et al. (2018) were the first ones to solve the PDPTW-T with a GA [14]. They performed a comparative study on the performance of an ALNS and GA on benchmark instances. The study shows that the GA slightly outperforms the ALNS. The GA achieves an optimal solution in 74% of the cases, compared to 65% for the ALNS. Furthermore, the proposed algorithm aims to find a set of routes, each starting and ending at one of the depots, and satisfying all requests while minimizing the number of vehicles used and the total traveled distance.

3.3.3 Selecting a Solution Method

As mentioned, the VRP and any of its derivatives are NP-hard problems. This means that the optimal solution cannot be guaranteed in polynomial time, particularly for larger instances. While exact approaches provide precision, they are computationally demanding, especially for large instances. Heuristics, although less precise, provide more practical solutions within a reasonable time. Given the large scale of the problem addressed in this paper, a heuristic approach is the most suitable choice.

Among the various heuristic approaches introduced above, the ALNS appears to be the most suitable for this study. This approach is well-established in the current literature and has performed well on similar VRP variants and problem instances of comparable size. Although, to the best of my knowledge, no studies address the exact VRP variant and specific problem context considered in this paper. However, the design of the destroy and repair operators offers adaptability to the specific constraints and requirements of this problem.

3.4 Conclusion

This chapter provides a comprehensive literature review of the relevant concepts related to this research. Firstly, the problem class is defined, and it is concluded that the company's problem resembles a VRP. Secondly, various variants of the VRP are introduced and compared. This study is most aligned with the MD-PDPTW-T. Thirdly, exact and heuristic solution approaches are presented. It suggests that the ALNS is a promising method to solve the addressed problem. The next chapter, Chapter 4, provides a problem description and a mathematical formulation of the VRP adapted to this research's context.

4| Problem Description & Mathematical Formulation

This chapter aims to design a mathematical model for DBL's planning for two key purposes. First, it provides a mathematical understanding of the optimization problem, including an illustrative example. Second, it justifies the need for a heuristic solution, which is designed in Chapter 5. Section 4.1 outlines the optimization problem. Section 4.2 elaborates on the modeling assumptions. Section 4.3 displays a complete mathematical model and evaluates its feasibility. It ends with section 4.4 concluding the chapter.

4.1 Problem Description

This section outlines the model, including its objective and constraints. As mentioned in Section 3.2, the problem closely aligns with an MD-PDPTW-T problem. The formulation builds on the multi-commodity flow network of Rais et al. (2014) [50] and the improvements of Lyu and Yu (2023) [38]. This network has two flows: one considering the flow of requests carried by trailers from pick-up to delivery location and one considering the flow of trucks from leaving to returning to the depot. The primary objective is to minimize the empty kilometers, thus minimizing the distance trucks drive with an empty trailer. However, to reflect Nijhof Wassink's other operational considerations, the model also incorporates secondary objectives. In particular, it also minimizes the total traveled distance, the penalty incurred for violating the time window constraints, and the number of vehicles used. These additional components are incorporated to ensure more effective and practically feasible solutions. The inclusion of the minimization of total distance is required to avoid unrealistic solutions. To illustrate, if only empty kilometers were minimized, combined with the ability to transfer requests, a truck could continuously couple with loaded trailers to avoid empty kilometers. While this would benefit the empty kilometer KPI, such routing strategies can lead to unnecessarily high amounts of total travel distance and infeasible routing solutions in practice. Furthermore, the time window component is included to represent service level expectations from customers. Additionally, the number of vehicles used is considered. If the model should use all vehicles, it might spread requests across all vehicles, each with minimal empty kilometers. However, this can lead to an excessive number of vehicles used, which can be more costly than a slightly higher proportion of empty kilometers. Therefore, allowing the model to choose the number of vehicles helps to balance empty kilometers and vehicle usage. The model further accommodates transfers, full-truck loads, driver hour limitations, and soft time windows for requests. The model's output is a routing solution that enables vehicles to fulfill all pick-up and delivery requests while allowing them to transfer their loads at transfer stations.

4.1.1 Illustrative Example

To illustrate the optimization problem, an example routing solution is shown in Figure 4.1, with 2 vehicles, a depot, a transfer station, and 3 requests. Each request $i \in \{1, 2, 3\}$ consists of a pick-up P_i and its corresponding delivery D_i. All distances are Euclidean, and time units are defined as equal distance units. Each pick-up and delivery require 2 and 1 unit(s) of service time, respectively. Loading or unloading at transfers requires no service time. However, synchronization is necessary, meaning that the pick-up must first arrive at the transfer station before it can be transported to the delivery location. That is, the vehicle assigned to the delivery cannot depart from the transfer before the corresponding has been dropped off. The input data, including a distance matrix, are presented in Appendix D. Vehicles must be returned to the depot at time $t = 25$, which is a hard constraint. Additionally, vehicles require a 5-unit break after 10 units of accumulated work. All locations have time windows with penalties for early or late arrivals, as presented in a table on the right side of the figure. If a vehicle arrives before the time window opens, the vehicle must wait. The objective minimizes the sum of empty distance, total distance, time window violations, and the number of vehicles used. All objective components are weighted equally in this example.

Figure 4.1 presents the optimal routing solution. The colored arcs represent the vehicles' traveled paths. The numbers on the arc show the distance and time units per arc, and bold represents empty distance. The $t = \ldots$ indicates the vehicle's arrival time at that location. Vehicle 1 departs from the depot at $t = 0$ and reaches P_3 at $t = 2.2$. After which, it serves P_3 for 2 units of time and travels 2 units of time to arrive at D_3 at $t = 6.2$. It then travels to the transfer station to receive the load of request 1, which was dropped off earlier by vehicle 2. To deliver request 1 at D_1, the vehicle would arrive at $t = 11.4$ but this exceeds the working limit. Therefore, a 5-unit break is required and postpones the arrival time to $t = 16.4$ instead. It returns to the depot before it closes. Vehicle 2 travels to location P_1 and drops this load off at the transfer station. It also serves request 2, with time window violations of 0.2 for both pick-up and delivery. For instance, vehicle 2 arrives at P_2 at 15.2, while the location closes at 15, incurring a 0.2 units penalty. To calculate the objective of this solution, we sum the components. First, empty kilometers are indicated in bold on arcs without a load, in total 17.1 (see Figure 4.1). The total distance is the sum of all traversed arcs, totaling 27.1. Time window penalties are incurred if the arrival time is outside the bounds of the time window, which occurs at P_2 and D_2. Two vehicles are used for this solution. As all components are weighted equally, the total objective value is 46.6, as also shown in the table in the figure. This solution satisfies the constraints and achieves the minimum objective for this example problem. Alternative routing solutions are either infeasible or yield a higher objective value.

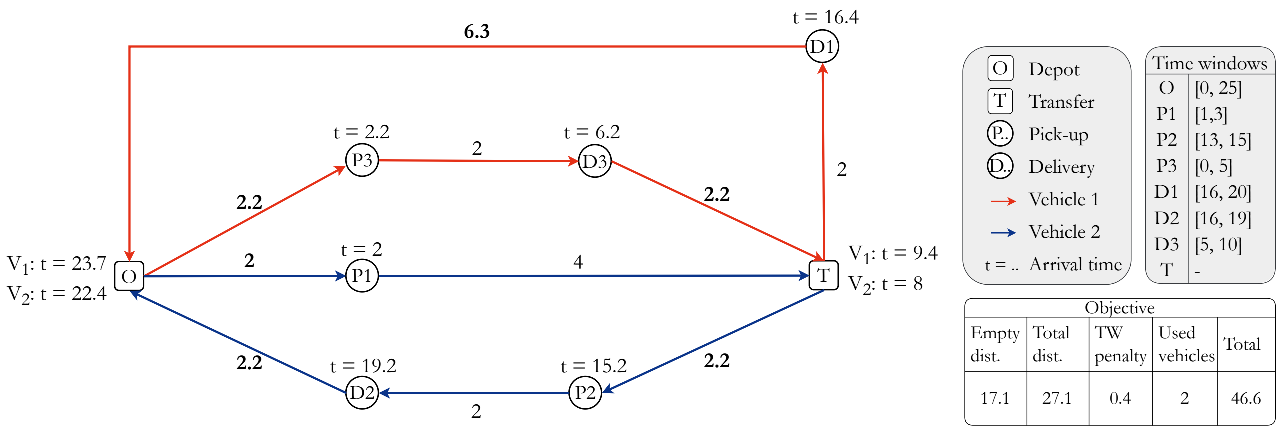


Figure 4.1: Optimal routing solution of illustrative example, including time windows and objective value.

4.2 Modeling Assumptions

After outlining the optimization problem, the modeling assumptions are defined. They simplify the complexities of real-world planning while maintaining the model's practical applicability. The assumptions are listed below:

Resources

- All available trucks during a certain period match the resources from historical planning data from the same period.
- Trucks can always be staffed with a driver, so the number of trucks determines the resources.
- Each truck is staffed by one driver. If the driver is resting, the truck is idle.
- All Nijhof Wassink's trailers can be utilized, even if they were not used in historical data.
- Trailers are homogeneous with identical capacity and travel times for all vehicles.
- Each truck starts and ends at the pre-assigned depot.
- All trailers are empty at the start of the planning horizon.
- Each trailer can hold at most one request. The truck is unable to carry any load.

Requests

- All requests must be served. Requests that were outsourced are already removed from the input data.
- Service times at pick-up and delivery locations are deterministic and predefined.
- Each request is a full-truck load and cannot be split.

Travel Distance and Times

- The distance matrix is assumed to be symmetric.

Cleaning Stations

- Due to the abundance of cleaning stations with comparable service times and costs, a predefined cleaning station is chosen for each delivery. The travel time from the delivery to the cleaning station and cleaning service time are added to the service time of the delivery. The detour distance to and from the cleaning station is included in the travel distance to the next location.

Transfer Stations

- Transfers occur only at transfer locations.
- Initially, there are no requests at the transfer stations.
- Transfer stations have no capacity limits.
- Vehicles can wait for a request at a transfer station.
- Transfer stations are always open, meaning they have no time windows.
- Transfer stations store empty, clean trailers ready for use.
- Vehicles can visit a transfer station at most once.

Time Windows

- Vehicles are expected to arrive within the specified time windows. Deviating from this time window is permitted but incurs a penalty proportional to the violation and the strictness per location.
- Violations are determined by the time of arrival. As the time of departure is primarily influenced by the customer, it is not considered when evaluating these violations.

Driver Hour Limitations

- Trucks operate for a fixed 5-day workweek.
- Daily work, including driving and waiting time, is limited to a maximum of 13 hours to comply with Dutch regulations. The exemption allowing shorter daily rests is intentionally not scheduled to allow flexibility for potential delays during the execution.
- An 11-hour rest period is scheduled after reaching the daily work limit.
- Short breaks are not explicitly scheduled, as they typically occur during waiting times at pick-up or delivery locations.
- Drivers are not scheduled for overtime.

4.3 MIP

This section presents the mathematical model formulation concerning the optimization problem of Nijhof Wassink's DBL. It begins with defining the sets, parameters, and decision variables, followed by the objective function and the constraints. Then, the model solves some planning instances, and the feasibility of the model is evaluated.

4.3.1 Model Notation

Let $G(N, A)$ be a directed graph, where set N represents the nodes and set A represents the arcs between the nodes. For each pair of nodes i and $j \in N$, the arc from node i to node j is denoted as $ij \in A$. The graph, consisting of nodes and arcs, is referred to as a network. The requests to be fulfilled involve transporting loads from a pick-up to a delivery location. A request r is defined by its pick-up location $p(r)$, with time window $[a_{p(r)}, b_{p(r)}]$ and delivery location $d(r)$ with time window $[a_{d(r)}, b_{d(r)}]$.

In Nijhof Wassink's context, requests can be transferred at depot locations. To follow the notation of Rais et al. (2014), a dummy set of transfer locations, corresponding to each depot location is used [50].

Network Preprocessing

Since vehicles in this problem only transport full truck loads, certain arcs in the network cannot be part of a valid routing solution. For example, after visiting a pick-up location, a vehicle is loaded and cannot pick up another load until it has delivered or transferred the current load. Note that a location cannot serve as both a pick-up and delivery location for different requests. If a location functions as both, it is duplicated in the network to ensure each instance serves only one purpose. Also, vehicles must return to their assigned depot and are not permitted to visit other depots. Table 4.1 shows a summary of the feasible and infeasible arcs based on the types of nodes. Infeasible arcs are removed during the preprocessing to reduce the number of decision variables and, consequently, the problem's complexity.

From / to	Depot	Pick-up	Delivery	Transfer
Depot	$*^1$	✔	✗	✔
Pick-up	✗	✗	$*^2$	✔
Delivery	$*^1$	✔	✗	✔
Transfer	$*^1$	✔	✔	✔

$*^1$ Only to the vehicle's assigned depot

$*^2$ Only to the corresponding delivery of the pick-up

Table 4.1: Feasible and infeasible arcs in the network based on the type of nodes.

Sets

The sets are defined as follows:

K	indexed by $k = \{1, \ldots, \|K\|\}$	Set of vehicles
R	indexed by $r = \{1, \ldots, \|R\|\}$	Set of pick-up and delivery requests
O	indexed by $o = \{1, \ldots, \|O\|\}$	Set of depot locations
P	indexed by $p(r) \in N, \forall r \in R$	Set of pick-up locations
D	indexed by $d(r) \in N, \forall r \in R$	Set of delivery locations
T	indexed by $t = \{1, \ldots, \|T\|\}$	Set of transfer locations
$N = O \cup T \cup P \cup D$		Set of all nodes

Note that the sets O, T, P, and D are disjoint. Each transfer location is a dummy node of its corresponding depot location. Additionally, if a location serves as both a pick-up and delivery location in different requests, it is duplicated to ensure that each location only serves one purpose.

Parameters

The following parameters are defined:

$o(k)$	Start and end depot of vehicle $k \in K$, with $o(k) \in O$
d_{ij}	Travel distance on arc $ij \in A$
t_{ij}	Time required to move from node i to node j for $i, j \in N$
$[a_{p(r)}, b_{p(r)}]$	Time window for the pick-up of request $r \in R$
$[a_{d(r)}, b_{d(r)}]$	Time window for the delivery of request $r \in R$
$[a_{o(k)}, b_{o(k)}]$	Time window for the depot locations $o(k) \in O$ of vehicle $k \in K$
s_i	Service time at request node $i \in P \cup D$
w	Maximum length of a vehicle's working shift
f	Minimum length of a vehicle's resting period
α_i	Penalty costs of arriving early at node $i \in P \cup D$
β_i	Penalty costs of arriving late at node $i \in P \cup D$
ϵ	Fixed costs per unit travel distance without a load (empty kilometers)
γ	Fixed costs per unit travel distance with a load
δ	Fixed costs per utilized vehicle
M	Big M parameter, used for conditional constraints

Decision Variables

The decision variables are defined as follows:

X_{ij}^k	$= 1$ if vehicle $k \in K$uses the arc $ij \in A, 0$ otherwise
Y_{ij}^{kr}	$= 1$ if vehicle $k \in K$ carries request $r \in R$ on the arc $ij \in A, 0$ otherwise
U^k	$= 1$ if vehicle $k \in K$ is used, 0 otherwise
Z_{ij}^k	$= 1$ if node i (at any time) precedes node j on route of vehicle $k \in K$, where $i, j \in N, 0$ otherwise
S_{jr}^{kl}	$= 1$ if request $r \in R$ is transferred from vehicle $k \in K$ to vehicle $l \in K$, at node $j \in N, 0$ otherwise
T_j^k	$\in \mathbb{R}^+$ indicating the arrival time at node $j \in N$ with vehicle $k \in K$
$\bar{T}_j^k$	$\in \mathbb{R}^+$ indicating the departure time at node $j \in N$ with vehicle $k \in K$
D_i^k	$\in \mathbb{R}^+$ as a slack variable for arriving early at pick-up and delivery locations $i \in P \cup D$
$\bar{D}_i^k$	$\in \mathbb{R}^+$ as a slack variable for arriving late at pick-up and delivery locations $i \in P \cup D$
Q_i^k	$\in \mathbb{R}^+$ indicating the cumulative time worked after vehicle $k \in K$ visits node $i \in N$
B_{ij}^k	$= 1$ if vehicle $k \in K$takes a break after traversing arc $ij \in A, 0$ otherwise
W_{ij}^{kr}	$\in [0, 1]$ as an auxiliary variable, used for linearization of the objective function

4.3.2 Objective and Constraints

Objective

The model aims to minimize costs, consisting of different components. These are the following:

- Empty kilometers, representing the distance traveled without load. Scaled with a cost factor per unit distance traveled without a load, ϵ.

- Total distance traveled to discourage unnecessary traveling with a load. Scaled with the cost factor per unit distance traveled with a load, γ.
- Time window violations, which are calculated by the costs of arriving too early, α_i, and arriving too late, β_i. Costs are determined per request location.
- Number of vehicles used, multiplied by a cost factor per used vehicle, denoted by δ.

Combining these components gives the following objective function.

$$\underbrace{\epsilon \sum_{\substack{ij\in A,\\ k\in K}} d_{ij} X_{ij}^k \big(1 - \sum_{r\in R} Y_{ij}^{kr}\big) +}_{\text{Empty kilometers costs}} \quad \underbrace{\gamma \sum_{\substack{ij\in A,\\ k\in K}} d_{ij} X_{ij}^k}_{\text{Travel distance costs}} \quad + \underbrace{\sum_{\substack{i\in P\cup D,\\ k\in K}} \big(\alpha_i D_i^k + \beta_i \bar{D}_i^k\big)}_{\text{Violation of time windows costs}} \quad + \quad \underbrace{\delta \sum_{k\in K} U^k}_{\text{Use of vehicles costs}} \tag{4.1}$$

Note that the empty kilometers costs are nonlinear due to the multiplication of decision variables. To linearize this component of the objective, the McCormick relaxation constraints are used. By introducing the auxiliary decision variable $W_{ij}^{kr} = X_{ij}^k \cdot Y_{ij}^{kr}$, the empty kilometers component of the objective can be reformulated as follows:

$$\epsilon \sum_{\substack{ij\in A,\\ k\in K}} d_{ij} \big(X_{ij}^k - \sum_{r\in R} W_{ij}^{kr}\big)$$

Although linearizing an objective function does not always improve a model's performance, in this case, the linearized objective outperforms the quadratic objective. This is further discussed in Subsection 4.3.3. Additional constraints are required to bound the auxiliary variable and are explained in the next paragraph.

Constraints
To construct a feasible routing solution, the model has constraints for the truck and trailer availability, driver hour limitations, and request requirements. Each set of constraints is explained below.

McCormick Constraints

$$W_{ij}^{kr} \leq X_{ij}^k \qquad \forall ij \in A, k \in K, r \in R \tag{4.2}$$

$$W_{ij}^{kr} \leq Y_{ij}^{kr} \qquad \forall ij \in A, k \in K, r \in R \tag{4.3}$$

$$W_{ij}^{kr} \geq X_{ij}^k + Y_{ij}^{kr} - 1 \qquad \forall ij \in A, k \in K, r \in R \tag{4.4}$$

Constraints 4.2 and 4.3 provide upper bounds for the auxiliary variable by ensuring that it cannot exceed the value of each of the binary decision variables. Constraints 4.4 provide lower bounds for the auxiliary variable. Together, these constraints ensure that decision variable W_{ij}^{kr} behaves like $X_{ij}^k \cdot Y_{ij}^{kr}$.

Vehicle and Request Flows

$$\sum_{j:ij\in A} X_{ij}^{k} \leq 1 \qquad i = o(k), \forall k \in K \tag{4.5}$$

$$\sum_{j:ij\in A} X_{ij}^{k} - \sum_{j:ji\in A} X_{ji}^{k} = 0 \qquad i = o(k), i \neq j, \forall k \in K \tag{4.6}$$

$$\sum_{j:ij\in A} X_{ij}^{k} - \sum_{j:ji\in A} X_{ji}^{k} = 0 \qquad \forall i \in N \backslash O, \forall k \in K \tag{4.7}$$

$$\sum_{j:ij\in A} X_{ij}^{k} = U^{k} \qquad i = o(k), j \in N \backslash O, \forall k \in K \tag{4.8}$$

$$\sum_{k\in K} \sum_{j:ij\in A} Y_{ij}^{kr} = 1 \qquad i = p(r), \forall r \in R \tag{4.9}$$

$$\sum_{k\in K} \sum_{j:ji\in A} Y_{ji}^{kr} = 1 \qquad i = d(r), \forall r \in R \tag{4.10}$$

$$\sum_{k\in K} \sum_{j:ij\in A} Y_{ij}^{kr} - \sum_{k\in K} \sum_{j:ji\in A} Y_{ji}^{kr} = 0 \qquad \forall i \in T, \forall r \in R \tag{4.11}$$

$$\sum_{j:ij\in A} Y_{ij}^{kr} - \sum_{j:ji\in A} Y_{ji}^{kr} = 0 \qquad \forall i \in N \backslash \{T \cup \{p(r), d(r)\}\}, \forall r \in R, \forall k \in K \tag{4.12}$$

$$\sum_{k\in K} \sum_{j:ji\in A} Y_{ji}^{kr} = 0 \qquad i = p(r), \forall r \in R \tag{4.13}$$

$$\sum_{k\in K} \sum_{j:ij\in A} Y_{ij}^{kr} = 0 \qquad i = d(r), \forall r \in R \tag{4.14}$$

$$\sum_{r\in R} Y_{ij}^{kr} \leq 1 \qquad i \neq j, \forall ij \in A, \forall k \in K \tag{4.15}$$

$$Y_{ij}^{kr} \leq X_{ij}^{k} \qquad \forall ij \in A, \forall r \in R, \forall k \in K \tag{4.16}$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall ij \in A, \forall k \in K \tag{4.17}$$

$$Y_{ij}^{kr} \in \{0,1\} \qquad \forall ij \in A, \forall r \in R, \forall k \in K \tag{4.18}$$

$$U^{k} \in \{0,1\} \qquad \forall k \in K \tag{4.19}$$

Constraints 4.5 ensure that, at most, one route is initiated for each vehicle from its depot. Using "$\leq$" instead of "$=$" allows the model to use fewer vehicles than available. Constraints 4.6 ensure that each vehicle begins and ends its route at its assigned depot. Constraints 4.7 maintain flow conservation for vehicles throughout the nodes in the network. Constraints 4.8 indicate whether a vehicle is used, required for minimizing the total number of vehicles used. Constraints 4.9 and 4.10 ensure that all pick-ups and deliveries are fulfilled, respectively. Constraints 4.11 enforce flow conservation of requests at transfer nodes and restrict transfers to the designated nodes. On the other hand, constraints 4.12 maintain request flow at all other nodes, ensuring that if a request arrives at a node, it must leave with the same request. Depots are excluded from these constraints, as no request can be carried from and to the depot. Constraints 4.13 and 4.14 prevent requests from being carried on arcs before their pick-up or after their delivery. These constraints are needed to calculate the empty kilometers. Otherwise, the model may carry unnecessary loads to reduce the empty kilometers. Previous formulations did not require these constraints as they were independent of the objective function. Constraints 4.15 ensure that at most one request is carried on an arc for a given vehicle. Constraints 4.16 require vehicle flow if a request is carried on a certain arc. Note that constraints 4.5 to 4.7 ensure flow of vehicles and constraints 4.9 to 4.12 ensure flow of requests. 4.16 is the constraint that links the two flows. Constraints 4.17, 4.18 and 4.19 ensure that these decision variables are binary.

Subtour Elimination Constraints

$$X_{ij}^k \leq Z_{ij}^k \qquad \forall i,j \in N \backslash o(k), \forall k \in K \qquad (4.20)$$
$$Z_{ij}^k + Z_{ji}^k = 1 \qquad \forall i,j \in N \backslash o(k), \forall k \in K \qquad (4.21)$$
$$Z_{ij}^k + Z_{jl}^k + Z_{li}^k \leq 2 \qquad \forall i,j,l \in N \backslash o(k), \forall k \in K \qquad (4.22)$$
$$Z_{ij}^k \in \{0,1\} \qquad \forall i,j \in N, k \in K \qquad (4.23)$$

For feasible routing solutions, the possibility of vehicles driving in subtours should be eliminated. Constraints 4.20, 4.21, and 4.22 serve as subtour elimination constraints by enforcing precedence relations between nodes in linear ordering. Constraints 4.20 ensure that for any $X_{ij}^k = 1$, node i must precede node j as they are immediate predecessors in the vehicle flow. Constraints 4.21 allow that either node i precede j or node j precede i but not both. Note that the depot locations are excluded to allow vehicles to return to the depot. Constraints 4.22 avoid cycles between three distinct nodes visited by the same vehicle. Constraints 4.23 ensure that the decision variables are binary.

Time Window Constraints

$$T_{p(r)}^k + D_{p(r)}^k \geq a_{p(r)} \qquad \forall r \in R, \forall k \in K \qquad (4.24)$$
$$T_{d(r)}^k + D_{d(r)}^k \geq a_{d(r)} \qquad \forall r \in R, \forall k \in K \qquad (4.25)$$
$$T_{p(r)}^k - \bar{D}_{p(r)}^k \leq b_{p(r)} \qquad \forall r \in R, \forall k \in K \qquad (4.26)$$
$$T_{d(r)}^k - \bar{D}_{d(r)}^k \leq b_{d(r)} \qquad \forall r \in R, \forall k \in K \qquad (4.27)$$
$$T_{o(k)}^k \geq a_{o(k)} \qquad \forall k \in K \qquad (4.28)$$
$$\bar{T}_i^k + t_{ij} - b_{o(k)} \leq M(1 - X_{ij}) \qquad j = o(k), \forall i \in N \backslash O, \forall k \in K \qquad (4.29)$$
$$\sum_{j:ji \in A} Y_{ji}^{kr} + \sum_{j:ij \in A} Y_{ij}^{lr} \leq S_{jr}^{kl} + 1 \qquad \forall r \in R, \forall i \in T, k \neq l, \forall k,l \in K \qquad (4.30)$$
$$T_j^k - \bar{T}_j^l \leq M(1 - S_{jr}^{kl}) \qquad \forall r \in R, \forall j \in T, k \neq l, \forall k,l \in K \qquad (4.31)$$
$$T_{p(r)}^k, T_{d(r)}^k, D_{p(r)}, D_{d(r)}, \bar{D}_{p(r)}, \bar{D}_{d(r)}^k \geq 0 \qquad p(r) \in P, d(r) \in D, \forall r \in R, \forall k \in K \qquad (4.32)$$
$$\bar{T}_i^k \geq 0 \qquad \forall i \in N \backslash O, \forall k \in K \qquad (4.33)$$
$$S_{jr}^{kl} \geq 0 \qquad \forall j \in N, \forall r \in R, \forall k,l \in K \qquad (4.34)$$

Time windows in this problem are soft, meaning that violations are permitted but incur penalties. Constraints 4.24 and 4.26 ensure that vehicles arrive at the pick-up location after it opens and before it closes. 4.25 and 4.27 ensure that vehicles arrive at the delivery location between opening and closing. Constraints 4.28 and 4.29 enforce the same but for delivery locations. Constraints 4.30 maintain synchronization of requests during transfers between vehicles. Constraints 4.31 allow a request to transfer from a vehicle k to vehicle l, provided that vehicle k arrives at the transfer station before the departure of vehicle l. Constraints 4.32, 4.33, 4.34 enforce that the decision variables are nonnegative.

Driver Hours Constraints

$$\bar{T}_i^k + t_{ij} + f \cdot B_{ij}^k - T_j^k \leq M(1 - X_{ij}^k) \qquad \forall i,j \in N, \forall ij \in A, \forall k \in K \qquad (4.35)$$
$$Q_i^k + t_{ij} + s_j - w \cdot B_{ij}^k - Q_j^k \leq M(1 - X_{ij}^k) \qquad \forall i,j \in N, \forall ij \in A, \forall k \in K \qquad (4.36)$$
$$Q_i^k - w \leq M(1 - B_{ij}^k) \qquad \forall i \in N, \forall ij \in A, \forall k \in K \qquad (4.37)$$
$$B_{ij}^k \leq X_{ij}^k \qquad \forall ij \in A, \forall k \in K \qquad (4.38)$$
$$B_{ij}^k \in \{0,1\} \qquad \forall ij \in A, \forall k \in K \qquad (4.39)$$
$$Q_i^k \geq 0 \qquad \forall i \in N, \forall k \in K \qquad (4.40)$$

Constraints 4.35 calculate the arrival time at node j if vehicle k has traversed arc ij. These constraints account for the travel time and the mandatory break time if a break is required. The constraints are nonbinding if the arc ij is not traversed by vehicle k. Constraints 4.36 update a vehicle's cumulative working time after visiting a node. When a break is taken, the cumulative working time is reset to zero. Constraints 4.37 force a vehicle to take a break after its cumulative driving time exceeds the limit. Constraints 4.38 require breaks to be taken on an arc included in the vehicle's route. Constraints 4.39 and 4.40 ensure that the decision variables are binary and nonnegative, respectively.

4.3.3 Model Size and Feasibility

To evaluate the computational feasibility of the model, test instances with varying numbers of depot locations ($|O|$), requests ($|R|$), and vehicles ($|K|$) are defined. The input parameters are derived from the historical planning data. The travel distances were assumed to be Euclidean, and vehicle speeds are considered constant at 60 km/h. The cost factor weights were defined as: $\epsilon = \gamma = 10, \delta = 60, \alpha_i = \beta_i = 5$. The verification of the distribution weights is left for the heuristic solution, which aims to provide practical results. The experiments were conducted using an 8 GB RAM system with the Gurobi optimizer in Python, and a 30-minute time limit was imposed per run.

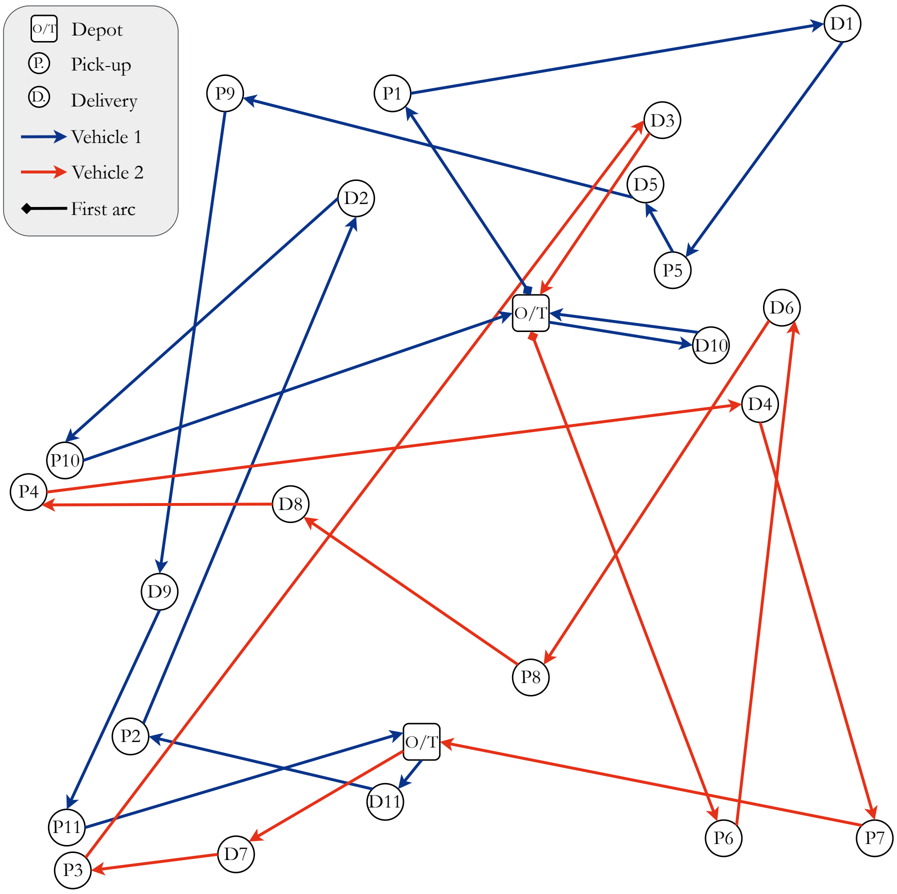


Figure 4.2: Routing solution of the instance with 2 depots, 11 requests, and 3 vehicles.

To provide an intuitive understanding of the model's output, Figure 4.2 displays the optimal routing solution for the solved instance with 2 depots, 11 requests and 3 vehicles. The solution deploys only 2 vehicles, both starting from the same depot. The figure illustrates how the model assigns requests to vehicle routes while minimizing the objective components subject to the given constraints. Notable is that vehicles visit transfer stations en route from the pick-up location to the delivery location, even when no actual transfer takes place. This occurs because transfer stations do not impose waiting time penalties, whereas pick-up and delivery locations incur penalties for early and late arrivals. As a result, the model uses transfer stations as intermediate waiting points to reduce time window penalties. While the model permits this behavior, a stricter formulation could have restricted transfer visits to only when an actual transfer takes place. However, this was not enforced in the current formulation.

Table 4.2 summarizes the results of the experiments with the linearized objective function. Although the original quadratic objective function more directly presents the objective in this problem context, the model with the quadratic objective consistently led to larger optimality gaps and longer computation times for the same experiment settings. Therefore, the linearized objective was chosen for the computational experiments.

Instances			Model size			Experiment results		
$\|O\|$	$\|R\|$	$\|K\|$	Binary variables	Continuous variables	Constraints	Objective (€)	CPU time (s)	Optimality gap (%)
1	11	2	6262	4654	43085	4552	153	-
1	14	3	16779	12912	125139	5872	1802	3.95%
2	9	3	7542	5346	49698	4578	33	-
2	11	3	11646	8574	81824	5086	1533	-
2	13	2	11248	8572	83496	5226	1801	9.33%
3	9	3	9240	6489	63303	4488	145	-
3	11	3	13968	10233	101023	4997	1801	1.82%
4	12	4	26308	19404	19212	5765	1803	2.51%

Table 4.2: Computation time of exact model experiments.

While the illustrative example shows that the model can generate feasible and interpretable routing solutions for small instances, Table 4.2 shows that the model's scalability is limited. As expected, the number of decision variables and constraints increases rapidly as the number of depots, requests, and vehicles increases. The test case with 3 depots, 9 requests, and 3 vehicles was solved to optimality within the time limit. Larger instances were not solved to optimality within the time limit, and the optimality gap increases with the problem size. To apply this model to the DBL planning, the model requires, on average, 8 depots, 97 vehicles, and 360 requests per week. This amounts to $1.18 \cdot 10^{11}$ integer variables, $1.12 \cdot 10^{11}$ continuous variables and $3.42 \cdot 10^{11}$ constraints. This practical problem size is significantly greater than what this exact model can handle within reasonable computation times. Consequently, these results emphasize the need for a heuristic approach to handle large-scale instances.

4.4 Conclusion

This chapter has developed and evaluated a mathematical model to represent Nijhof Wassink's optimization problem. Firstly, the optimization problem is outlined, and the modeling assumptions are discussed. Then, the problem is defined mathematically by introducing sets, parameters, and decision variables. The objective and constraints explain the goal and restrictions of the model. Lastly, the chapter concludes with an evaluation of the computation time of the exact model. This also serves as a motivation for the design of a heuristic solution, proposed in Chapter 5.

5| Heuristic Approach

The previous chapter highlighted the need for a heuristic approach to solve Nijhof Wassink's optimization problem. Therefore, this chapter focuses on the design of a heuristic approach. Section 5.1 provides a high-level overview of the algorithm. To initialize the heuristic, a constructive heuristic is designed in Section 5.2. To evaluate the performance of the improved routing solutions, the objective function tailored to the designed heuristic is given in Section 5.3. Sections 5.4 and 5.5 elaborate on the generation of neighborhood solutions and the adaptive weight mechanism. Finally, the stopping criteria are explained in Section 5.6. To summarize the algorithm, it concludes with a complete pseudocode.

5.1 Overview

A suitable heuristic to solve Nijhof Wassink's optimization problem is the Adaptive Large Neighborhood Search (ALNS). The general layout of the algorithm is illustrated in Figure 5.1. ALNS uses an initial feasible solution as a starting point. It then iteratively improves the solution using destroy and repair operators. In each iteration, a pair of operators is selected to destroy the current solution and repair it into a new feasible solution. The solution is then accepted based on its objective value and some acceptance criteria. This allows diversification and intensification. The algorithm updates the current solution. If the new solution is also better than the best-known solution so far, it is saved. Thereafter, the algorithmic parameters are updated accordingly. This process repeats until the stopping criteria are met and the best solution is returned.

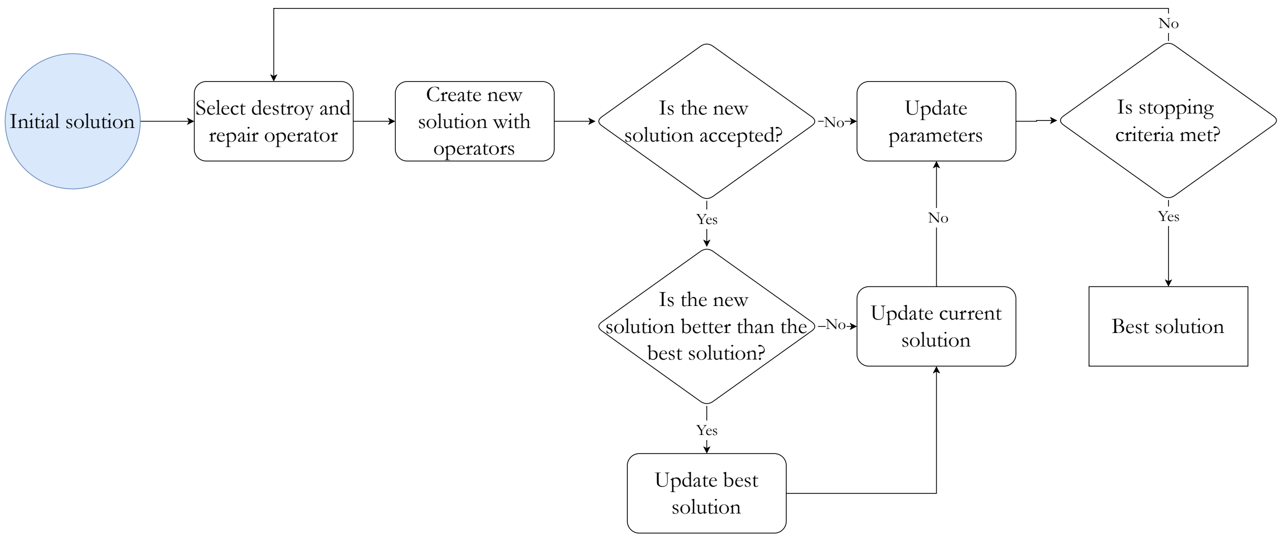


Figure 5.1: Flowchart of overview ALNS

5.2 Initial Solution

ALNS uses the initial solution as a starting point. From this solution, it begins to search for neighboring solutions. The initial solution is constructed with a randomized greedy heuristic. The process is outlined in Figure 5.2. A more elaborate pseudocode can be found in Appendix E. The algorithm iteratively assigns each request to a vehicle route. First, requests are sorted by some metric, such as the earliest pick-up or delivery time window opening time. The order of the vehicle routes is shuffled for some randomness. In each iteration, a request is selected and inserted with a stop at a transfer station, or the same vehicle handles the pick-up and delivery. A subset of requests is inserted with a transfer, as Sampaio et al. have shown that initial solutions with transferred requests are more likely to utilize transfers effectively [57]. If a request can be inserted into multiple vehicle routes, a vehicle is chosen based on weighted random selection. This means smaller detour distances from the current vehicle location to the request's pick-up location are more likely to be selected. If insertion with a transfer fails for every combination of vehicles, a direct insertion is attempted. Only if no vehicle can feasibly fulfill the direct request may the request be partially fulfilled. This means the pick-up is visited, and the request is stored at a transfer station for delivery in the next planning period. The partial request will be finished in the next planning period. If no feasible vehicle is available for the request, it is marked as unplanned. The process continues until all requests are assigned or marked as unplanned. The partial and unplanned requests are marked and will be addressed during the improvement phase. The relaxation allowing partial requests is added to increase the number of requests served in the initial solution. Without this addition, a larger number of requests would be marked as unplanned, resulting in a decreased routing performance. The relaxation is further justified as partial requests are also used in real-world operations, for example, after a driver's shift has ended. Partial requests, therefore, make the initial solution more effective and better aligned with practical planning strategies. However, partial requests are not allowed in the improvement heuristic, as is discussed in Section 5.4.

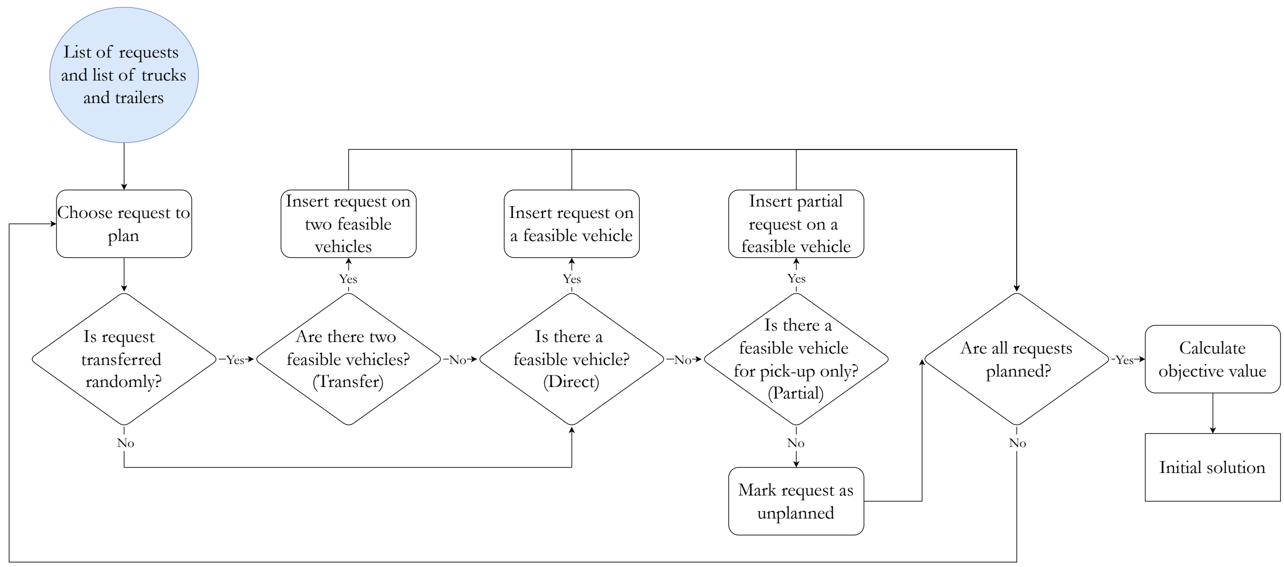


Figure 5.2: Flowchart of construction of initial solution

5.3 Objective Function

Each routing solution is scored with an associated objective value. The objective function for the algorithm is similar to the objective function from the exact model (see Equation 4.1). The objective aims to minimize empty kilometer costs, total travel distance costs, time window violation costs for both early and late arrivals (combined), and use of vehicles costs. While minimizing empty kilometers remains the primary objective, the inclusion of the other components is essential to obtain feasible routing solutions in practice. As discussed in Section 4.1, focussing solely on empty kilometers can lead to unrealistic or inefficient routing strategies, such as excessive detours or vehicles used to avoid empty kilometers. These added objective components, therefore, ensure more practical and efficient routing solutions.

Unlike the exact model from the previous chapter, the heuristic allows requests to be left unserved. In the exact model, fulfilling all requests is a hard constraint, as it can still find optimal solutions even with this constraint. While serving all requests is preferable from a practical perspective, enforcing this as a hard constraint would often result in infeasible routing solutions and restrict the algorithm's ability to explore the neighboring solution space. Therefore, the heuristic allows for more flexibility, which is especially important in large problem instances. Each component is scaled with a scaling factor, respectively $(s_1, \ldots, s_5)$ to balance their impact based on priority and magnitude. This is crucial as the components are in different units and scales. For the remainder of this study, the objective function $Z(S)$ for a routing solution S will be calculated as follows:

$$Z(S) = s_1 C_{empty} + s_2 C_{dist} + s_3 C_{timewindows} + s_4 C_{vehicles} + s_5 C_{unserved} \tag{5.1}$$

The load factor will also be used to evaluate the routing solution. This is derived from the empty and total distance, as defined in Equation 2.1.

5.4 Neighborhood Solutions

To create neighborhood solutions, operators are needed to adapt the solution. Like in many other ALNS algorithms, destroy operators are meant to destroy a set of requests from the solution. The removed requests are temporarily stored in a request bank. The repair operators then reconstruct the solution differently by reinserting the requests from the request bank. If a request cannot feasibly be inserted, it remains in the request bank. Note that the neighborhood solutions are restricted to either fully serving a request or not serving it at all. In particular, partial requests are not allowed. This is because the improvement phase is designed to optimize routing solutions by better request planning rather than by optimizing the use of partial fulfillment, which should remain a last-resort measure. However, the algorithm could have included a fallback to reintroduce partial requests during the repair phases. However, this was omitted due to the additional computational complexity and time constraints. After each removal or insertion of requests, the arrival times in the routing solution are fully updated. This maintains feasibility and consistency after each repair. This is particularly important for requests with transfers, as the vehicle assigned to the delivery cannot depart from the transfer station until the vehicle carrying the pick-up arrives. Ideally, the selection of destroy operators should contain a mix of heuristics that can diversify and intensify the search. Repair operators should focus on repairing the destroyed solution in a computationally efficient way. Each operator aims to improve a certain part of the solution and, by doing so, contributes to lowering the overall objective function. The following subsections describe the operators. For more detailed descriptions, pseudocodes can be found in Appendix F.

5.4.1 Destroy Operators

The set of destroy operators, $DO = \{DO_1, \ldots, DO_6\}$, used for this ALNS heuristic are described below. An important parameter for these operators is the degree of destruction, denoted as DOD where $DOD \in [0, 1]$. This value represents the fraction of requests that are temporarily removed from the routing solution. A low value results in small adaptations to the current solution, limiting the potential for improvement in the repair phase. On the contrary, a high value causes rigorous changes to the routing solution. This increases the potential to escape local minima but also risks losing too much solution integrity. This can make it difficult for repair operators to reconstruct a feasible solution.

DO_1 Random destroy: randomly removes DOD requests that are not transferred in the current routing solution. This operator can help diversify the search.

DO_2 Worst destroy time windows: removes DOD requests that have the highest contribution to the time window penalty. The penalty accounts for stricter locations, making requests with unflexible time windows more likely to be chosen. Requests with a transfer cannot be selected.

DO_3 Worst vehicle: removes all requests of a vehicle route based on load factor. A route with a lower load factor is more likely to be chosen than a route with a high load factor. This cleans up inefficient vehicle routes.

DO_4 Transfer station removal: removes DOD transferred requests from the same transfer station. The transfer station is selected based on its cumulative synchronization time. Synchronization time refers to the waiting time at a transfer station when the vehicle assigned to the delivery arrives before the vehicle carrying the pick-up. In such cases, the delivery vehicle must wait until the request is dropped off.

DO_5 Related requests removal: removes pairs of requests that are related. Their relatedness is calculated based on the overlap in time windows and the spatial closeness of locations for both the pick-up and delivery. First, a random set of $\frac{\text{DOD}}{2}$ requests is drawn. Then, for each request, a related request is selected based on its relatedness score. Removing similar requests allows repair operators to explore new combinations. They can be interchanged in the solution without significantly increasing the objective function.

DO_6 Long distance request removal: removes DOD requests based on their travel distance. Requests with a longer distance between pick-up and delivery are more likely to be chosen because they have a high influence on the objective value if placed inefficiently.

5.4.2 Repair Operators

The set of repair operators, RO $= \{RO_1, \ldots, RO_5\}$, used in this ALNS framework are described as follows:

RO_1 Random insertion: inserts as many requests as possible from the request bank. For each request, it randomly selects a feasible vehicle and position within the route. It only inserts requests without a transfer. This operator diversifies the solution space.

RO_2 Greedy detour insertion: inserts requests directly by selecting the route and position, resulting in the least detour distance. The detour is calculated as the additional distance caused by the insertion. This operator prioritizes minimizing the distance components of the objective function.

RO_3 Regret-2 empty distance: inserts requests based on the difference in empty kilometers between its best and second-best routes. This is referred to as the regret value. Requests with the highest regret are chosen first, as postponing their insertion may lead to a larger increase in empty distance later.

RO_4 Regret-2 time windows: inserts requests similar to RO_3, except the regret is calculated based on time window penalties. Requests are chosen based on how much worse the penalties may become if not inserted now.

RO_5 Best transfer: inserts requests via transfer stations. It only considers requests that have a high insertion cost if inserted directly. The cost is calculated using the objective function, as shown in Equation 5.1 with $s_5 = s_6 = 0$, applied to a single route. Then, it selects the best combination of transfer station, vehicle for pick-up, and vehicle for delivery. The insertion should result in the least increase in objective value and the shortest synchronization time between the two vehicles.

5.5 Adaptive Weight Operators

The adaptability of the ALNS lies in the dynamic selection of destroy and repair operators based on their historical performance. This is achieved with a roulette wheel selection principle, as also explained by Ropke and Pisinger in [54].

Operator Selection
Let k denote the number of operators, each associated with a weight w_i, for $i \in \{1, \ldots, k\}$. The weights determine the selection probability of each operator. In particular, the selection probability of operator j is given by $p_j = \frac{w_j}{\sum_{i=1}^{k} w_i}$. Note that the probabilities are a normalization of the weights. At each iteration, a destroy operator, DO_{i^*}, and a repair operator, RO_{j^*}, are selected independently based on these probabilities. The selected operators then transform the current routing solution S_{current} into a new routing solution: $S_{\text{new}} = RO_{j^*}(DO_{i^*}(S_{\text{current}}))$. This selection mechanism ensures a balance

between favoring operators based on historical successes and allowing less effective operators to be selected for diversity.

Markov Segments
The entire search is divided into segments, each with a Markov chain length of L iterations. The Markov chain length determines the number of iterations in which parameters, such as operator weights and temperature, remain fixed. A smaller value for L updates their parameters frequently and encourages a faster convergence with limited exploration of worse solutions. In contrast, a higher value allows for more extensive exploration of the solution space before updating but leads to slower convergence. After each set of L iterations, the weights are updated.

Scoring Mechanism
To track the performance of the operators during a segment, each operator $i \in \{1, \ldots, k\}$ accumulates a total score of π_i. This total score is incremented with scores $\sigma_1, \sigma_2, \sigma_3$, or 0 whenever operator i is selected, based on the quality of the new routing solution. If the new solution, S_{new}, improves the global best solution S_{best}, then score σ_1 is added to the total score of the destroy and repair operators. If the new solution is accepted and is better than the current solution but worse than the best solution, σ_2 is added to the destroy and repair operator's total score. Lastly, the destroy and repair operators are scored with σ_3 if the new solution is accepted but is worse than the current solution. If the solution is not accepted, the total score remains unchanged. This is also summarized in Table 5.1. These scores are predefined constants that require parameter tuning. Their relative magnitudes influence the algorithm's search behavior. A high value for σ_1 relative to σ_2 and σ_3 favors operators that lead to global improvements, encouraging the exploitation of the solution space. Conversely, higher values for σ_2 and σ_3 encourage less effective operators.

Score	Condition
σ_1	If S_{new} is accepted and $Z(S_{\text{new}}) < Z(S_{\text{best}})$
σ_2	If S_{new} is accepted and $Z(S_{\text{best}}) < Z(S_{\text{new}}) < Z(S_{\text{current}})$
σ_3	If S_{new} is accepted and $Z(S_{\text{new}}) > Z(S_{\text{current}})$
0	Otherwise

Table 5.1: Parameters for score adjustment.

Updating Weights
At the end of a segment, the weights of each operator are updated based on its current weight, accumulated score, and the predetermined reaction factor, ρ. The latter controls how quickly the weights adapt to new scores. A low value for ρ results in a slower adaptation. This is useful if operator performance is highly variable. In contrast, a high value for ρ reacts quickly to recent performance and is used for cases with rapidly changing operator effectiveness.

Let $w_{i,l}$ be the weight of operator $i \in \{1, \ldots, k\}$ in segment l and let θ_i be the number of times operator i was selected during segment l. Then, the new weight for operator i in segment $l+1$ is computed as follows:

$$w_{i,l+1} = w_{i,l}(1-\rho) + \rho\frac{\pi_i}{\theta_i} \tag{5.2}$$

The operator's initial weights, before the first segment starts, are typically set equally to ensure a fair early exploration. However, they can also be differentiated based on results from preliminary experiments or problem-specific insights to guide the start of the search.

5.6 Acceptance and Stopping Criteria

To avoid the risk of the heuristic getting trapped in local minima, it uses acceptance and stopping criteria based on the principles of simulated annealing. The acceptance criterion is defined as follows:

if the candidate solution, S_{new}, is better than the current solution S_{current}, it is always accepted. Otherwise, it is accepted with a probability of:

$$e^{-(Z(S_{\text{new}})-Z(S_{\text{current}}))\cdot\frac{1}{T}} \tag{5.3}$$

where $T > 0$ is the current temperature. This parameter controls the degree of exploration during the search process. A higher temperature is more likely to accept worse solutions to help escape local minima, whereas a lower temperature favors the exploitation of promising solution spaces. The temperature is reduced at each segment of L iterations using a cooling schedule, defined as $T \leftarrow T \cdot c$, where $0 < c < 1$ is the cooling rate. The cooling rate determines how quickly the temperature decreases in each segment. A higher value for the cooling rate results in a more thorough exploration of the solution space and slower convergence. In contrast, a lower value reduces the likelihood of accepting worse solutions faster, resulting in faster convergence but a higher risk of getting trapped in a local minimum. The algorithm starts with the initial temperature, T_{init}, and terminates when the temperature T falls below some temperature threshold T_{end}.

5.7 ALNS Overview

The pseudocode in Algorithm 1 integrates all discussed components from the preceding sections. Before the improvement phase begins, all partially fulfilled requests are removed from the initial solution. These, together with requests marked as unplanned, are used to initialize the request bank, denoted by RB. In this way, the improvement phase can reassign the initially infeasible requests to the improved routing solution.

Algorithm 1 Adaptive Large Neighborhood Search (ALNS).

Input: T_{init}, T_{end}, c, L, $\{DO_1, \ldots, DO_6\}$, $\{RO_1, \ldots, RO_5\}$
Output: S_{best}
$S_0 \leftarrow$ ConstructInitialSolution() ▷ Section 5.2
Initialize $S_{\text{best}} \leftarrow S_0$ without partial requests
Initialize $S_{\text{current}} \leftarrow S_0$ without partial requests
Initialize $RB \leftarrow$ PartialRequestsInitialSolution()
Initialize $T \leftarrow T_{\text{init}}$
Initialize $1 \leftarrow w_d, w_r, \forall r \in RO, d \in DO$
Initialize $0 \leftarrow \pi_d, \pi_r, \forall r \in RO, d \in DO$
while $T > T_{\text{end}}$ **do**
  for $l \in \{1, \ldots, L\}$ **do**
    Select destroy operator d based on w_d, and update $\pi_d, d \in DO$ ▷ Section 5.4.1
    Select repair operator r based on w_r, and update $\pi_r, r \in RO$ ▷ Section 5.4.2
    $S_{\text{new}} := r(d(S_{\text{current}}))$
    if $Z(S_{\text{new}}) < Z(S_{\text{current}})$ **then** ▷ Section 5.3
      $S_{\text{current}} \leftarrow S_{\text{new}}$
      if $Z(S_{\text{new}}) < Z(S_{\text{best}})$ **then**
        $S_{\text{best}} \leftarrow S_{\text{new}}$
      end if
    else if RandomAccept(S_{new}) = True **then** ▷ Section 5.6
      $S_{\text{current}} \leftarrow S_{\text{new}}$
    end if
    Update RB based on S_{current} ▷ Section 5.4
  end for
  Update w_d, w_r based on π_d, π_r, respectively ▷ Section 5.5
  $T \leftarrow T \cdot c$
end while

5.8 Conclusion

This chapter presents an ALNS algorithm tailored to the routing problem of Nijhof Wassink. The set of destroy and repair operators is designed to improve the routing solution reflecting the company's priorities. Furthermore, it elaborated on the mechanisms incorporated to balance exploration and exploitation of the solution space. The algorithm serves as a basis for the computational experiments in Chapter 6.

6| Experiments & Results

This chapter describes the experiments and results obtained using the previously described ALNS algorithm. First, the input data provided by Nijhof Wassink is discussed in Section 6.1. Then, the input parameter of the initial solution and the objective weights are determined in Section 6.2 and Section 6.3. The algorithmic parameter tuning is tested in Section 6.4. Thereafter, the ALNS algorithm's performance is compared to both the initial solution and the historical routing plan in Section 6.5. This is followed by a scenario analysis to find the solution's sensitivity to certain input parameters.

6.1 Data Input DBL Nijhof Wassink

Two initial data sets from Nijhof Wassink's DBL are used as input data for the optimization model.

Historical Data
The first data set is the historical routing plan from the planning department. It contains detailed route information for each vehicle, including which request was assigned to which vehicle, when the pick-ups and deliveries of each request took place, and if any transfers occurred. The data set is split into 5-day periods, each creating a test instance. The performance of the optimization model is compared to the historical routing plan quality. To eliminate any external factors, such as traffic congestion and road conditions, in the historical data, the routes are recalculated using Euclidean distance, assuming a constant vehicle speed of 60 km/h.

Open Requests
The second data set, referred to as "open requests", contains all requests that are accepted by customer service but have yet to be assigned to a vehicle. Each request consists of a pick-up and a delivery, defined by their locations, time windows, and service times. Based on experience, the planning department has categorized each location by the strictness of its time windows. The first category includes flexible locations, where deviations from the time windows incur minimal penalties, and the vehicle can always be served at arrival. The second category operates using deadlines, allowing early arrivals to be served immediately but incurs high penalties for late arrivals. The last category enforces strict time windows. Any deviations incur substantial penalties, and vehicles must wait until the slot opens if arriving early. Moreover, the outsourced requests are filtered from the data as they are managed externally rather than by Nijhof Wassink. A test instance of open requests is derived from the requests handled in the corresponding 5-day historical period, allowing a direct comparison.

6.2 Initial Solution

The initial solution is generated with a randomized greedy insertion heuristic, as explained in Section 5.2. Different sorting metrics are used to test the difference in objective value. First, requests are sorted based on the opening of the pick-up time window. This aims to maximize vehicle utilization at the start of the week and minimize the time window penalties for the pick-ups. Alternatively, the sorting is based on the earliest delivery time window, which minimizes the time window penalty incurred at the delivery locations. Lastly, the construction heuristic may prioritize requests with the least slack time between a request's pick-up and delivery. This ensures that requests that are harder to schedule are given the best chance of fitting into the routing plan. Furthermore, probabilities $p_{\text{transfer}} \in \{0.01, 0.05, 0.15, 0.2\}$ for inserting a request with transfer are tested. The experiments are done for all combinations of p_{transfer} and sorting methods for 25 data weeks. Each data week has 10 runs to account for any variability caused by randomness. This amounts to 1500 experiments. The results are displayed in Table 6.1. The table compares the mean total objective value across all weeks.

As the number of requests varies over the weeks, the objectives per week have different magnitudes. To make a fair comparison, the week's objective is normalized. For each week, the objective values for the initial settings are normalized as follows:

$$Z_{\text{scaled}}(S) = \frac{Z(S) - Z(S)_{\min}}{Z(S)_{\max} - Z(S)_{\min}} \tag{6.1}$$

Here, $Z(S)_{\min}$ and $Z(S)_{\max}$ are the minimum and maximum objective values of that week, respectively. This scaling ensures that each week's objective values are scaled between 0, representing the best objective, to 1, representing the worst. The column "Obj(scaled)" represents the average normalized objective across all weeks. Furthermore, the standard deviation of the total scaled objective, average load factor, average unserved requests, average used vehicles, and average computation time are included.

The results show that the earliest delivery sorting method yields slightly better results compared to the earliest pick-up sorting method. The least slack method yields the worst mean objectives. The transfer probability of 0.01 yields the best mean objective values and leaves an acceptable amount of requests unserved. However, as mentioned in Section 5.2, transferred requests in the initial solution are essential for the improvement phase to work effectively. Therefore, the best initial parameter settings are based on the second-best experiment, which uses the sorting method with the earliest delivery and $p_{\text{transfer}} = 0.05$. These settings achieve comparable standard deviation, requests unserved, and vehicles used. The computation time is deemed insignificant compared to the improvement phase. The remainder of the experiments uses these parameters to generate the initial solution.

p_{transfer}	Sorting Method	Obj(scaled)	Obj(scaled) SD	LoadFactor	Unserved	Vehicles	Time(s)
0.01	Earliest Delivery	0.0564	0.0703	0.5871	11.26	94.70	3.72
0.05	Earliest Delivery	0.0842	0.0974	0.5818	12.88	93.94	4.42
0.01	Earliest Pick-up	0.1282	0.0794	0.5872	10.79	95.04	3.74
0.05	Earliest Pick-up	0.1967	0.0939	0.5793	11.48	94.73	4.49
0.10	Earliest Delivery	0.2180	0.1262	0.5748	15.79	93.53	6.00
0.10	Earliest Pick-up	0.2590	0.1241	0.5729	14.23	93.99	5.87
0.15	Earliest Delivery	0.3708	0.1647	0.5644	19.67	93.34	8.20
0.15	Earliest Pick-up	0.3805	0.1196	0.5642	19.11	93.21	8.15
0.20	Earliest Delivery	0.5048	0.1275	0.5530	22.38	93.51	10.48
0.20	Earliest Pick-up	0.5862	0.1865	0.5534	22.99	93.97	10.77
0.05	Least Slack	0.6311	0.1568	0.5834	12.82	95.72	4.76
0.01	Least Slack	0.6438	0.1874	0.5871	12.29	96.41	3.71
0.10	Least Slack	0.6985	0.1731	0.5755	15.09	95.17	6.47
0.15	Least Slack	0.8138	0.1523	0.5678	19.24	94.86	8.93
0.20	Least Slack	0.9508	0.1557	0.5570	22.80	94.93	11.41

Table 6.1: Experiments on initial solution settings.

6.3 Objective Weights

The objective weights, $s_1, \ldots, s_5$, as mentioned in Section 5.3, are determined by empirical testing of different weight configurations. Since the objective components differ in scale and importance, the weights have two key roles. First, to normalize the components to make them comparable despite their different units. Second, reflect the relative importance of the components in this problem context. As a result of these roles, the weights are unitless and not physically interpretable. The weight selection for the experiments was guided by empirical observations on the resulting routing solutions. The other parameters involved in the ALNS are not yet tuned and are set to reasonable default values. This is sufficient at this stage, as the focus lies on the sensitivity of objective weights rather than solution quality. The relative importance of weights is tailored to Nijhof Wassink's business case. Their main goal is to minimize the empty kilometers. Therefore, this component should be weighted most heavily. However, to retain practically feasible routing solutions, the other objective components are also weighted accordingly. The distance was given a lower weight. It is considered a less critical factor, as this is naturally incurred by serving requests and is partially accounted for in the empty kilometers component. Violating time windows are penalized to encourage timely pick-ups and deliveries. However, an excessively high value is very restrictive in the solution search. This ultimately results in unfavorably low load factors. Moreover, unserved requests are penalized to preserve solution flexibility. However, tests showed that the penalty must be sufficiently high to prioritize servicing requests. Too low values cause the algorithm to return near-empty solutions to avoid violating any other constraints. Lastly, the costs of using a vehicle are necessary to promote efficient use of the resources. It is observed that low values lead to high vehicle idle times. Based on these observations, the final objective weights are scaled as follows: $(s_1, s_2, s_3, s_4, s_5) = (0.25, 0.05, 0.25, 1250, 500)$.

The load factor is an important KPI for the routing solution. During the experiments, the solutions showed a substantial amount of empty kilometers from departing from the depot to the first route location and returning to the depot. Due to the modeling assumption that requests need to be fulfilled completely, these two travel actions are always driven empty. However, in practice, requests can stay at the transfer location over the weekend, reducing empty kilometers. To make a fair comparison between the modeled solution and the historical data, a modified version of the load factor will be presented. Instead of the ratio between total empty kilometers and the total driven distance, we use the empty and total distance driven between the first location after the depot and the last location before returning to the depot. Therefore, the first and last travel actions from and to the depot have no effect on the load factor for both the historical and the model routing solution.

6.4 Parameter Tuning

In this section, the parameters of the ALNS algorithm are tuned. This is important for the solution quality. The parameters that require tuning are the ones involved in the simulated annealing criteria, namely $T_{\text{init}}, T_{\text{end}}, c$, and L, and the ALNS parameters, namely the degree of destruction (DOD) and the reaction factor (ρ). The tuning is divided into two stages to manage complexity. First, the simulated annealing parameters are tuned. These are fixed after finding promising values for those parameters, and then ALNS parameters are tuned. The parameter tuning is done on 2 different representative data sets. The tuned parameters are assumed to also work well for the other data sets. Additionally, a limit of 30 minutes has been set for each experiment to manage computational resources. Given the large number of potential configurations and limited computational resources, this tuning approach is chosen to reduce the search space.

6.4.1 Simulated Annealing Parameters

The parameters T_{init}, T_{end}, c and L work closely together in the acceptance and stopping criteria. Their influence on the solution search is discussed in Section 5.6. Therefore, these are tuned simultaneously. Even with four parameters, the number of possible configurations is large. An iterative approach is adopted to manage this. Initially, a set of candidate values is defined for each parameter. The elements in the set span a wide range, covering a large area of the parameter space. A total of 18 random parameter combinations are sampled from these sets and tested on two datasets, resulting in 36 experiments per iteration. Then, the performance of the experiments is evaluated, and a refined set is determined, focusing on the most promising values. This process is repeated twice to obtain a refined set of parameters with promising results. The ALNS parameters DOD and ρ are kept constant during the experiments. Specifically, DOD was fixed at 10%, meaning that a tenth of the total requests were destroyed in each iteration. In some preliminary testing, this value appears to balance between the diversification of the solution and the preservation of sufficient solution structure, enabling effective repair. The reaction factor was set to 0.5, which is an average responsiveness to recent operator effectiveness.

The first iteration has a wide range of values, facilitating a broad exploration of the parameter space. The initial range of T_{init} is set such that there is approximately a 50% chance of accepting a worse solution in the first iteration of the ALNS. This can be computed with Equation 5.3. The most promising parameter combinations were identified based on the lowest total objective, a relatively low percentage of unserved requests, and a reasonable runtime. Total objectives have been normalized to compare across different weeks. Consistency has also contributed to the selection of parameter values. That is, parameters that performed well in multiple configurations are favored. Table 6.2 displays the parameter sets for each iteration. Each parameter value is accompanied by the average normalized objective value of each solution in which this value occurs. The average excludes solutions with more than 2 % unserved requests and solutions that reached the time limit. The most promising parameter values are marked with an asterisk (*). The second and third iterations use narrower parameter ranges, which are centered around the marked values from the previous iteration. The annotated values highlight noteworthy selection considerations. Note [1] is a second marked value as values 10 and 20 yield similar promising results. Notes [2] and [3] indicate values that appear to outperform the chosen promising value but, in some cases, lead to solutions that did not converge within the time limit. The full results of the iterations are presented in Appendices G.1, G.2 and G.3. The final selected parameter set is $(T_{\text{init}}, T_{\text{end}}, c, L) = (5000, 90, 0.94, 15)$.

Iteration	T_{init}		T_{end}		c		L	
	Value	Avg. obj.	Value	Avg. obj.	Value	Avg. obj.	Value	Avg. obj.
Iteration 1	3000	0.2457	10	0.3254	0.9	0.3276	10*	0.2357
	5000*	0.2094	50*	0.2107	0.95*	0.2666	20*[1]	0.2361
	7000	0.2548	100*	0.2419	0.99	-	30	-
Iteration 2	4000	0.4955	60	-	0.925	-	10*	0.4206
	5000*	0.3253	80	0.5116	0.95*	0.4372	15*	0.4129
	6000	0.5116	100*	0.3820	0.975[2]	0.4206	20	0.4453
Iteration 3	4500	0.5164	90*	0.3567	0.94*	0.3043	12	0.5506
	5000*	0.2858	100	0.3756	0.95	0.3782	15*	0.3224
	5500	0.2979	110	0.4886	0.96	0.4184	18[3]	0.2948

Table 6.2: Parameter values for tuning simulated annealing parameters. Promising values based on overall performance trends are marked with (*). See Subsection 6.4.1 for details on annotated values.

6.4.2 ALNS Parameters

After fixing the simulated annealing parameters, the ALNS-specific parameters are tuned. These are the degree of destruction DOD and the reaction factor ρ, as explained in Section 5.4 and Section 5.5, respectively. The tested values for DOD are $\{0.05, 0.1, 0.15, 0.2\}$ and the tested values for ρ are $\{0.1, 0.35, 0.7, 0.9\}$. Testing all combinations on two datasets amounts to 32 experiments. The results are displayed in Appendix G.4. Given the limited range of the parameters, a single iteration of parameter refinement is considered sufficient. Preliminary testing showed that a DOD greater than 0.2 leads to a growing request bank size, resulting in inefficient repair phases and ultimately infeasible solutions within the time limit. The results are shown in Appendix G.4. The parameters that most consistently reached the lowest total objective in reasonable time were parameters $(DOD, \rho) = (0.1, 0.35)$. These are, therefore, used during the experiments.

6.4.3 Other Parameters

Other parameters that influence the adaptive weight mechanism of the ALNS are the operator scores $\sigma_1, \sigma_2, \sigma_3$ and the initial operator weights, as discussed in Section 5.5. These scores are awarded to an operator based on its effectiveness. Given the computational cost of tuning the parameters mentioned above, this research adopts the score values proposed by Ropke and Pisinger [54]. These values have been validated in a similar context and showed a strong performance on benchmark instances. The operator scores are fixed at $(\sigma_1, \sigma_2, \sigma_3) = (33, 9, 13)$. Additionally, the initial weights of the destroy and repair operators are set to 1. At the start of the algorithm, each operator has an equal probability of being selected.

6.5 ALNS Solution Performance

This section evaluates whether the ALNS algorithm yields improved routing solutions compared to the initial heuristic generated by the constructive heuristic and the historical routing plan. The comparison is based on the previously discussed components of the objective function. Additionally, differences in other solution characteristics, such as the number of transfers, are also analyzed. The experiments are executed on 25 data instances based on Nijhof Wassink's historical routing plan. Detailed results per data week and aggregated statistics are shown in Appendix H. Figure 6.1 presents the objective values for each solution approach. The total objective value is displayed on the right side of the legend. Figure 6.2 shows the composition of the objective components for each of the routing solutions. Table 6.3 shows a summary of additional solution metrics. Both the figure and table include average values across the data instances. The following subsections analyze each objective component individually.

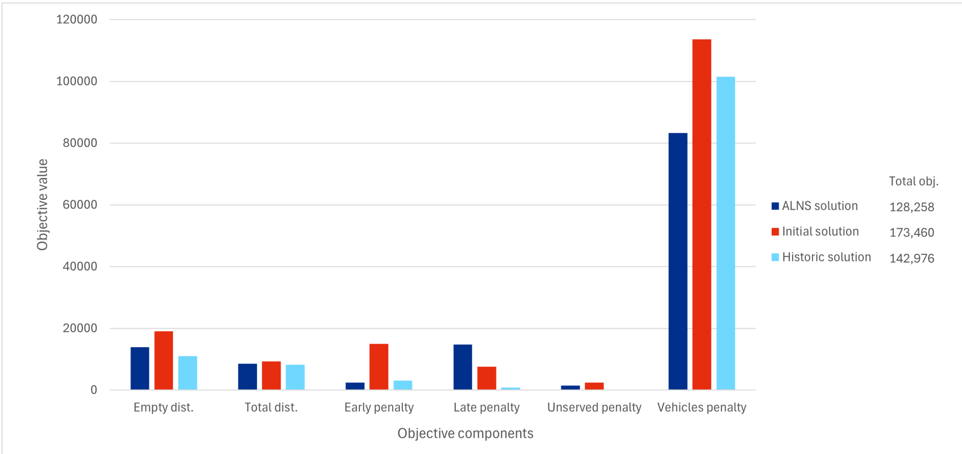


Figure 6.1: Bar chart of objective components historic, initial, and ALNS routing solutions.

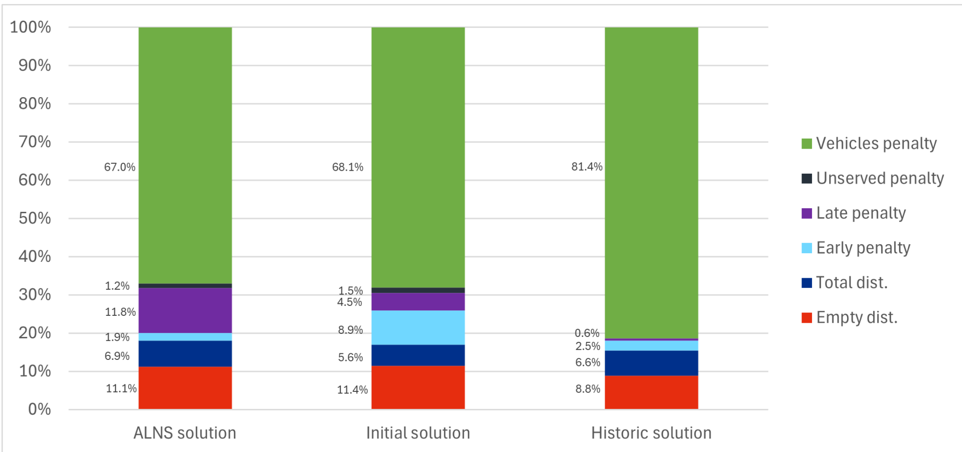


Figure 6.2: Stacked chart of objective components historic, initial, and ALNS routing solutions.

Metric	ALNS	Initial	Historical
Load factor	0.75	0.74	0.73
Nr. vehicles	66.60	90.91	81.20
Unserved req. (%)	2.94	4.91	-
Partial req. (%)	-	4.50	19.82
Nr. transfers	1.48	34.43	53.36

Table 6.3: Other average solution metrics for each solution type.

Empty Distance

The ALNS algorithm reduces the empty distance by approximately 27% compared to the initial solution. However, the historical solution has a better performance for this metric. Specifically, the historical routing plan has 26% fewer empty kilometers than the ALNS solution. Despite this, the ALNS solution can serve 19.82% of the partial requests fully, minus the 2.91% that remain unserved.

Total Distance

Regarding total distance, the ALNS solution achieves an average reduction of 8% compared to the initial solution. Compared to the historical data, it incurs a slight increase of approximately 3%. This increase in total distance aligns with the increased number of fully served requests, as serving requests naturally involves more travel distance. This suggests a trade-off between the served requests and the travel distance.

Load Factor

The load factor, representing the ratio of distance traveled with a load, only improves marginally between the different solutions. Compared to the initial solution, the ALNS solution improves the load factor by 2%. The improvement over the historical solution was only 1%. The average load factor of the ALNS solution is 0.75, with a standard deviation of 1.6%. It consistently outperforms the historical routing plan on this objective component. However, in weeks 5 and 25, the load factor from the initial solution is better than that of the ALNS solution. These lower performances coincide with weeks that have fewer requests. Compared to weeks with higher demand, weeks with fewer requests have a reduced potential for efficient routing solutions.

Vehicle Utilization

The ALNS solution achieves the largest reduction of vehicle utilization penalties. On average, it requires 27% fewer vehicles than the initial solution, with a standard deviation of 11%. This is equivalent to a variation of up to 8 vehicles per week. Compared to the historical solution, ALNS uses around 18% fewer vehicles. The most significant reduction is found in weeks 14, 15, and 20. However, these weeks also show an above-average number of requests being unserved compared to other improved solutions. In particular, these weeks have around 5% of unserved requests, while the average is 3%. Only in week 12 is a large reduction of utilized vehicles observed, while the number of unserved requests is below average in the improved solution. While these patterns suggest a trade-off between vehicle utilization and request fulfillment, the overall efficiency of the solution routes improves. In the ALNS solution, each vehicle serves, on average, 5.44 requests compared to the 4.59 requests in the historical solution. This indicates that the ALNS routes utilize fewer vehicles but also use them more efficiently.

Time Window Penalties

The ALNS solution incurs the highest penalties for arriving late at the pick-up and delivery locations. Conversely, the initial solution is more frequently penalized for arriving too early. This is likely as it starts to plan every request at the start of the week, servicing requests before the time windows open. The historical routing plan incurs minimal time window penalties for early and late arrivals. This indicates that the time windows are aligned with the request constraints. The observed variation suggests that the solution approaches prioritize this metric differently.

Unserved and Partial Requests

The initial solution leaves a total of 4.07% either unserved or partially served. The historical routing plan has, on average, 20% partial requests. In contrast, the ALNS solution has only 3% unserved requests. Unlike the historical and initial routing plans, the ALNS solution does not allow partial servicing during the improvement phase and instead aims to fully satisfy each request. This leads to a higher total fulfilled demand. Although serving more requests fully incurs more total and potentially empty distance, the results show that the load factor was not negatively influenced.

Transferred Requests
The ALNS solution rarely includes transferred requests. The added solution complexity does not improve the objective components sufficiently to include transfers in the best-found solution. In most weeks, fewer than 3 transfers are included, with one outlier in week 19, where 24 requests are transferred. The initial solution requires approximately 35 transfers per data week due to the design of the heuristic, as was explained in Section 5.2. The historical routing plan uses around 53 transfers per week.

Full Solution
To compare the solution as a whole, the total objective of the ALNS solution is approximately 128,258, with a standard deviation of 8%. This shows that the solution is robust when applied to different data instances. A total improvement of 26% compared to the initial solution and 10% compared to the historical solution is achieved. The only exception is in week 22, as the ALNS algorithm was unable to improve upon the historical solution. The ALNS solution has a 7% higher objective value due to a very high penalty for arriving late and using a relatively high number of vehicles. Moreover, the stacked bar chart illustrates the relative importance of each objective component per solution type. It is observed that vehicle penalties dominate the objective value in all three routing solutions, varying between 67% and 81%. This prominence reflects the high weight assigned to vehicle penalties in the objective function. As a result, vehicle utilization plays a key role in improving routing efficiency. Moreover, the penalty for arriving late in the historical routing plan is less than 1%, whereas it constitutes 12% and 5% of the ALNS solution and initial solution, respectively. The empty and total distances have similar contributions in the three routing solutions, with variations of only 2.6% and 1.3%, respectively.

To conclude, the ALNS algorithm effectively improves the routing performance compared to the constructive heuristic and the historical routing plan. Although minimizing empty kilometers was the key priority, the heuristic solution was unable to effectively reduce this compared to the historical plan. The improvement in the load factor was also marginal and did significantly influence the performance of the ALNS algorithm. Instead, the key improvement was primarily due to the reduction of the number of vehicles required. Furthermore, while the improved solution incurs higher time window penalties and more total and empty distances than the historical routing plan, it can serve more requests completely rather than partially.

6.6 Scenario Analysis

This section tests different input parameters to analyze the effect on the routing solution. In addition, Section 6.6 investigates the impact of varying input parameters. These scenario experiments aim to assess the robustness and adaptability of the ALNS solution approach under different conditions.

6.6.1 Empty Kilometers Weight

The main results indicate that the heuristic provides only a limited reduction of empty kilometers compared to the historical and initial solutions. To better understand this behavior, a scenario analysis was conducted in which the weight of the empty kilometers component in the objective function was varied. The aim is to examine how the routing solutions respond to these changes and to explore trade-offs with other objective components. While the baseline experiments use a weight of 0.25, this analysis tests weights $\{0.01, 0.1, 0.4, 0.65, 0.8\}$ on 4 different data weeks. Full numerical results are provided in Appendix I.1. Figure 6.3 presents a combined chart illustrating the behavior of the objective components across the tested weights. The absolute, unweighted number of empty kilometers and the vehicle penalty are shown as line plots with the right-hand axis. All other objective components are plotted using the left-hand y-axis. The load factor, representing the ratio of distance traveled with a load, is included at the bottom of the figure for each weight.

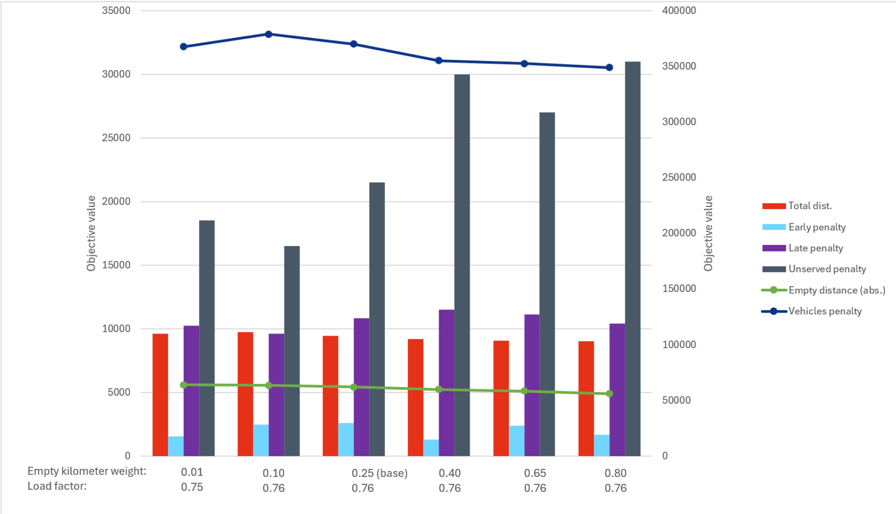


Figure 6.3: Combined bar and line chart with different empty kilometer weights.

Results show a consistent decrease of the number of empty kilometers as their objective weight increases. This confirms that the heuristic correctly prioritizes empty kilometers as their importance increases. Together with the decrease of empty kilometers, the total travel distance also decreases as the weight increases. This indicates that the heuristic does not reduce empty kilometers alone but shortens the overall routes. As a result, the load factor remains relatively stable across the scenarios. A notable side effect is the increasing number of unserved requests as the weight increases. This behavior implies that the heuristic achieves lower total and empty distance by omitting more requests and suggests a trade-off between service level and operational efficiency. Similarly, the vehicle penalty decreases with increasing weight. This can also be a consequence of the higher proportion of unserved requests, as fewer vehicles are required if fewer requests are served. Alternatively, the heuristic may be able to create more efficient and compact routes. Time window penalties do not show a clear trend across the weight configurations, thus suggesting that they are either robust to changes in the empty kilometer weight or not strongly prioritized relative to the other objective components.

6.6.2 Vehicle Utilization

The main results showed that vehicle utilization has a significant impact on the solution quality. Therefore, these experiments test the impact on the solution quality with different weight assignments on the use of vehicles. While the baseline experiments use a vehicle weight of 1250 (Section 6.3), these experiments test the vehicle weights $\{100, 750, 1750, 2400\}$ across 4 data weeks, totaling 20 experiments. The full results and a summary are presented in Appendix I.2. Figure 6.4 displays a bar chart with objective components for each vehicle weight. In addition, the absolute number of utilized vehicles is plotted as a line graph using the secondary axis on the right.

The figure shows a clear trend in which the number of utilized vehicles decreases as the weight of the vehicle increases. By assigning a higher vehicle penalty weight, the model becomes more cautious in deploying vehicles. For the highest vehicle weight, the solution requires 4 fewer vehicles compared to the baseline. However, this reduction also causes the number of unfulfilled requests to increase. With fewer requests being served, both empty and total distances are also reduced. In contrast, the lowest weight results in the use of, on average, 6 more vehicles and a higher number of served requests. It also has a slight increase in empty distance and total distance, by 1% and 3%, respectively. These findings suggest that a higher service level may lead to routing inefficiencies. An analysis of vehicle waiting or idle time would provide further insight into potential routing inefficiencies when the vehicle weight is low. This is currently not accounted for in the objective function. However, this metric was not available after the experiments were completed.

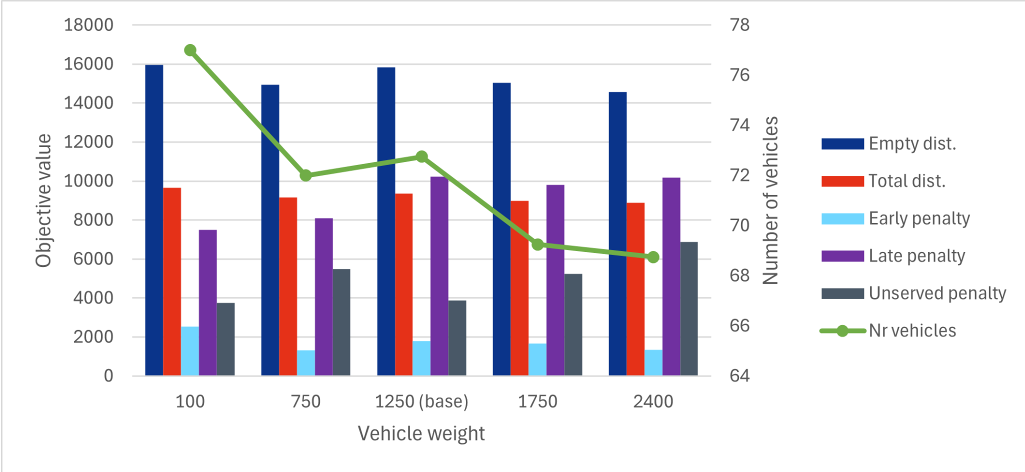


Figure 6.4: Bar chart of objectives with different vehicle weights and the line representing the number of vehicles.

6.6.3 Time Windows

To evaluate the sensitivity of the routing solution to time window constraints, 4 test cases are designed, and each is tested on the same 5 data weeks. This amounts to 20 experiments. The first scenario categorizes all locations as category A, meaning there is a little time window penalty for arriving early or late at any of the locations. The second scenario only serves locations as category B, where arriving before the time window opens incurs small penalties and, after a window closes, a heavy penalty. In the third scenario, all locations incur substantial penalties for arriving early and late. The fourth case is a mixed distribution of the time window categories. The ratios in which the locations are categorized are as follows: $(A, B, C) = (0.42, 0.18, 0.4)$, whereas the distribution in the baseline experiments is $(A, B, C) = (0.8, 0.01, 0.1)$. The full results and the aggregated data across the weeks are shown in Appendix I.3. Figure 6.5 illustrates a bar chart of the average values for the objective components per time window type, and the average total objective is shown on the legend's right.

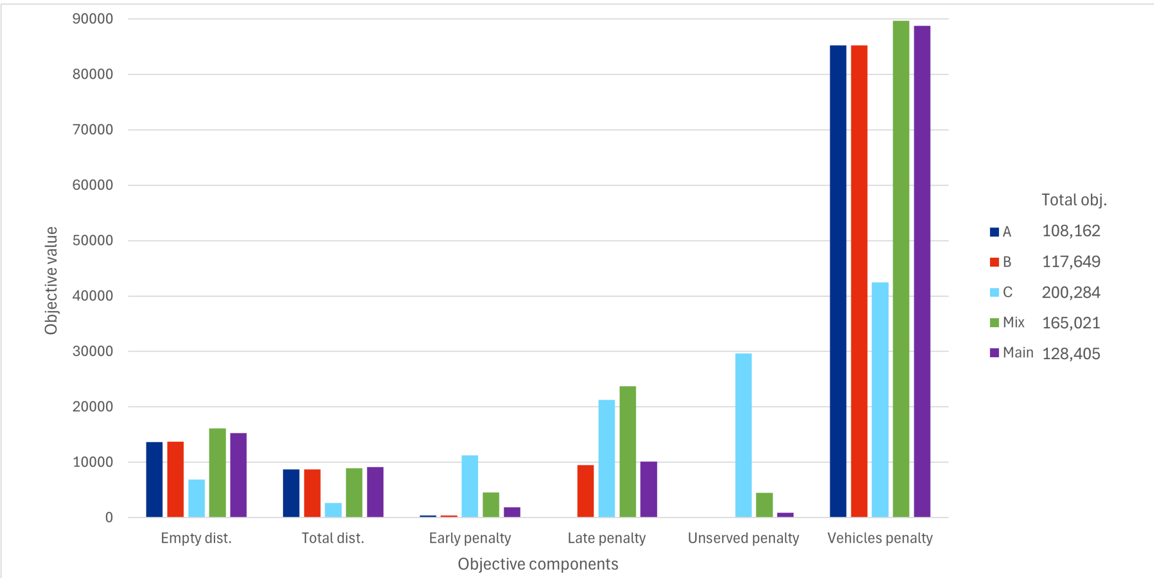


Figure 6.5: Objective components per time window category.

The routing solution with only category A locations yields the lowest average objective compared to the baseline experiments. It improves the baseline experiments by approximately 15.8% with an average of 108,162 and a standard deviation of 5.5%. This improvement is primarily due to a decrease of late arrival time window penalties. It also shows a lower percentage of unserved requests. Solutions with category B improve the objective of the baseline experiments of 8.4%. Also, the performance of the load factor is, on average, 0.03 higher and reaches the solution around 200 seconds faster than in the baseline experiments. Category C has a high penalty for not serving requests. The data shows that it only fulfills 20 to 65% of the requests. This leads to reduced vehicle penalties for category C. However, as the other components are also relatively high, the total average objective of routing solutions with only strict customers are the highest. Additionally, 4 out of 5 experiments were terminated prematurely due to the time limit. This indicates that the algorithm has difficulties finding repair mechanisms that can feasibly reconstruct the routes within the hard time window constraints. The mixed distribution, on average, has a higher objective value of 28% compared to the baseline results. This is largely caused by early and late time penalties, as well as the increased number of unserved requests. Notably, these experiments were also cut short due to the time limit.

To conclude, soft time window constraints provide more flexibility to the routing solution. Therefore, the algorithm can effectively optimize the routes and leave fewer unserved requests. More constrained time windows lead to worse objective values due to the increased time window penalties. Additionally, they have longer run times, as the algorithm has more difficulties in finding feasible routing solutions.

6.6.4 Requests Received on Time

From the historical data, it was observed that some of the requests were received late, which is defined as two days before the opening of their pick-up time window. The aim is to investigate whether these late requests have complemented the historical routing plan or have worsened the solution quality. Moreover, it also demonstrates how sensitive each objective component is to change. To test this, the 25 experiments as in Section 6.5 are executed again but with a filtered set of open request data, excluding the considered late requests. On average, 7.5% of requests per week were filtered. This percentage deviates from the 9% stated in Chapter 2 because outsourced requests were excluded from the filtered dataset used as model input. Figures 6.6 and 6.7 present box plots of the objective components. For a fair comparison, each metric is normalized by dividing by the total number of requests available in that week, including unserved requests. The y-axis represents the deviation from the scenario experiments compared to the baseline experiments. Due to their different scales on the y-axis, they are plotted in separate figures. Appendix I.4 provides the full results.

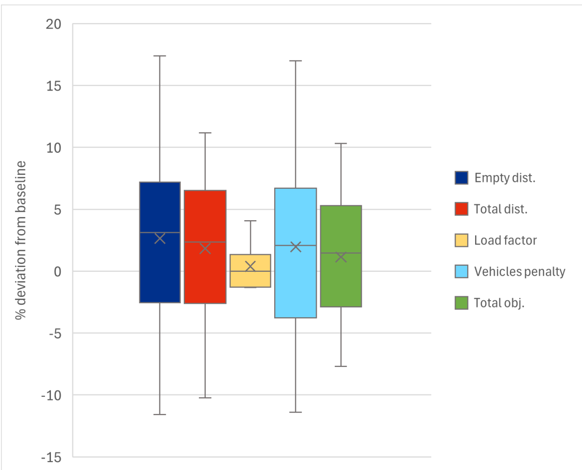


Figure 6.6: Box plot of several objective components deviations relative to the baseline experiments.

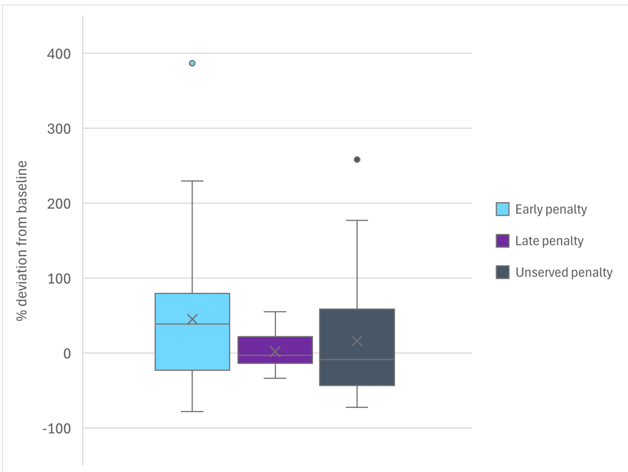


Figure 6.7: Box plot of several objective components deviations relative to the baseline experiments.

The results show that empty and total distances have a slight positive deviation when late requests are excluded. This suggests that removing late requests does not consistently improve the performance of these metrics. The deviation of both the metrics is in a similar range for each data week due to the correlation between the metrics. The load factor's deviation remains stable around 0, indicating no significant change to this metric. The vehicle penalty deviation ranges from -11% to 17%. This implies that the negative deviation does not proportionally decrease the number of vehicles with fewer requests. Weeks 25 and 30 have a higher vehicle penalty than the baseline experiments, as the solution has a lower rate of unserved requests compared to the baseline experiments. This could happen if the algorithm had a more efficient method of incorporating requests that were left unserved in the baseline experiments. The total objective shows a balanced distribution of deviations, with an average value slightly above 0 and positive and negative deviations of up to approximately ± 10%. This implies that removing all late requests slightly degrades the routing performance on average. However, due to the positive and negative fluctuations, it does not improve or decrease performance consistently. The second figure illustrates the early and late time window penalties, as well as the unserved penalty. These components are far more unstable than other components. The early time window penalty increases by an average of 45% in the scenario experiments, meaning that the scenario experiments arrive earlier at requests than the baseline experiments. This suggests that the algorithm tends to shift arrival times earlier when there are fewer requests present. The average of late penalties and unserved requests penalties is near 0, but they display large fluctuations. Therefore, it is difficult to determine a trend. Across all metrics, there is no clear relationship between the percentage of removed requests and the deviation from the baseline experiments. However, time window penalties and the unserved penalty are more sensitive to changes in the input requests than the empty and total distance and vehicle penalty metrics.

6.7 Conclusion

This chapter presents the methodology for finding the optimal parameter settings for the ALNS algorithm, tailored to Nijhof Wassinks routing data. Using these settings, the algorithm's performance is compared to that of the constructive heuristic and the historical routing plan. The ALNS algorithm improves the objective function by 26% compared to the initial solution and 10% compared to the historical solution. These improvements were mainly caused by the more efficient use of vehicles rather than improvements in load factor or (empty) distance. Using these experiments as a baseline, a scenario analysis is conducted to evaluate the effect of varying vehicle use weights, time window constraints, and the exclusion of late requests. The insights from these analyses form the basis for the conclusions and recommendations in Chapter 7.

7| Conclusion, Contributions & Recommendations

This chapter elaborates on the conclusions that can be drawn from the results from the previous chapter and answers the main research question. Section 7.1 concludes the findings of this paper. This is followed by the academic and practical contributions in Section 7.2. Section 7.3 details the limitations of the research. After which, Section 7.2 describes the recommendations to Nijhof Wassink. Lastly, Section 7.5 concludes with directions for future research.

7.1 Conclusions

The main research question this study aims to answer is: “How can the Dry Bulk Logistics planning of Nijhof Wassink be optimized in order to reduce empty kilometers?”. Following the research framework from Section 1.3, the remainder of this section will formulate an answer to this question.

The analysis of the current planning strategy revealed a reliance on manual planning strategies and the absence of a routing planning optimization tool. Given the complex request, driver, and vehicle constraints, minimizing empty kilometers manually is highly challenging.

A review of the academic literature helped formalize Nijhof Wassink’s planning problem as a new VRP variant: the multi-depot, pick-up, and delivery with time windows and transfers (MD-PDPTW-T) problem. Existing models addressed parts of this formulation but not in its full complexity. While a mathematical model was formulated to represent the planning problem and its logic, solving it exactly was proven to be infeasible for realistic problem sizes.

As a result, an Adaptive Large Neighborhood Search heuristic was designed to solve real-scale instances. Although minimizing empty kilometers is a priority, minimizing this alone can lead to unrealistic and impractical routing solutions. For example, it can lead to solutions with excessive numbers of transfers or unserved requests. To counter this, additional objectives, such as vehicle usage and total distance, were included to promote realistic routing behavior. The flexibility of leaving requests unserved was necessary to provide the heuristic with sufficient flexibility to explore the solution space.

The applied ALNS heuristic consistently produced more efficient routing plans compared to both the initial solution from a constructive heuristic and historical routing plan. However, improvements in empty kilometers were limited, and the load factor, representing the fraction of distance with a load, remained stable. Moreover, the emissions, which are directly related to the empty and total distance, have also not shown significant improvement. Notably, the ALNS achieved a significant reduction in the number of vehicles used, indicating that operational costs can be saved by shifting the focus from empty kilometers to improved vehicle utilization.

Scenario analysis further confirmed the trade-offs between objective components. Increasing the relevance of empty kilometers reduces empty and total distance but also increases the number of unserved requests. Similarly, reducing the penalty on vehicle usage increased service level but worsened routing efficiency. These outcomes highlight the importance of balancing the competing objective components.

In conclusion, the findings of this study show that Nijhof Wassink’s vehicle routing can be effectively optimized using a heuristic model. While the reduction of empty kilometers is limited under the proposed model assumptions, significant improvement can be achieved through better vehicle utilization. The findings also highlight a strong interdependence between the objective components. Therefore, it is essential to strategically prioritize between the components to ensure routing solutions align with operational priorities.

7.2 Contributions

7.2.1 Practical Contributions

This study offers several practical contributions. Firstly, it provides an extensive context analysis of the planning department, which offers insights into the current process and the underlying causes of planning inefficiencies. Moreover, the ALNS algorithm demonstrates that creating a concept routing plan with comparable or improved performance to the historical routing plan is possible. Notably, the algorithm showed that the main KPI, empty kilometers, had limited potential for improvement under this model's assumptions. It suggests that another KPI, vehicle utilization, offers more improvement potential and should be prioritized to optimize routing efficiency. In addition, the proposed algorithm provides a standardized and faster approach for creating a concept routing plan with improved or comparable performance. This reduces dependence on current resources and the time-intensive manual process. In addition, it supports a more consistent and scalable planning approach. For Nijhof Wassink, it may serve as a first step toward integrating optimization tools into the planning process and, therefore, supporting more informed and efficient decision-making.

7.2.2 Academic Contributions

This paper contributes to the existing VRP literature by formulating a new variant. In particular, this paper introduces the multi-depot, pick-up, and delivery problem with time windows and transfers (MD-PDPTW-T). While individual routing characteristics, such as pick-up and deliveries, time windows, and transfers, have been studied in prior studies, to the best of my knowledge, no study has combined all of them into a single model. The mathematical model extends the mathematical notation proposed by [57] and [50]. It adds a mix of soft and hard time windows, as well as the break requirement for vehicles. Therefore, this model is more adapted to realistic operational planning. However, due to the computational challenges of the exact model on a realistic scale, the study advances with a meta-heuristic approach, namely the ALNS algorithm. Although this algorithm is a well-founded improvement heuristic for all types of VRP problems, this paper contributes to the literature by applying an ALNS algorithm with transfers to more locations and with a longer planning horizon than previously researched. Prior studies have tested the ALNS algorithm with transfers up to 200 locations and with a planning horizon of 9 hours. In contrast, this study uses 670 unique locations and a planning horizon of 5 days.

7.3 Limitations

While the presented model demonstrates a promising routing performance, this study relies on several limitations that impact the research's practical applicability, computational efficiency, and robustness.

The current model does not include several practical constraints in the planning process. For example, the model does not account for partial request fulfillment, varying driver shift lengths, actual travel times and distances, and detailed driver rest regulations. These factors can influence route feasibility, and without them, routing solutions may be impractical or overly optimistic. Moreover, a modeling decision was made to allow unserved requests. Without this flexibility, many solutions were infeasible, which severely limited the heuristic's ability to explore the solution space. Although leaving requests unserved by outsourcing is common in operational practice, it complicates a direct comparison between the historical and ALNS solutions.

Furthermore, the algorithm is limited by the computation time of the heuristic for two main reasons. Firstly, each iteration has a duration of a few seconds, depending on the selected operators and the number of transfers. To achieve quality solutions on a real problem scale, many iterations are required. Second, due to the synchronization constraint of the transfers, request insertions must be sequentially assessed. Specifically, if a vehicle route that, at some point in time, also participates in transferring a request is altered, the complementing vehicle route must be updated simultaneously. This is necessary to avoid a mismatch of requests at the transfer station. The slow search process can limit the model's scalability and practical applicability. To manage the computation complexity, the current repair is limited to evaluating a set of promising insertion positions. This approach improves runtime but risks overlooking potential insertion positions and limits the heuristic to exploit the entire solution space.

Another limitation is the parameter tuning. Due to the large number of configurable options and the computational complexity, not all algorithmic parameters were tuned as extensively. As a result, the current parameter configuration may not reflect the optimal settings for the heuristic. Moreover, the weights used in the objective function were tuned empirically to balance trade-offs between the objective components. While this resulted in efficient solutions, the scenario analysis showed that small changes in vehicle weight led to notable changes in request fulfillment, total, and empty distance. This suggests that the model is sensitive to parameter settings, which may compromise the robustness of the heuristic and the consistency of its solutions.

Lastly, the results showed that very few requests were transferred in the routing solutions. This suggests that either the transfer-based operators cannot function efficiently or transfers are not beneficial to the routing solutions. However, the cause of the limitation has not been tested and remains unclear.

7.4 Recommendations

Based on the findings in this study, several recommendations are proposed for Nijhof Wassink to enhance routing efficiency and support the transition towards more data-driven planning.

The key finding of the study concerns the limited reduction of empty kilometers. Although the model improved overall routing efficiency, the reduction of empty kilometers in the improved solutions was limited, and the load factor remained stable. This indicates that, under current assumptions, the model was unable to find a better assignment of requests to vehicles. Instead, the historic solutions were improved due to a reduction in vehicle usage. In other words, the model solution can serve the same number of requests with fewer vehicles. This indicates a potential underutilization of vehicles in the historical routing plans. It is, therefore, recommended that further analysis be conducted to verify this. If so, Nijhof Wassink may have the opportunity to take on an additional workload without needing more resources, ultimately increasing operational efficiency. Due to the potential improvement of vehicle usage, it is advisable to shift the focus from reducing empty kilometers to more efficiently utilizing vehicles.

If reducing empty kilometers remains a priority, further analysis is recommended on the impact of outsourcing strategies. The scenario analysis indicated that minimizing empty kilometers may result in a higher number of unserved requests. Since outsourcing was excluded from the model's scope, it aims to serve as many requests as possible without specifically optimizing requests that could be handled more efficiently if outsourced. Further research could examine the effects of selectively outsourcing certain requests on empty kilometers and service levels.

This study successfully presented a model that creates concept routing plans based on the available resources and requests at the start of the week. However, it is not yet suitable for operational use for two main reasons. First, the model cannot yet account for all practical constraints, such as varying driver shift lengths and cleaning requirements. Second, the current planning process is not yet mature enough to support this level of automation. Currently, no system is in place to support planners during the decision-making process of the planning. To bridge this gap, planners need time to be gradually introduced to data-driven planning decisions. Current initiatives, such as cleaning suggestions and next-location advice, are promising starting points. In parallel, the ALNS model should be expanded to incorporate more operational constraints. Ultimately, the aim is to integrate both the ALNS model and tools into a unified system that supports the full planning process. In the future, it may serve as a robust concept planning tool and can potentially even support real-time planning decisions.

At this stage, the ALNS algorithm is best suited to support tactical and strategic decision-making without interfering with the operational planning process. On this level, it may contribute in two ways. Firstly, it can support the post-analysis of historical data. In this case, the model's assumption that the requests are known at the start of the week is already satisfied. Examples could be the customer's cost-effectiveness or impact on the network density. Second, it may support forward-looking decisions. It can, for example, support the tender process. Currently, the tender decisions are based on historical data and the current workload. However, if historical data is outdated or missing, there is no tool to simulate the effect on the transport network. The algorithm can fill this gap if given an average weekly workload. Additionally, it can support strategic decisions such as upscaling or downsizing the number of vehicles or evaluating the impact of each depot location. These contributions can provide a robust and data-driven basis for decision-making. For these applications, it is essential that the algorithm roughly mimics the actual planning behavior. In particular, an assessment is recommended to understand why the model consistently uses fewer vehicles than the current manual routing plans.

Finally, some pointers for the operational planning can be given based on the scenario analysis. It showed that flexible time windows significantly improve routing flexibility and efficiency. However, customers are increasingly demanding hard time windows, which can pose challenges to the routing flexibility of Nijhof Wassink. It is advisable to delay the booking of time slots where possible or, if required by the customer, make agreements carefully to ensure feasibility. Next to the time windows, no consistent relationship is found between declining late requests and the overall solution quality. It is advised to analyze these late requests case-by-case and only accept them if they can improve the current routing plan without increasing empty kilometers or causing time violations.

7.5 Future Research

Although the proposed model provides a foundation for solving vehicle routing problems, several opportunities for future research exist to enhance the model's performance and operational applicability.

Future work could explore the use of different programming software for faster performance, such as C++. Or it could develop methods that support parallel computation while ensuring the synchronization of requests during transfers. Faster computation improves both computational efficiency and scalability and allows the repair phase to explore a broader solution space.

Further improvements can focus on systemic parameter tuning. Methods such as grid search or meta-optimization could be used to refine the weights in the multi-criteria objective and other algorithm parameters. In addition, a more detailed statistical analysis of the solution quality and runtime behavior can provide deeper insight into the model's effectiveness and limitations. For example, analyzing vehicle idle time may reveal relevant routing insights and highlight areas for improvement.

Given the limited use of transfers in the ALNS solutions, future work could investigate the effectiveness of the transfer mechanisms. This may include refinement of the existing operators or designing new ones.

To increase the model's applicability in real-world operations, several modeling extensions are proposed. Allowing partial request fulfillment could further improve vehicle utilization and empty kilometers. Supporting dynamic changes, such as new or canceled requests, and relaxing the requirement for vehicles to return to their depot can make the model more applicable for re-optimization during the week.

References

[1] Baldacci, R., Bartolini, E., Mingozzi, A., and Roberti, R. (2010). An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, 7(3):229–268.

[2] Bent, R. and Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research*, 33(4):875–893.

[3] Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31.

[4] Boonsam, P., Suthikarnnarunai, N., and Chitphaiboon, W. (2011). Assignment Problem and Vehicle Routing Problem for an Improvement of Cash Distribution. *Engineering and Computer Science*, page 1185.

[5] Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering*, 99(1):300–313.

[6] Cattaruzza, D., Absi, N., and Feillet, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2):676–693.

[7] Cattrysse, D. G. and Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60:260–272.

[8] Chiang, W.-C. and Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Operations Research*, 63:3–27.

[9] Chu, J. C., Yan, S., and Huang, H. J. (2017). A Multi-Trip Split-Delivery Vehicle Routing Problem with Time Windows for Inventory Replenishment Under Stochastic Travel Times. *Networks and Spatial Economics*, 17(1):41–68.

[10] Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *Operational Research Society*, 52(8):928–936.

[11] Cortés, C. E., Matamala, M., and Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724.

[12] Cortes, J. D. and Suzuki, Y. (2020). Vehicle Routing with Shipment Consolidation. *International Journal of Production Economics*, 227.

[13] Crevier, B., Cordeau, J. F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773.

[14] Danloup, N., Allaoui, H., and Goncalves, G. (2018). A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. *Computers and Operations Research*, 100:155–171.

[15] Dantzig, G. B. and Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1):80–91.

[16] Del Pia, A. and Filippi, C. (2006). A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2):125–141.

[17] Derigs, U., Pullmann, M., and Vogel, U. (2013). Truck and trailer routing - Problems, heuristics and computational experience. *Computers and Operations Research*, 40(2):536–546.

[18] Drexl, M. (2012). Rich vehicle routing in theory and practice. *Logistics Research*, 5(1-2):47–63.

[19] Ehrler, V., Lewis, A., Schön, A., and https://smart-freight-centre-media.s3.amazonaws.com/documents/GLEC_FRAMEWORK_v3_23_10_24_B.pdf (2024). Global Logistics Emissions Council Framework.

[20] El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, 22(3):123–131.

[21] Friedrich, C. and Elbert, R. (2022). Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics. *Computers and Operations Research*, 137.

[22] Gendreau, M. and Potvin, J.-Y. (2005). Chapter 6 TABU SEARCH. *Search Methodologies, https://doi.org/10.1007/0-387-28356-0_6*.

[23] Gerdessen, J. C. (1996). Vehicle routing problem with trailers. *European Journal of Operational Research*, 93:135–147.

[24] Glover, F. (1986). Future Paths For Integer Programming And Links To Artificial Intelligence. *Computer & Operation Research*, 13(5):533–549.

[25] Ho, W., Ho, G. T., Ji, P., and Lau, H. C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557.

[26] Hoff, A. and Løkketangen, A. (2008). *A Tabu Seach Approach For Milk Collection In Western Norway Using Trucks And Trailers*. PhD thesis, Molde University College.

[27] Huang, N., Qin, H., Du, Y., and Wang, L. (2024). An exact algorithm for the multi-trip vehicle routing problem with time windows and multi-skilled manpower. *European Journal of Operational Research*, 319(1):31–49.

[28] Kammarti, R., Borne, P., and Ksouri, M. (2007). Hybrid Evolutionary Approaches and New Benchmarks for the 1-PDPTW. In *Management and Control of Production and Logistics*, pages 27–30.

[29] Khoo, T. S. and Bonab, M. B. (2022). Solving multi-objective pickup and delivery with time windows using mediocre evolutionary distributed microservices re-optimization algorithm. *Applied Soft Computing*, 125.

[30] Koç, . and Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives.

[31] Koch, H., Bortfeldt, A., and Wäscher, G. (2018). A hybrid algorithm for the vehicle routing problem with backhauls, time windows and three-dimensional loading constraints. *OR Spectrum*, 40(4):1029–1075.

[32] Konstantakopoulos, G. D., Gayialis, S. P., and Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research*, 22(3):2033–2062.

[33] Kuo, Y. and Wang, C. C. (2012). A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications*, 39(8):6949–6954.

[34] Laporte, G. (2009). Fifty Years of Vehicle Routing. *Transportation Science*, 43(4):408–416.

[35] Lehmann, J. and Winkenbach, M. (2024). A matheuristic for the Two-Echelon Multi-Trip Vehicle Routing Problem with mixed pickup and delivery demand and time windows. *Transportation Research Part C: Emerging Technologies*, 160.

[36] Lin, S. W., Yu, V. F., and Lu, C. C. (2011). A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, 38(12):15244–15252.

[37] Liu, R., Jiang, Z., Fung, R. Y., Chen, F., and Liu, X. (2010). Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Computers and Operations Research*, 37(5):950–959.

[38] Lyu, Z. and Yu, A. J. (2023). The pickup and delivery problem with transshipments: Critical review of two existing models and a new formulation. *European Journal of Operational Research*, 305(1):260–270.

[39] Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.

[40] Mingozzi, A. and Baldacci, R. (1999). Exact method for the vehicle routing problem with backhauls. *Transportation Science*, 33(3):315–329.

[41] Montané, F. A. T. and Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research*, 33(3):595–619.

[42] Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots.

[43] Nanry, W. P. and Barnes, J. W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research*, pages 107–121.

[44] Nijhof, F. (2024). About Nijhof-Wassink.

[45] Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. *Journal fur Betriebswirtschaft*, 58(2):81–117.

[46] Pemberthy R, J. I., Correa Espinal, A. A., and Muriel Villegas, J. E. (2019). A cross-border, long haul freight transportation problem with transhipments. *International Journal of Logistics Systems and Management*, 32(3/4):437.

[47] Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2019). A Generic Exact Solver for Vehicle Routing and Related Problems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11480 LNCS, pages 354–369. Springer Verlag.

[48] Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435.

[49] Qureshi, A. G., Taniguchi, E., and Yamada, T. (2009). An exact solution approach for vehicle routing and scheduling problems with soft time windows. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):960–977.

[50] Rais, A., Alvelos, F., and Carvalho, M. S. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.

[51] Ralphs, T., Kopman, L., Pulleyblank, W., and Trotter, L. (2003). On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2-3):343–359.

[52] Rijksoverheid (2024). Wat is er wettelijk geregeld voor de rijtijden en rusttijden bij wegvervoer?

[53] Ropke, S. and Cordeau, J.-F. (2009). Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 43(3):267–286.

[54] 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.

[55] Rutenbar, R. A. (1989). Simulated Annealing Algorithms: An Overview. *Circuits and Devices Magazine*, 5(1):19–26.

[56] Salimifard, K. and Bigharaz, S. (2022). The multicommodity network flow problem: state of the art classification, applications, and solution methods. *Operational Research*, 22(1).

[57] Sampaio, A., Savelsbergh, M., Veelenturf, L. P., and Van Woensel, T. (2020). Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. *Networks*, 76(2):232–255.

[58] Semet, F. and Taillard, E. (1993). Solving real-life vehicle routing problems efficiently using tabu search. *Operations Research*, 41:469–488.

[59] Tang, J., Pan, Z., Fung, R. Y., and Lau, H. (2009). Vehicle routing problem with fuzzy time windows. *Fuzzy Sets and Systems*, 160(5):683–695.

[60] Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers and Operations Research*, 40(1):475–489.

[61] Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2):231–244.

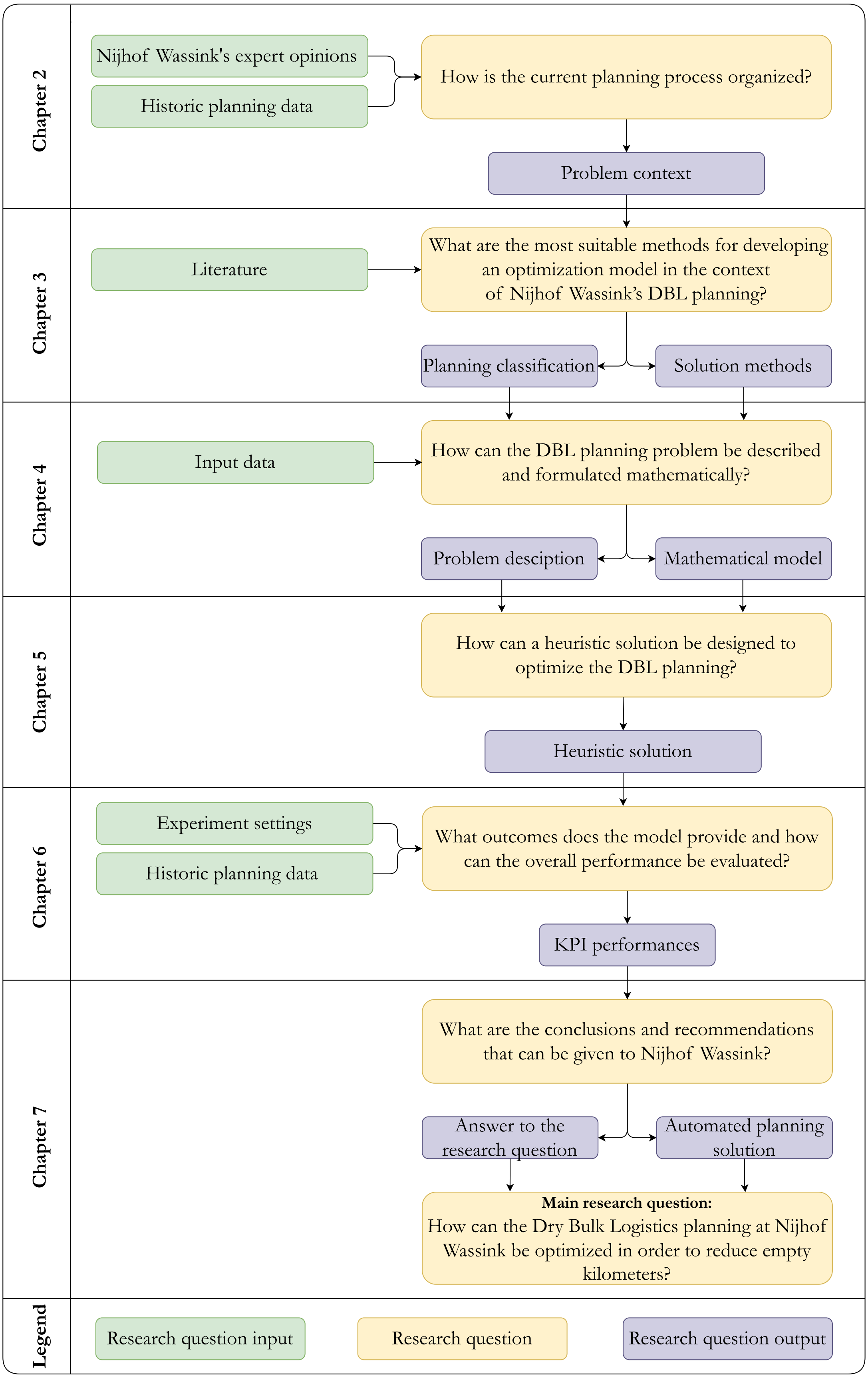
[62] Wang, C., Mu, D., Zhao, F., and Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Computers and Industrial Engineering*, 83:111–122.

[63] Xiong, H., Shi, S., Ren, D., and Hu, J. (2022). A survey of job shop scheduling problem: The types and models. *Computers and Operations Research*, 142.

[64] Zhang, H., Ge, H., Yang, J., and Tong, Y. (2022). Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms. *Archives of Computational Methods in Engineering*, 29(1):195–221.

Appendices

A Research Design



B Planning Process Flowchart

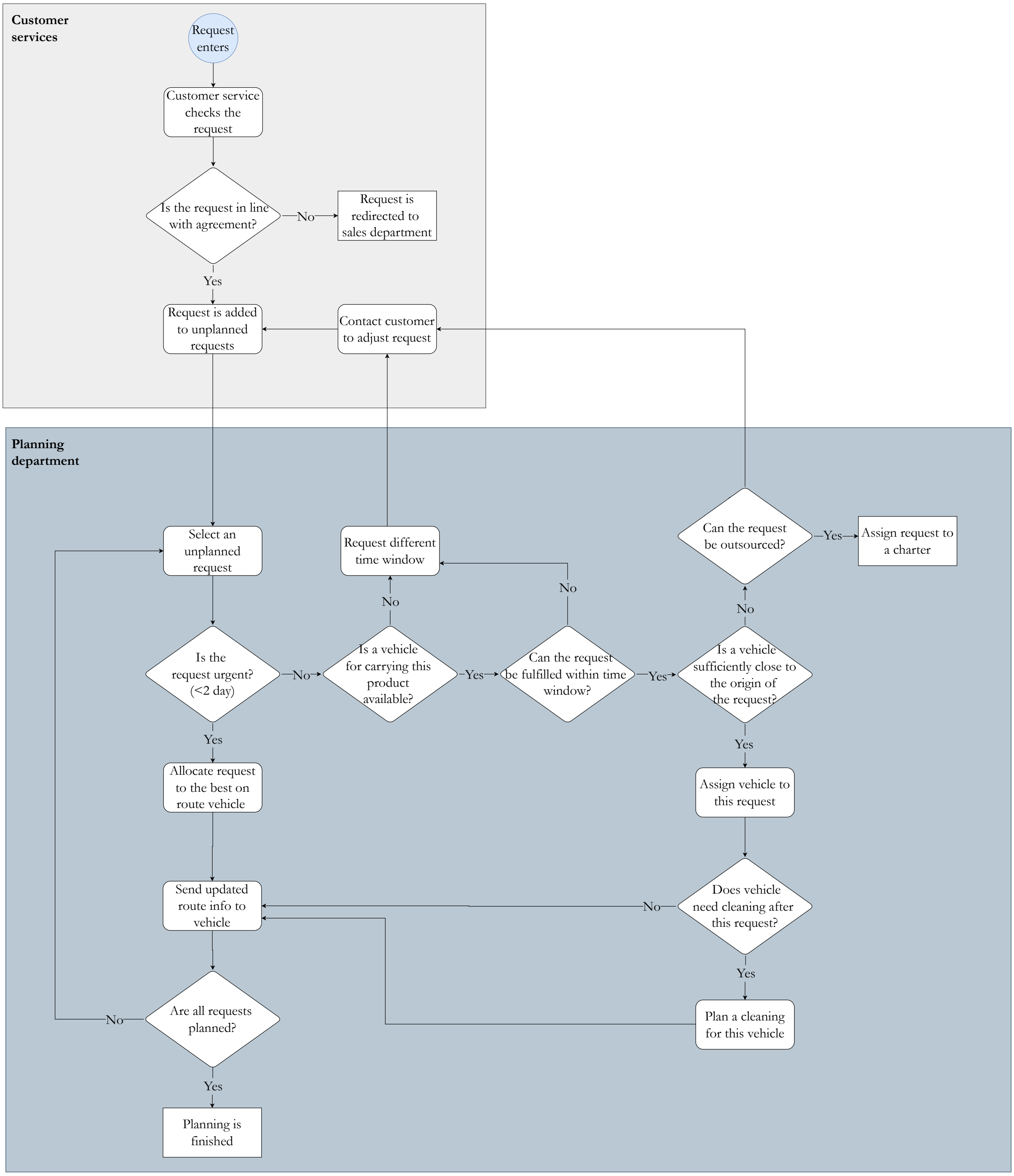


Figure 2: Current planning process in a flowchart

C Literature Review Overview

Article	VRP variant								Solution method
	TW	PDP	MD	TTP	MT	B	T	Other	
Baldacci et al., 2010 [1]	✓	✓	✓						Column-and-cut generation
Bent & Hentenreyk, 2006 [2]	✓	✓							Two-phase with simulated annealing and large neighborhood search
Cattaruzza et al., 2016 [6]	✓				✓			Release dates	Hybrid genetic algorithm
Chiang and Russel, 1996 [8]	✓								Simulated annealing
Chu et al., 2017 [9]					✓			Split delivery, stochastic travel time	Simulation-based method
Cordeau et al., 2001 [10]	✓		✓						Tabu search
○ Cortés et al., 2010 [11]	✓	✓	✓				✓		Branch-and-cut
Cortes & Suzuki, 2020 [12]		✓					✓	Split delivery	Two-stage simulated annealing
Crevier et al., 2007 [13]			✓				✓		Three-phase with tabu search, adaptive memory and integer programming
○ Danloup et al, 2018 [14]	✓	✓	✓				✓		Large neighborhood search and genetic algorithm (separate)
Derigs et al. 2013 [17]	✓			✓					Two-phase with local search and large neighborhood search
Friedrich & Elbert, 2022 [21]	✓						✓		Adaptive large neighborhood search
Gerdessen, 1996 [23]				✓					-
Ho et al., 2008 [25]			✓						Genetic algorithm
Huang et al., 2024 [27]	✓				✓				Branch-and-price-and-cut
Koch et al., 2018						✓		Loading constraints	Adaptive large neighborhood search
Lehmann & Winkenbach, 2024 [35]	✓				✓		✓	Two-echelon	Adaptive large neighborhood search with exact MILP
Lin et al., 2011 [36]	✓			✓					Simulated annealing
Liu, 2010 [37]		✓	✓						Two-phase greedy algorithm
○ Lyu and Yu, 2023 [38]	✓	✓	✓				✓		Mixed integer linear programming
• Masson et al., 2013 [39]	✓	✓	✓				✓		Adaptive large neighborhood search
Mingozorri et al., 1999						✓			MILP with variable reduction algorithm
Montané and Galvão, 2006 [41]		✓						Simultaneous PDP	Tabu search
Nanry & Barnes, 2000 [43]	✓	✓							Tabu search
Pessoa et al., 2019 [47]	✓	✓	✓						Branch-and-cut-and-price
Del Pia & Filippi, 2006 [16]				✓					Variable neighborhood descent
Pisinger & Ropke, 2007 [48]	✓		✓						Adaptive large neighborhood search
○ Rais et al., 2014 [50]	✓	✓	✓				✓		MILP
Ropke & Cordeau, 2009 [53]	✓	✓							Branch-and-cut-and-price
• Sampaio et al., 2020 [57]	✓	✓	✓				✓		Adaptive large neighborhood search
Semet & Taillard, 1993 [58]				✓					Tabu search
Vidal et al., 2013 [60]	✓		✓	✓					Hybrid genetic algorithm
Villegas et al., 2013 [61]	✓			✓					Large neighborhood search
Pemberthy et al., 2019 [46]	✓						✓		Two-phase with GRASP and ILS
Wang et al., 2015 [62]	✓	✓						Simultaneous PDP	Parallel simulated annealing
• This research	✓	✓	✓				✓		Adaptive large neighborhood search

D Additional Information Illustrative Example

This table includes additional defined parameters for the illustrative example in Section 4.1. These can be used to find the optimal solution.

Node from	Time window	Service time	Node to							
			O	P1	P2	P3	D1	D2	D3	T
O	-	-	-	2.00	4.12	2.24	6.32	2.24	4.12	6.00
P1	[1,3]	2	2.00	-	2.24	1.00	4.47	1.00	2.24	4.00
P2	[10,12]	2	4.12	2.24	-	2.83	3.61	2.00	2.00	2.24
P3	[0,5]	2	2.24	1.00	2.83	-	4.12	2.00	2.00	4.12
D1	[16,20]	1	6.32	4.47	3.61	4.12	-	5.00	2.24	2.00
D2	[16,19]	1	2.24	1.00	2.00	2.00	5.00	-	2.83	4.12
D3	[5,10]	1	4.12	2.24	2.00	2.00	2.24	2.83	-	2.24
T	-	-	6.00	4.00	2.24	4.12	2.00	4.12	2.24	-

E Pseudocode Initial Solution

The initial solution is constructed by assigning the set of requests, denoted as Requests. Each request $r \in$ Requests is defined by a pick-up location $p(r)$ and a delivery location $d(r)$. Furthermore, a set of vehicle routes, denoted as VehicleRoutes, is an input parameter. At the start, this contains all available vehicles, with each vehicle route starting at its predefined depot. The output is a set of vehicle routes containing all requests and each starting and ending at the depot.

Algorithm 2 Construct Initial Solution

```
Input: Requests, VehicleRoutes
Output: VehicleRoutes
Initialize Unassigned ← Requests
Initialize PartiallyAssignedRequests ← []
Sort Unassigned based on earliest opening time window
Shuffle vehicle IDs of VehicleRoutes
while Unassigned ≠ ∅ do
    IsAssigned ← False
    r ← first request of Unassigned
    TryTransfer ← TransferRandomly()
    if TryTransfer then
        Choose transfer station t based on smallest detour distance
        Choose v_1 ∈ VehicleRoutes with weighted randomness (inversly proportional to detour distance)
        Choose v_2 ∈ VehicleRoutes based on the least time difference with v_1
        if (p(r), t) ∈ v_1 and (t, d(r)) ∈ v_2 Feasible then
            Insert pick-up and transfer of r on v_1
            Insert transfer and delivery of r on v_2
            IsAssigned ← True
            Break
        end if
    end if
    if not IsAssigned then
        for v ∈ VehicleRoutes do
            if (p(r), d(r)) ∈ v is Feasible then
                Insert pick-up and delivery of r on route of vehicle v
                IsAssigned ← True
                Break
            end if
        end for
    end if
    if not IsAssigned then t ← ChooseTransferStation() based on least detour distance
        for v ∈ VehicleRoutes do
            if (p(r), t) ∈ v is Feasible then
                Insert pick-up and transfer of r on v
                Add r to PartiallyAssignedRequests
                IsAssigned ← True
                Break
            end if
        end for
    end if
    VehicleRoutes ← ReturnToDepot()
end while
```

F Pseudocode Operators

The destroy operators require a vehicle routing solution, denoted S_{current} and RB respectively. They return a destroyed routing solution, $S_{\text{destroyed}}$ and an updated requests bank RB. Upon which the repair operators require the destroyed routing solution and the request bank as returned by a destroy operator. The repair operator then returns a new routing solution and an updated request bank, denoted as S_{new} and RB. Additionally, index v is used to indicate a vehicle route in a routing solution.

F.1 Destroy Operators

Algorithm 3 Random destroy (DO_1)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
PotentialRequests $\leftarrow$ Filter requests without transfers
RemovedRequests $\leftarrow$ Randomly select DOD requests
$S_{\text{destroyed}} \leftarrow$ Remove RemovedRequests from S_{current}
$S_{\text{destroyed}} \leftarrow$ Update the arrival times
$RB \leftarrow$ Add RemovedRequests

Algorithm 4 Greedy destroy based on time window penalties of requests (without transfer) (DO_2)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
RequestTimePenalties $\leftarrow$ Find current time window penalties per request without transfer
RemovedRequests $\leftarrow$ Select DOD requests with worst time window penalties
$S_{\text{destroyed}} \leftarrow$ Remove RemovedRequests
$S_{\text{destroyed}} \leftarrow$ Update the arrival times
$RB \leftarrow$ Add RemovedRequests

Algorithm 5 Greedy destroy based on vehicle route (without transfers) with worst load factor (DO_3)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
VehicleLoadFactors $\leftarrow$ Find load factor per vehicle route without transfers
RemovedVehicles $\leftarrow$ Select a vehicle based on load factor, low load factor is highest probability.
$S_{\text{destroyed}} \leftarrow$ Remove the chosen vehicle route
$S_{\text{destroyed}} \leftarrow$ Update arrival times
$RB \leftarrow$ Add RemovedRequests

Algorithm 6 Greedy destroy based on highest synchronization time at transfer station (DO_4)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
TransferStationInfo $\leftarrow$ Find transferred requests and their synchronization times in S_{current}
if TransferStationInfo = None **then**
    Skip operator, because no transfers in solution
end if
SelectedTransfer $\leftarrow$ Select transfer station proportional to cum. sync. time
RemovedRequests $\leftarrow$ Select DOD requests at SelectedTransfer proportional to sync. time
$S_{\text{destroyed}}$ $\leftarrow$ Remove RemovedRequests
$S_{\text{destroyed}}$ $\leftarrow$ Update arrival times
RB $\leftarrow$ Add RemovedRequests

Algorithm 7 Cluster removal based on proximity and temporal overlap (DO_5)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
RelatedRequests $\leftarrow$ Find related requests based on date and location
RemovedRequests $\leftarrow$ Randomly select at most $\frac{\text{DOD}}{2}$ RelatedRequests proportional to relatedness
$S_{\text{destroyed}}$ $\leftarrow$ Remove RemovedRequests
PartialVehicleRoutes $\leftarrow$ Update arrival times
RB $\leftarrow$ Add RemovedRequests

Algorithm 8 Random removal of requests with long travel time (DO_6)

Input: S_{current}, RB
Output: $S_{\text{destroyed}}$, RB
PotentialRequests $\leftarrow$ Filter requests without transfers
PotentialRequests $\leftarrow$ Sort travel distances
RemovedRequests $\leftarrow$ Select DOD PotentialRequests proportional to travel distance
$S_{\text{destroyed}}$ $\leftarrow$ Remove RemovedRequests
$S_{\text{destroyed}}$ $\leftarrow$ Update arrival times
RB $\leftarrow$ Add RemovedRequests

F.2 Repair Operators

Note that x and y are some integer value, to limit the search space and improve computational efficiency.

Algorithm 9 Random insertion (RO_1)

Input: $S_{\text{destroyed}}$, RB
Output: S_{new}, RB
PotentialRequests $\leftarrow$ Randomly select $\frac{DOD}{2}$ requests from RB
for request $\in$ PotentialRequests **do**
Inserted $\leftarrow$ False
for v $\in S_{\text{destroyed}}$ **do**
Positions $\leftarrow$ Find insertion positions in v
for position $\in$ Positions **do**
CandidateRoute $\leftarrow$ Insert request at position
Feasible $\leftarrow$ Check feasibility of v ▷ Includes check of dependent vehicles
if Feasible **then**
S_{new} $\leftarrow$ Update vehicle routes
Inserted $\leftarrow$ True
Break
end if
if Inserted **then**
RB $\leftarrow$ Remove request
Break
end if
end for
end for
end for

Algorithm 10 Greedy detour insertion (RO_2)

Input: $S_{\text{destroyed}}$, RB
Output: S_{new}, RB
for request $\in RB$ **do**
LeastDetourCost $\leftarrow \infty$
Inserted $\leftarrow$ None
for v $\in S_{\text{destroyed}}$ **do**
Positions $\leftarrow$ Find insertion positions in v
for position $\in$ Positions **do**
CandidateRoute $\leftarrow$ Insert request at position of route v
Cost $\leftarrow$ Find detour distance incurred by request insertion
if Cost $<$ LeastDetourCost **then**
Skip ▷ Not the best, so skip early
end if
Feasible $\leftarrow$ Check feasibility of CandidateRoute
if Feasible **then** S_{new} $\leftarrow$ Update vehicle routes Inserted $\leftarrow$ True
end if
end for
if Inserted **then**
RB $\leftarrow$ Remove request
end if
end for
end for

Algorithm 11 Regret-2 based on empty distance (RO_3)

```
Input: S_destroyed, RB
Output: S_new, RB
while RB ≠ do
    for request ∈ RB do
        Best ← ∞, SecondBest ← ∞
        CandidateVehicles ← Select x vehicles with most empty kilometers and empty vehicles
        for v ∈ CandidateVehicles do
            CandidatePositions ← Select y lowest detour insertion in route v
            for position ∈ CandidatePositions do
                CandidateRoute ← Insert request at position of route v
                Feasible ← Check feasibility of CandidateRoute (with dependencies)
                if Feasible then
                    Cost ← Calculate empty kilometers cost after insertion
                    Best, SecondBest ← Update best and second best insertion cost if needed
                end if
            end for
            Regret ← Compute regret (second best insertion cost - best insertion cost)
            Bestvehicle ← Save the best vehicle, position
        end for
    end for
    ChosenRequest ← Choose request with highest Regret
    if No more insertions found then
        Break
    end if
    S_new ← Insert ChosenRequest in BestVehicle corresponding to best insertion cost
    RB ← Remove ChosenRequest
end while
```

Algorithm 12 Regret-2 based on time windows (RO_4)

Input: $S_{\text{destroyed}}$, RB
Output: S_{new}, RB
while $RB \neq$ **do**
  for request $\in RB$ **do**
    Best $\leftarrow \infty$, SecondBest $\leftarrow \infty$
    CandidateVehicles $\leftarrow$ Select x vehicles with most empty kilometers and empty vehicles
    for $v \in$ CandidateVehicles **do**
      CandidatePositions $\leftarrow$ Select y lowest time window penalty due to insertion in route v
      for position $\in$ CandidatePositions **do**
        CandidateRoute $\leftarrow$ Insert request at position of route v
        Feasible $\leftarrow$ Check feasibility of CandidateRoute (with dependencies)
        if Feasible **then**
          Cost $\leftarrow$ Calculate empty kilometers cost after insertion
          Best, SecondBest $\leftarrow$ Update best and second best insertion cost if needed
        end if
      end for
      Regret $\leftarrow$ Compute regret (second best insertion cost - best insertion cost)
      Bestvehicle $\leftarrow$ Save the best vehicle, position
    end for
  end for
  ChosenRequest $\leftarrow$ Choose request with highest Regret
  if No more insertions found **then**
    Break
  end if
  $S_{\text{new}} \leftarrow$ Insert ChosenRequest in BestVehicle corresponding to best insertion cost
  $RB \leftarrow$ Remove ChosenRequest
end while

Algorithm 13 Best transfer insertion (RO_5)

Input: $S_{\text{destroyed}}$, RB
Output: S_{new}, RB
for request $\in RB$ **do** DirectInsertion, Cost $\leftarrow$ Check if direct insertion is possible
  if DirectInsertion = False **then**
    Score $\leftarrow$ Calculate objective value for the best possible insertion in $v \in S_{\text{destroyed}}$
  end if
end for
CandidateRequests $\leftarrow$ Choose top x requests, sorted by descending Cost
for request $\in$ CandidateRequests **do**
  Transfer $\leftarrow$ Find transfer station proportional to detour distance
  Vehicles1 $\leftarrow$ Find top y vehicles for pick-up - transfer $\triangleright$ Feasible and least insertion cost on route
  Vehicles2 $\leftarrow$ Find top y vehicles for transfer - delivery $\triangleright$ Feasible and least insertion cost on route
  for $v_1 \in$ Vehicles1 **do**
    CandidateRoute1 $\leftarrow$ Insert request in route v_1
    Feasible $\leftarrow$ Check feasibility of CandidateRoute1 (with dependencies)
    if Feasible **then**
      Route1 $\leftarrow$ Save vehicle route
      ChosenVehicle1 $\leftarrow$ Update vehicle with least insertion cost
    end if
  end for
  for $v_2 \in$ Vehicles2 **do**
    CandidateRoute2 $\leftarrow$ Insert request in route v_2
    Feasible $\leftarrow$ Check feasibility of CandidateRoute2 (with dependencies based on ChosenVehicle1)
    if Feasible **then**
      Route2 $\leftarrow$ Save vehicle route
      ChosenVehicle2 $\leftarrow$ Update vehicle with lowest synchronization time
    end if
  end for
  if BestCost1 not ∞, BestCost2 not ∞ **then**
    $S_{\text{new}} \leftarrow$ Update routes with insertion of request at ChosenVehicle1 and ChosenVehicle2
    $RB \leftarrow$ Remove request
  end if
end for

G Parameter Tuning Results

This appendix presents the experiments conducted to tune the parameters of the algorithm. The tuning of the simulated annealing parameters, T_{init}, T_{end}, c and L, is shown in Appendices G.1, G.2, and G.3. The experiments for ALNS specific parameters, DOD and ρ, are displayed in Appendix G.4. The excecution time is given in seconds. Experiments with a run time of 1800 seconds were terminated due to the time limit.

G.1 Parameter Tuning Simulated Annealing - Iteration 1

SA1	T_{init}	T_{end}	c	L	Total obj.	Empty dist.	Dist.	Early penalty	Late penalty	LF	Nr. vehicles	Unserved req. (%)	Time (s)
1	7000	50	0.95	10	0.0473	0.2544	0.3437	0.0326	0.1377	0.76	69	2.10	744
2	3000	100	0.99	10	0.0554	0.0879	0.0397	0.1295	0.3665	0.76	67	2.73	1772
3	5000	100	0.95	20	0.1252	0.2859	0.3721	0.1110	0.2965	0.76	70	1.93	1525
4	3000	100	0.95	20	0.1925	0.4812	0.5361	0.1311	0.4522	0.75	71	1.37	1102
5	5000	50	0.95	20	0.2076	0.4446	0.6185	0.1403	0.3642	0.75	72.5	1.00	1222
6	7000	50	0.90	10	0.2348	0.4341	0.3543	0.0983	0.5161	0.75	71	2.00	349
7	5000	50	0.90	10	0.2387	0.3828	0.3427	0.0416	0.9825	0.76	68.5	2.73	262
8	5000	100	0.90	10	0.2662	0.6033	0.4608	0.0562	0.7094	0.76	70	2.73	247
9	5000	100	0.95	30	0.3114	0.6914	0.8345	0.0889	0.4538	0.74	73	1.62	1800
10	3000	50	0.90	20	0.3252	0.6578	0.7249	0.1630	0.4263	0.75	74	1.37	451
11	3000	10	0.95	10	0.3254	0.3593	0.5704	0.2663	0.0720	0.76	77.5	1.00	1063
12	3000	100	0.90	20	0.3301	0.6075	0.6912	0.1632	0.4617	0.75	74	1.37	423
13	7000	50	0.99	20	0.4102	0.4922	0.5085	0.3323	0.7088	0.76	74	2.10	1804
14	5000	10	0.99	10	0.4642	0.5373	0.7206	0.5240	0.4227	0.76	78	1.12	1800
15	7000	100	0.95	10	0.4822	0.3153	0.4280	0.4395	0.3798	0.77	78.5	1.37	924
16	7000	10	0.99	30	0.5638	0.6383	0.6672	0.2846	0.8585	0.76	77	1.62	1800
17	7000	100	0.99	20	0.6125	0.6455	0.5976	0.5700	0.7783	0.76	77	1.51	1800
18	7000	100	0.99	10	0.7228	0.5071	0.5209	0.5901	0.4789	0.78	84	1.49	1800

Table 1: Results of simulated annealing parameter tuning experiments iteration 1 (SA1).

G.2 Parameter Tuning Simulated Annealing - Iteration 2

SA2	T_{init}	T_{end}	c	L	Total obj.	Empty dist.	Dist.	Early penalty	Late penalty	LF	Nr. vehicles	Unserved req. (%)	Time (s)
1	6000	60	0.95	10	0.1346	0.3672	0.2935	0.3224	0.1835	0.76	68.5	2.23	1300
2	4000	80	0.975	15	0.2052	0.3152	0.2732	0.3092	0.3008	0.76	68.5	2.23	1418
3	5000	100	0.95	20	0.2300	0.4478	0.6199	0.1978	0.2127	0.75	72	1.12	1294
4	5000	60	0.925	10	0.2885	0.3212	0.3559	0.1803	0.4804	0.76	69.5	2.23	383
5	5000	60	0.95	10	0.3109	0.4123	0.3122	0.2286	0.5459	0.75	68.5	2.48	607
6	4000	100	0.975	15	0.3638	0.4739	0.5113	0.1267	0.1636	0.74	71.5	2.73	878
7	6000	80	0.95	15	0.4129	0.6540	0.7446	0.6578	0.1924	0.75	73	1.00	1298
8	5000	100	0.975	10	0.4206	0.4058	0.6010	0.4545	0.5517	0.76	72.5	1.12	1270
9	5000	80	0.975	15	0.4329	0.3539	0.5194	0.2184	0.5162	0.75	70	2.10	1388
10	4000	100	0.925	15	0.4340	0.4944	0.4988	0.5072	0.3402	0.76	70.5	2.23	475
11	4000	100	0.975	10	0.4464	0.3145	0.3713	0.5347	0.1411	0.76	72	2.10	1123
12	4000	100	0.95	20	0.4955	0.5098	0.7235	0.4727	0.4634	0.75	73	1.00	1445
13	4000	60	0.975	20	0.5330	0.7299	0.6691	0.3020	0.4980	0.73	72	1.98	1800
14	4000	80	0.925	15	0.5489	0.5000	0.4967	0.1668	0.3657	0.74	72.5	2.73	300
15	5000	80	0.975	20	0.5556	0.3662	0.5391	0.6180	0.4074	0.77	74	1.62	1800
16	6000	80	0.95	20	0.6103	0.5615	0.8955	0.5555	0.4368	0.76	74.5	0.87	1691
17	6000	100	0.975	20	0.7680	0.5377	0.8845	0.5633	0.5702	0.75	75.5	1.24	1800
18	6000	60	0.975	20	0.8385	0.4661	0.6574	0.7304	0.7627	0.76	75	1.49	1800

Table 2: Results of simulated annealing parameter tuning experiments iteration 2 (SA2).

G.3 Parameter Tuning Simulated Annealing - Iteration 3

SA3	T_{init}	T_{end}	c	L	Total obj.	Empty dist.	Dist.	Early penalty	Late penalty	LF	Nr. vehicles	Unserved req. (%)	Time (s)
1	5500	100	0.94	18	0.1352	0.3358	0.3563	0.1196	0.6067	0.76	68.5	2.10	1028
2	5000	90	0.95	18	0.1471	0.3669	0.3345	0.2107	0.5577	0.76	68.5	2.10	1800
3	5000	100	0.95	15	0.1674	0.3576	0.3285	0.1133	0.5378	0.76	68.5	2.36	958
4	5500	90	0.96	18	0.1807	0.7445	0.7453	0.3188	0.3274	0.75	70.5	1.00	1377
5	5000	110	0.95	15	0.1818	0.3283	0.3138	0.1909	0.5649	0.76	68.5	2.36	935
6	4500	110	0.95	12	0.2093	0.1158	0.1872	0.0842	0.8816	0.76	67.5	2.73	628
7	5000	110	0.94	15	0.2257	0.2795	0.2416	0.3750	0.5073	0.76	69.75	1.80	872
8	5500	100	0.95	15	0.3043	0.5307	0.4793	0.2615	0.4035	0.76	70.5	1.87	1200
9	5500	100	0.96	12	0.3093	0.5251	0.4747	0.2289	0.6398	0.75	69	2.36	1048
10	5000	90	0.96	15	0.3458	0.8283	0.7550	0.3761	0.5350	0.74	70.5	1.24	1184
11	4500	90	0.96	12	0.4051	0.5281	0.5618	0.6678	0.2560	0.76	71.25	1.93	942
12	5500	100	0.96	18	0.4088	0.6375	0.6634	0.0593	0.3275	0.76	71.5	1.87	1634
13	4500	100	0.94	15	0.4137	0.6894	0.6051	0.4085	0.6128	0.74	70	1.98	727
14	5000	100	0.95	12	0.4582	0.5502	0.5596	0.1869	0.5396	0.75	70	2.73	640
15	4500	90	0.94	12	0.4952	0.3825	0.5311	0.6307	0.2156	0.77	73.5	1.24	764
16	5500	90	0.95	12	0.6138	0.2760	0.2013	0.5080	0.9805	0.76	70.5	2.24	765
17	4500	110	0.96	12	0.7514	0.5976	0.6834	0.6594	0.4742	0.76	74	1.37	1045

Table 3: Results of simulated annealing parameter tuning experiments iteration 3 (SA3).

G.4 Parameter Tuning ALNS Parameters

ALNS	DOD	ρ	Total obj.	Empty dist.	Dist.	Early penalty	Late penalty	LF	Nr. vehicles	Unserved req. (%)	Time (s)
1	0.1	0.35	0.0883	0.1562	0.5023	0.1461	0.1951	0.76	70	1.62	905
2	0.1	0.7	0.0946	0.3134	0.3877	0.1369	0.2472	0.74	69.5	1.63	904
3	0.2	0.9	0.1099	0.4875	0.3789	0.1544	0.2195	0.75	68.5	2.49	1450
4	0.15	0.1	0.1435	0.2188	0.3963	0.1278	0.2338	0.76	70	2.00	1594
5	0.2	0.35	0.1775	0.2011	0.2740	0.1161	0.6211	0.76	68.5	2.23	1221
6	0.15	0.35	0.2451	0.2938	0.3888	0.0537	0.6065	0.75	69.5	1.87	1601
7	0.05	0.1	0.2730	0.0718	0.3802	0.6401	0.3729	0.76	71.5	1.37	562
8	0.15	0.7	0.2742	0.2784	0.4613	0.1948	0.4018	0.75	71	1.75	1349
9	0.1	0.1	0.2827	0.1387	0.5490	0.2787	0.3750	0.76	71	1.74	1164
10	0.05	0.7	0.2851	0.4268	0.8607	0.1874	0.3778	0.75	72	1.00	576
11	0.05	0.35	0.2934	0.2592	0.4924	0.3132	0.4787	0.76	71	1.51	571
12	0.2	0.7	0.3448	0.5578	0.4996	0.0598	0.6368	0.75	70.5	1.87	1183
13	0.2	0.1	0.3892	0.0941	0.4288	0.4204	0.6039	0.76	71	1.86	1800
14	0.15	0.9	0.4375	0.4158	0.5746	0.3335	0.7288	0.74	71.5	1.38	732
15	0.1	0.9	0.6426	0.3857	0.7422	0.5606	0.2724	0.76	75	1.49	1111
16	0.05	0.9	0.7346	0.5227	0.5000	0.1944	0.5000	0.74	73.5	2.73	228

Table 4: Results of parameter tuning experiments DOD and ρ.

H Experiment Results

Week	Solution	Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Req. unser. (%)	Partial req. (%)	Nr. transfers	Imp. w.r.t. init. (%)	Imp. w.r.t. hist. (%)
	Initial	163827	19540	9655	9152	4555	0.73	92	3.28	4.39	35.20		
W1	Historical	145552	11925	8620	2033	475	0.72	86	-	17.79	60.00		
	ALNS	129506	15988	9381	1828	7559	0.73	73	1.75		0.00	-20.95	-11.02
	Initial	170993	21869	10447	7583	4620	0.72	95	3.73	3.10	31.90		
W2	Historical	152673	13552	9284	5370	717	0.71	88	-	20.39	67.00		
	ALNS	136259	17418	10262	950	9878	0.74	77	0.74		0.00	-20.31	-10.75
	Initial	175831	20442	9776	15489	6475	0.74	97	1.46	4.38	37.30		
W3	Historical	145487	11156	8327	3483	1271	0.73	85	-	18.89	56.00		
	ALNS	133662	16075	9426	2067	8845	0.74	75	1.76		3.00	-23.98	-8.13
	Initial	163854	17993	8836	13369	7606	0.74	87	4.33	5.75	37.30		
W4	Historical	142758	10504	7978	1972	1054	0.74	88	-	23.89	60.00		
	ALNS	128603	13842	8352	2276	14633	0.75	66	3.89		2.00	-21.51	-9.92
	Initial	157569	16918	8542	8047	6312	0.77	88	4.78	3.54	29.80		
W5	Historical	150625	10217	7708	5946	3004	0.73	86	-	24.06	54.00		
	ALNS	113996	12978	8224	2167	9628	0.76	64	0.58		0.00	-27.65	-24.32
	Initial	184241	23620	10839	15335	6722	0.70	99	2.00	5.13	43.00		
W6	Historical	157715	12563	9200	7531	2172	0.73	90	-	16.32	71.00		
	ALNS	149759	17547	10331	3465	10167	0.74	83	2.07		3.00	-18.72	-5.04
	Initial	162540	18300	8838	13253	5324	0.74	88	3.95	4.27	33.00		
W7	Historical	140793	9972	7615	2583	623	0.74	65	-	20.27	46.00		
	ALNS	114895	11986	7646	954	10559	0.77	59	5.48		0.00	-29.31	-18.39
	Initial	186530	22302	10853	15048	7076	0.73	99	3.89	3.89	36.30		
W8	Historical	148994	11955	9342	5450	2247	0.74	81	-	18.25	71.00		
	ALNS	136031	16067	9555	1929	9981	0.75	72	4.14		2.00	-27.07	-8.70
	Initial	179730	21048	10164	17511	5208	0.73	97	2.50	5.30	39.30		
W9	Historical	150431	11004	8645	7528	2005	0.75	84	-	18.07	63.00		
	ALNS	140861	16408	9957	4041	9456	0.75	78	1.73		0.00	-21.63	-6.36
	Initial	184234	21220	10258	17016	7265	0.73	98	3.20	4.56	40.50		
W10	Historical	149097	10853	8683	6107	2205	0.75	83	-	16.75	53.00		
	ALNS	141308	15413	9484	1430	15231	0.76	75	2.91		3.00	-23.30	-5.22
	Initial	166682	19567	9300	13772	5294	0.72	92	1.79	4.60	35.20		
W11	Historical	144144	9832	7764	5809	1989	0.75	82	-	16.62	52.00		
	ALNS	122458	12736	8032	711	11978	0.76	64	4.60		0.00	-26.53	-15.04
	Initial	160945	18240	8832	13796	8452	0.73	84	3.65	3.45	28.50		
W12	Historical	130985	9812	7546	1074	53	0.74	81	-	16.96	57.00		
	ALNS	110421	11431	7489	1219	14032	0.76	59	1.46		0.00	-31.39	-15.70

Week	Solution	Total obj.	Empty dist.	Dist	TWearly	TWlate	LF	Nr. veh.	Req. unser. (%)	Partial req. (%)	Nr. transfers	Imp. w.r.t. init. (%)	Imp. w.r.t. hist. (%)
W13	Initial	162822	16955	8412	14265	4590	0.75	85	6.75	5.61	35.90		
	Historical	138413	10066	7767	1435	395	0.74	73	-	19.94	52.00		
	ALNS	113625	12142	7683	905	9895	0.76	60	4.56		0.00	-30.22	-17.91
W14	Initial	185257	20214	9240	19287	12266	0.72	93	4.53	6.08	39.00		
	Historical	141438	11696	8182	2528	282	0.71	83	-	21.33	67.00		
	ALNS	134735	13642	7892	2601	21600	0.74	62	6.13		0.00	-27.27	-4.74
W15	Initial	182675	18913	9357	16557	8148	0.74	93	7.14	5.27	38.00		
	Historical	142913	10808	8535	2296	24	0.75	84	-	19.18	35.00		
	ALNS	131509	13143	8296	1330	19240	0.75	64	4.86		0.00	-28.01	-7.98
W16	Initial	176411	17637	8464	18145	9290	0.74	90	5.85	5.56	36.40		
	Historical	134218	10386	7738	1050	44	0.73	76	-	18.42	50.00		
	ALNS	127740	12561	7621	2792	25017	0.74	59	3.51		0.00	-27.59	-4.83
W17	Initial	180643	16971	8783	18213	9826	0.76	89	8.72	5.11	31.90		
	Historical	134348	9969	8144	1190	45	0.76	73	-	19.83	46.00		
	ALNS	128619	13351	8430	2913	18924	0.76	64	2.79		0.00	-28.80	-4.26
W18	Initial	173951	17093	8513	17940	7656	0.76	87	8.16	4.36	30.50		
	Historical	139921	11108	8119	627	68	0.73	73	-	25.22	45.00		
	ALNS	124433	12275	7935	3835	19888	0.77	62	1.78		0.00	-28.47	-11.07
W19	Initial	191297	21183	9820	17367	10751	0.72	94	7.52	3.88	33.80		
	Historical	144781	12284	8733	2300	215	0.72	85	-	17.83	53.00		
	ALNS	137073	15282	8911	4500	17130	0.75	71	1.29		24.00	-28.35	-5.32
W20	Initial	161569	15813	8103	13174	8828	0.76	84	6.37	4.21	30.50		
	Historical	137871	10371	7614	1112	25	0.73	81	-	23.10	45.00		
	ALNS	115450	11451	7193	900	18656	0.76	55	4.97		0.00	-28.54	-16.26
W21	Initial	174652	18169	8717	16640	8126	0.73	87	7.62	6.26	38.10		
	Historical	138816	11378	8115	1707	116	0.72	82	-	19.39	41.00		
	ALNS	121719	14001	8224	2805	13939	0.74	63	2.22		0.00	-30.31	-12.32
W22	Initial	193976	20650	10207	18408	12362	0.74	98	5.42	2.91	30.70		
	Historical	140940	10850	8624	2530	187	0.75	86	-	20.42	59.00		
	ALNS	150830	13826	8846	5286	24872	0.78	74	2.88		0.00	-22.24	7.02
W23	Initial	168513	17665	8825	15160	6864	0.75	88	5.82	4.80	34.70		
	Historical	136854	10701	7901	1930	72	0.73	79	-	20.74	43.00		
	ALNS	123808	12844	8264	4272	12428	0.77	66	1.99		0.00	-26.53	-9.53
W24	Initial	170731	17373	8702	15638	7269	0.75	88	6.79	4.36	31.00		
	Historical	141373	11668	8391	2436	128	0.72	81	-	21.10	46.00		
	ALNS	122128	12327	7790	2251	15011	0.76	61	4.91		0.00	-28.47	-13.61
W25	Initial	157028	16799	8465	12918	7895	0.76	84	3.55	1.76	23.00		
	Historical	143265	10927	8043	1657	139	0.73	75	0.00	20.60	42.00		
	ALNS	117011	12556	7622	2661	19421	0.75	59	0.60	-	0.00	-25.48	-18.33

Table 5: ALNS results across all weeks.

H.1 Aggregated Experiment Results

Objective components		ALNS	Initial	ALNS impr wrt init (%)	Historical	ALNS impr wrt hist (%)
Total obj.	Mean	128258	173460	-26.06	142976	-10.29
	St. dev. (%)	8.46	6.10	-	4.28	-
Empty dist.	Mean	13892	19060	-27.12	11022	26.03
	St. dev. (%)	13.02	10.57	-	8.32	-
Dist.	Mean	8514	9278	-8.23	8265	3.01
	St. dev. (%)	10.49	8.58	-	6.28	-
Early penalty	Mean	2403	14923	-83.89	3107	-22.65
	St. dev. (%)	51.69	20.55	-	67.44	-
Late penalty	Mean	14719	7551	94.92	782	1782.15
	St. dev. (%)	33.98	28.18	-	116.01	-
Load factor	Mean	0.75	0.74	2.27	0.73	-2.76
	St. dev. (%)	1.59	2.19	-	1.62	-
Nr. vehicles	Mean	66.60	90.91	-26.74	81.20	-17.98
	St. dev. (%)	10.89	5.25	-	7.08	-
Unserved req. (%)	Mean	2.94	4.91	-40.07	-	-
	St. dev. (%)	54.77	41.50	-	-	-
Partial req. (%)	Mean	-	4.50	-	19.82	-
	St. dev. (%)	-	22.92	-	11.85	-
Nr. transfers	Mean	1.48	34.43	-95.70	53.36	-97.23
	St. dev. (%)	318.71	12.56	-	17.86	-

Table 6: Performance initial, historical and ALNS routing solutions.

I Scenario Analysis

I.1 Empty Kilometer Weight Analysis

Below the full experiments of the varying of the empty kilometer weights are presented. The asterisk marked (*) weight is the weight from the baseline experiments. Note that the empty distance column presents the absolute number of empty kilometer, without scaling it with the weight, for a clean comparison between the different weight configurations.

Week	Weight	Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Req. unser. (%)	Nr. transfers	Time (s)
W1	0.01	105657	57572	8479	1133	10969	0.74	64	2.47	0	1428
	0.1	108948	56145	8464	1221	7899	0.75	65	2.47	0	1402
	0.4	132102	54845	8199	1701	12515	0.76	65	3.56	0	1116
	0.65	140970	56424	8575	1637	8082	0.75	66	1.92	0	1217
	0.8	143524	53262	8230	973	8461	0.75	63	2.47	0	1179
	0.25*	117777	57270	8488	1652	8069	0.75	65	2.19	0	1115
W2	0.01	121845	68935	10458	1400	8797	0.75	78	1.46	0	1006
	0.1	133315	66309	10443	2607	10635	0.78	80	1.46	0	999
	0.4	148467	65889	10446	1313	10852	0.76	78	0.97	0	989
	0.65	167810	64661	9947	5005	11329	0.76	78	0.97	0	877
	0.8	174114	63248	10038	2694	9534	0.77	77	2.43	0	972
	0.25*	143506	65934	10256	3045	10973	0.77	79	1.95	0	1049
W3	0.01	123907	68444	10272	2184	7267	0.75	80	1.73	0	1243
	0.1	132056	68161	10330	2590	7820	0.75	80	2.23	0	1313
	0.4	142386	60382	9018	1399	9316	0.75	70	5.45	14	768
	0.65	149906	53273	8605	1550	10374	0.76	67	5.45	0	710
	0.8	156454	53239	8583	1532	10248	0.76	66	5.45	0	731
	0.25*	137695	63618	9891	2899	8750	0.76	77	1.98	0	1209
W4	0.01	122656	60973	9217	1472	13857	0.76	72	3.64	0	889
	0.1	133507	62875	9761	3423	12037	0.76	78	2.18	0	1090
	0.4	145520	57660	9129	828	13249	0.76	71	5.10	0	839
	0.65	162654	58962	9111	1324	14644	0.75	71	5.10	0	481
	0.8	169128	54189	9192	1479	13356	0.78	73	5.10	0	673
	0.25*	146110	61175	9193	2829	15544	0.76	75	4.61	19	972

Table 7: Full results with different empty kilometer weights.

Weight		Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Unser. (%)	Nr. transfers	Time (s)
0.01	Mean	118516	63981	9606	1547	10223	0.75	73.5	2.32	0	1141
	St. dev. (%)	6.29	7.60	8.38	25.12	24.23	0.94	8.47	36.31	0.00	18.30
0.1	Mean	126957	63373	9749	2460	9598	0.76	75.75	2.08	0	1201
	St. dev. (%)	8.20	7.23	8.06	32.14	18.83	1.61	8.26	18.05	0.00	13.55
0.4	Mean	142119	59694	9198	1310	11483	0.7575	71	3.77	3.5	928
	St. dev. (%)	4.34	5.37	8.75	23.94	13.26	0.57	6.53	46.77	173.21	14.52
0.65	Mean	155335	58330	9059	2379	11108	0.755	70.5	3.36	0	821
	St. dev. (%)	6.79	6.99	6.12	63.91	21.23	0.66	6.69	57.94	0.00	32.67
0.8	Mean	160805	55985	9011	1669	10400	0.765	69.75	3.86	0	888
	St. dev. (%)	7.38	7.46	7.61	37.76	17.52	1.46	7.94	36.69	0.00	22.68
0.25*	Mean	136272	61999	9457	2606	10834	0.76	74	2.68	4.75	1086
	St. dev. (%)	8.15	6.79	7.16	21.35	26.98	0.93	7.28	41.67	173.21	8.00

Table 8: Aggregated results of different empty kilometer weight scenarios

I.2 Vehicle Weight Analysis

Below the full experiments are presented. The experiment in week 4 for weight 750 failed before any of the stopping criteria was met, due to an error in the search process. The asterisk marked (*) weight is the weight from the baseline experiments.

Week	Weight	Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Req. unser. (%)	Nr. transfers	Time (s)
W1	100	47370	16705	10124	3607	5234	0.76	82	1.75	0	1049
	750	89667	12699	8298	764	7656	0.78	67	5.01	0	565
	1750	155496	13149	8352	952	7542	0.76	66	5.01	0	427
	2400	196762	13411	8365	1321	7665	0.76	65	5.01	0	307
	1250*	129506	15988	9381	1828	7559	0.73	73	1.75	0	1301
W2	100	46054	16889	9989	2419	8358	0.75	79	0.25	0	1037
	750	97840	17370	10193	1053	8723	0.75	78	0.98	0	912
	1750	176917	18253	10329	2010	9326	0.73	78	0.25	2	476
	2400	224233	16608	9873	1501	9851	0.75	76	1.97	2	756
	1250*	136259	17418	10262	950	9878	0.74	77	0.74	0	1034
W3	100	45279	15847	9793	2979	5259	0.77	79	1.76	0	1060
	750	91567	14744	9019	2150	7904	0.76	71	2.27	0	883
	1750	159996	14427	8860	2034	9925	0.76	69	2.02	0	1073
	2400	211305	14931	9088	1629	12156	0.75	70	2.77	0	1047
	1250*	133662	16075	9426	2067	8845	0.74	75	1.76	3	1272
W4	100	49726	14441	8693	1185	11107	0.76	68	4.17	0	1205
	1750	155284	14288	8423	1635	12438	0.75	64	3.61	0	579
	2400	195007	13321	8181	908	10997	0.75	64	4.44	0	917
	1250*	128603	13842	8352	2276	14633	0.75	66	3.89	2	964

Table 9: Full results with different vehicle weights.

Weight		Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Unser. (%)	Nr. transfers	Time (s)
100	Mean	47107	15970	9650	2548	7489	0.76	77	1.98	0	1088
	St. dev. (%)	3.58	6.05	5.85	35.02	32.64	0.93	6.93	70.83	0	6.26
750	Mean	93025	14938	9170	1322	8094	0.76	72	2.75	0	787
	St. dev. (%)	3.75	12.80	8.52	45.14	5.63	1.63	6.31	61.03	0	19.97
1750	Mean	161923	15029	8991	1658	9808	0.75	69.25	2.72	0.5	639
	St. dev. (%)	5.47	12.82	8.86	26.38	17.88	1.63	7.73	65.40	173.21	40.20
2400	Mean	206827	14568	8877	1340	10167	0.75	68.75	3.55	0.5	757
	St. dev. (%)	5.74	9.20	7.52	20.32	16.32	0.58	6.93	34.68	173.21	36.91
1250*	Mean	132008	15831	9355	1780	10229	0.74	72.75	2.04	1.25	1143
	St. dev. (%)	2.35	8.09	7.24	28.36	26.12	0.96	5.70	56.40	103.92	12.81

Table 10: Aggregated results of different vehicle weight scenarios.

I.3 Time Window Analysis

An error occured in the experiment in week 5 and the mixed category due to an error in the search process. The experiment did not produce usable results and was excluded from the aggregated table.

Week	Category	Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh	Req. unser.(%)	Nr. transfers	Time (s)
W1	A	110686	13790	8847	446	103	0.77	70	0.00	0	958
	B	118485	13454	8780	433	9568	0.77	69	0.00	0	948
	C	200008	3467	1196	9634	4461	0.78	25	75.19	10	1800
	AC	154609	15624	8635	3251	19599	0.73	68	11.28	0	1509
	Base	129506	15988	9381	1828	7559	0.73	73	1.75	0	1301
W2	A	117477	15012	9387	460	118	0.76	74	0.00	0	919
	B	129310	16333	9652	451	10374	0.74	74	0.00	0	697
	C	183420	1787	508	5340	2035	0.82	11	78.62	0	1800
	AC	183307	19647	10463	6961	21987	0.71	85	8.85	18	1800
	Base	136259	17418	10262	950	9878	0.74	77	0.74	0	1034
W3	A	107844	13594	8718	426	105	0.76	68	0.00	0	768
	B	116187	12824	8564	422	9377	0.78	68	0.00	0	1037
	C	189640	2450	775	7206	210	0.94	16	80.10	0	1800
	AC	156893	14037	8040	2283	26782	0.75	63	13.60	0	832
	Base	133662	16075	9426	2067	8845	0.74	75	1.76	3	1272
W4	A	105111	13536	8515	459	101	0.76	66	0.00	0	647
	B	114458	13373	8483	442	8410	0.77	67	0.00	0	808
	C	211764	13038	5462	19153	54861	0.68	57	26.67	7	1114
	AC	165275	15056	8506	5881	26582	0.74	71	11.39	0	1800
	Base	128603	13842	8352	2276	14633	0.75	66	3.89	2	964
W5	A	99690	12407	8061	375	98	0.78	63	0.00	0	671
	B	109803	12658	8111	376	9908	0.77	63	0.00	0	1045
	C	216591	13551	5176	14862	44752	0.67	61	35.94		1800
	Base	113996	12978	8224	2167	9628	0.76	64	0.58	0	953

Table 11: Full results of different time window categories.

Category		Total obj.	Empty dist.	Dist.	TWearly	TWlate	LF	Nr. veh.	Unser.(%)	Nr. transfers	Time (s)
A	Mean	108162	13668	8706	433	105	0.77	68.20	0	0	793
	St. dev. (%)	5.46	6.06	4.97	7.28	6.81	1.04	5.44	0	0	16
B	Mean	117649	13728	8718	425	9527	0.77	68.20	0	0	907
	St. dev. (%)	5.52	9.75	5.90	6.22	6.86	1.77	5.20	0	0	15
C	Mean	200285	6858	2624	11239	21264	0.78	34.00	59.30	4.25	1663
	St. dev. (%)	6.30	77.05	84.38	45.26	110.81	12.76	61.58	38.96	103.07	16
AC	Mean	165021	16091	8911	4594	23738	0.73	71.75	11.28	4.50	1485
	St. dev. (%)	6.84	13.24	10.36	41.30	12.91	2.02	11.38	14.92	173.21	27
Base	Mean	128405	15260	9129	1858	10109	0.74	71.00	1.74	1.00	1105
	St. dev. (%)	6.01	10.60	8.28	25.70	23.76	1.37	7.18	67.68	126.49	14

Table 12: Aggregated results of different time window categories.

I.4 Requests Received on Time

Each of the columns are in percentage deviation from the baseline experiments.

Week	Dist.	Empty dist.	TWearly	TWlate	LF	Unserv. penalty	Vehicles penalty	Req reduced (%)
W1	-6.03	-11.58	51.51	-12.53	2.74	24.93	-2.66	-8.52
W2	-10.24	-10.05	94.94	-33.62	0.00	81.37	-6.72	-8.11
W3	-3.75	-4.42	-14.47	-14.63	0.00	57.54	-4.43	-9.32
W4	2.34	5.24	-36.03	-3.39	0.00	12.15	-3.14	-10.83
W5	7.10	6.91	26.30	-29.93	1.32	258.31	7.17	-2.32
W6	3.96	5.98	-37.24	55.23	0.00	-17.88	3.04	-5.29
W7	5.78	7.13	51.20	-22.96	-1.30	-56.02	6.21	-9.04
W8	6.34	3.87	44.22	28.41	1.33	-63.00	4.85	-4.62
W9	-4.18	-2.91	-18.97	32.40	0.00	165.69	-1.97	-3.22
W10	5.60	3.11	386.95	-26.57	2.63	-64.33	17.00	-6.55
W11	6.67	10.20	124.03	17.67	-1.32	-39.49	8.91	-8.18
W12	3.61	7.25	23.36	-2.02	-1.32	29.06	2.08	-7.02
W13	-0.46	1.97	229.51	13.58	0.00	-38.30	0.55	-8.83
W14	4.88	6.60	46.10	-10.69	1.35	-44.10	8.87	-6.67
W15	6.99	10.01	125.25	-0.99	0.00	-72.19	10.63	-5.37
W16	-7.20	-11.49	-26.93	-11.39	4.05	44.30	-6.45	-7.60
W17	1.69	2.45	45.63	-10.42	1.32	-26.08	3.95	-5.31
W18	-1.44	2.09	-18.95	-18.94	0.00	-8.52	-4.39	-8.90
W19	-5.27	-6.80	-68.60	7.59	1.33	177.19	-11.41	-6.20
W20	11.16	17.38	100.48	7.98	-1.32	-67.66	13.97	-9.06
W21	-0.12	-2.17	31.93	26.26	1.35	9.06	2.14	-8.31
W22	8.44	9.26	-55.35	-3.69	-1.28	-42.28	0.10	-5.50
W23	0.44	2.62	-77.85	36.14	-1.30	59.64	-6.88	-10.51
W24	9.00	14.66	63.99	32.60	-1.32	-42.94	6.02	-7.23
W25	0.45	-1.16	39.06	-3.21	1.33	69.76	1.67	-11.64

Table 13: Full results of scenarios without late requests