Optimizing Meilink Borculo B.V.'s Transportation Planning: Cost-Effective Logistics through Vehicle Routing

Master Thesis

March 29, 2024

Student:

Tanser Karakash
s2104911
t.karakash@student.utwente.nl

University supervisors:

Dr.ir. E.A. Lalla (Eduardo)
Dr. A. Trivella (Alessio)

Company supervisor:

P. Smit (Peter) - Adjunct-directeur
Operations Meilink Beheer Borculo B.V.

Department of Industrial Engineering & Business Information Systems

Preface

Dear reader,

You are about to read the master thesis titled *Optimizing Meilink Borculo B.V.'s Transportation Planning: Cost-E ective Logistics through Vehicle Routing*. This research was conducted at Meilink Borculo B.V. in Borculo, the Netherlands, as the final assignment for my Master's in Industrial Engineering and Management at the University of Twente.

Throughout my Master's program, I have acquired extensive knowledge on various models, and theoretical and practical aspects across different directions of the IEM programme. In the first year of my education, I realized that transportation and logistics management within the IEM program captivated my interest the most. Consequently, I sought to gain more practical knowledge in this area through my Master's thesis assignment.

At Meilink Borculo B.V., I have gained numerous new insights and experiences, for which I am immensely grateful. I wish to extend my gratitude to my company supervisor, Peter Smit, who provided me with this opportunity, believed in my capabilities and skills to execute this project, and consistently guided and supported me throughout the process.

Undoubtedly, I also thank my UT supervisor, Eduardo Lalla, for his invaluable help and patience. His extensive feedback, profound expertise in the field, and practical tips were very crucial. Thanks to Eduardo's profound expertise and comprehensive feedback, I was able to enhance my knowledge, improve my thesis, and learn something new at every meeting we held. I would also like to express my appreciation to my second supervisor, Alessio Trivella, for his enthusiasm and insightful feedback.

Lastly, I am thankful to my family and friends for their consistent support, not only during this research but also throughout my entire study period. I have thoroughly enjoyed these five years of studying at the University of Twente and am looking forward to the next chapter in my life. My journey of learning does not end here, as we continue to learn throughout our lives.

I hope you find reading my thesis as enjoyable as I found working on it!

Tanser Karakash

Enschede, March 2023

Management Summary

This research was conducted at Meilink Borculo B.V., located in Borculo, the Netherlands. As one of the leading industrial packaging companies in the country, Meilink specializes in a wide range of custom packaging solutions, including wooden boxes, cardboard packaging, exhibit packaging, and flight cases. The company is committed to enhancing customer service by also managing the transportation of products to and from its production sites.

Since Meilink started as a family business, manual operations still dominate their workflow. This manual approach can be inefficient and costly, requiring attention as a core problem which was addressed within this thesis. The objective of this research was to solve the core problem of manual transportation planning. Because Meilink Borculo B.V.'s transportation services have significant prices, its primary objective is to lower those costs by optimizing the routing process. Thus, the research question was formulated as follows:

"How can Meilink Borculo B.V.'s current transportation planning processes be optimized to reduce its transportation costs?"

To achieve this, the study aimed to:

- Analyze the current transportation system and identify inefficiencies.
- Explore applicable techniques and methods for formulating and solving complex vehicle routing problems (VRP) in the existing literature.
- Develop and evaluate a solution approach tailored to Meilink's specific problem characteristics and KPIs.

The initial phase of this research performed comprehensive analysis to understand the current logistics operations at Meilink Borculo B.V. This involved gathering data on various critical aspects for model development, including the logistics structure, types of vehicles, routing practices, constraints, cost structure, and Key Performance Indicators (KPIs). The analysis revealed that Meilink's transportation system is complex, characterized by daily delivery operations and categorization of order requests into deliveries from the depot to customers and pickups from customers and suppliers to the depot.The analysis identified a routing optimization challenge at Meilink, aiming to reduce transportation costs. An extensive literature review was conducted to explore suitable modeling approaches and solution methods for Meilink's specific transportation characteristics.

The literature review clarified that Meilink's transportation problem can be formulated as a Vehicle Routing Problem (VRP). More specifically, the transportation problem of the company in case aligns with a Multi-Trip Capacitated Vehicle Routing Problem with Divisible Delivery and Pickup Time Windows and Private Fleet and Common Carriers (MTCVRPDDPTWPFCC). Although there are studies, which have been researching each of these characteristics, either separately or in combination, there is no model in the literature, which provides a comprehensive overview of all those characteristics together. In VRP literature, the closest model has five out of the six characteristics of Meilink's problem. Subsequently, the thesis proposed a Mixed Integer Linear Programming (MILP) model taking all the problem constraints into consideration. Due to the complexity of the model, the MILP is not able to solve the problem for large instances in an efficient computation time. Therefore, a metaheuristic approach was formulated, namely, a Variable Neighbourhood Search (VNS) metaheuristic, which is able to handle the larger data instances.

For the experimental design, artificial data instances were developed for parameter tuning, together with real-world data instances from company data. The settings for the VNS and MILP algorithms, such as computation time and iteration limits, were optimized to ensure applicability to similar VRPs. For the MILP, the maximum running time was set to 1800 seconds, which is aligned with the requirements from Meilink. The VNS algorithm consists of initialisation phase, where a random initialisation was selected, a shaking phase, with adaptive shaking on every 25% of the iterations without improvement, and a local search phase with five distinct operators - Swap vehicles, 2-Opt, Swap customer, Reinsertion, and Move.

Following parameter optimization, four experiments evaluated the methodologies under various conditions, examining their impact on the objective function across different fleet compositions and routing strategies. The experiments were focused on testing the models under different conditions and different number of instances, ranging between 7 and 100 nodes. The scenarios were focused on evaluating the efficiency and cost improvement of relying on mixed fleet, only private fleet, and solely common carriers.

The last experiment was performed to showcase the contribution of routing all vehicles in the fleet and the cost contribution of using such strategy.

The results of the experiments revealed the following:

- The MILP model performs better than the VNS in low-demand data instances by finding the optimal solution within reasonable time. However, the exact method generates high optimality gaps in high-demand scenarios, leading to very high costs and an average computation time of 1260 seconds.
- The VNS solution approach performs better than the MILP in high-demand data instances by generating high improvements of approximately 30% on average, compared to the initial solution. The VNS solves the problem within a reasonable computation time averaging only 18.53 seconds, excluding the initialisation.
- By using the VNS approach, the current costs can be improved up to 64%.
- The average objective values of the VNS are lower than the ones' of the MILP model across the feasible experiments.
- Relying solely on private fleet is not optimal for the company, since the vehicles are unable to serve all customers in high demand scenarios, generating an average of 23.40% unserved customers across the real-world data instances.
- Using solely common carriers, without routing them could increase the transportation costs on average.
- Routing the vehicles is essential for decreasing overall transportation costs.
- The most optimal strategies are using mixed fleet or relying solely on common carriers. However, in both cases, all vehicles should be routed to achieve the best cost performance.
- For optimal private fleet utilization, trucks with IDs 3, 4, and 5 are recommended, either by incorporating additional vehicles within the private fleet or alongside external trucks.
- Replacing manual transportation planning with optimized approach and incorporating that in a tool can provide the company with an optimal resource utilization and improve the transportation cost performance.

The solution approaches detailed in the thesis—namely, the Mixed Integer Linear Programming (MILP) model for smaller instances and the Variable Neighborhood Search (VNS) metaheuristic for larger ones—provide a comprehensive framework for optimizing Meilink Borculo B.V.'s transportation process. The results demonstrate significant improvements in operational efficiency and cost reductions. Therefore, it is recommended that Meilink Borculo B.V. incorporate routing tools into its operations, moving away from manual transportation planning. Additionally, the company should continue utilizing its private fleet vehicles as long as their daily fixed costs remain stable. However, it may need to either expand its private fleet or continue using common carrier services. Given the company's ability to control routing for both internal and external vehicles, it is highly recommended to do so, as this leads to significant cost reductions. The most cost-effective approach involves using common carriers exclusively, provided the company can ensure a proper routing strategy.

List of Acronyms

- **ACO** – Ant Colony Optimization
- **AVNS** – Adaptive Variable Neighbourhood Search
- **CVRP** – Capacitated Vehicle Routing Problem
- **GRASP** – Greedy Randomized Adaptive Search Procedures
- **ILS** – Iterated Local Search
- **KPI** – Key Performance Indicator
- **MILP** – Mixed Integer Linear Program
- **MTVRP** – Multi-Trip Vehicle Routing Problem
- **MTCVRPDDPTWPFCC** – Multi-Trip Capacitated Vehicle Routing Problem with Divisible Delivery and Pickup Time Windows and Private Fleet and Common Carriers
- **SA** – Simulated Annealing
- **SDVRP** – Split Delivery Vehicle Routing Problem
- **TS** – Tabu Search
- **VND** – Variable Neighbourhood Descent
- **VNS** – Variable Neighbourhood Search
- **VRP** – Vehicle Routing Problem
- **VRPB** – Vehicle Routing Problem with Backhauls
- **VRPCB** – Vehicle Routing Problem with Clustered Backhauls
- **VRPDDP** – Vehicle Routing Problem with Divisible Delivery and Pickup
- **VRPMB** – Vehicle Routing Problem with Mixed Linehaul and Backhaul
- **VRPPFCC** – Vehicle Routing Problem with Private Fleet and Common Carrier
- **VRPSDP** – Vehicle Routing Problem with Simultaneous Delivery and Pickup
- **VRPTW** – Vehicle Routing Problem with Time Windows

Contents

Preface i

Management Summary ii

List of Acronyms iv

1 Introduction 1
1.1 Company description 1
1.1.1 Meilink Borculo B.V 1
1.1.2 Transportation system 1
1.2 Problem statement 2
1.2.1 Action problem 3
1.2.2 Core problems 3
1.3 Research goal 3
1.4 Research Questions 4
1.5 Research design 5

2 Problem context analysis 7
2.1 Transportation planning process 7
2.1.1 Order processing and planning 7
2.1.2 Clustering and route optimization 8
2.1.3 In-house truck ownership vs. utilizing common carriers 8
2.2 Characteristics 9
2.2.1 Type of orders 9
2.2.2 Time windows 10
2.2.3 Vehicles 11
2.3 Current transportation cost structure 11
2.4 Key performance indicators 12
2.5 Requirements and limitations 12
2.6 Conclusion 13

3 Literature review 14
3.1 Vehicle Routing Problem 14
3.1.1 Mathematical model 15
3.2 VRP with backhauls 16
3.2.1 Mathematical model for VRPDDP 17
3.3 VRP with time windows 18
3.3.1 Mathematical model for VRPTW 19
3.4 Split delivery VRP 19
3.4.1 Mathematical model for SDVRP 19
3.5 VRP with private fleet and common carrier 20
3.5.1 Mathematical model for VRPPFCC 20
3.6 Multi-trip VRP 21
3.6.1 Mathematical model for MTVRP 21
3.7 Combining multiple features of the VRP 22
3.8 Heuristics 24
3.8.1 Constructive heuristics 24
3.8.2 Simulated Annealing 24
3.8.3 Tabu Search 25
3.8.4 Iterated Local Search 25
3.8.5 Variable Neighbourhood Search 25
3.8.6 Variable Neighbourhood Descent 26
3.8.7 Ant Colony Optimization 26
3.8.8 Greedy Randomized Adaptive Search Procedures 26
3.9 Conclusion 27

4 Solution Approach **28**
4.1 Model description . . . 28
4.1.1 Requirements . . . 28
4.1.2 Assumptions . . . 29
4.2 Mathematical formulation . . . 29
4.2.1 Parameters . . . 30
4.2.2 Decision variables . . . 30
4.2.3 Objective function . . . 30
4.2.4 Constraints . . . 31
4.3 Toy problem illustration . . . 32
4.4 Solution Methodology . . . 33
4.4.1 Variable Neighbourhood Search . . . 34
4.4.2 Initial solution . . . 34
4.4.3 Shaking . . . 36
4.4.4 Local Search . . . 37
4.4.5 Operators . . . 37
4.5 Conclusion . . . 40

5 Evaluation **41**
5.1 Experiment Design . . . 41
5.1.1 Technical details . . . 42
5.2 Data Instances . . . 42
5.2.1 Artificial data instances for parameter tuning . . . 42
5.2.2 Real-world instances . . . 43
5.2.3 Cost data for numerical experiments . . . 44
5.3 Parameter Tuning . . . 45
5.3.1 Initialisation . . . 45
5.3.2 Number of iterations without improvement . . . 46
5.3.3 Order of local search operators . . . 48
5.3.4 Adaptive VNS . . . 50
5.3.5 Maximum running time of MILP . . . 51
5.3.6 Final parameters . . . 52
5.4 Scenario Evaluation . . . 52
5.4.1 Mixed Fleet Utilization . . . 52
5.4.2 Exclusively private fleet . . . 55
5.4.3 Common carriers only . . . 57
5.4.4 Routing common carriers . . . 59
5.4.5 Comparison across the experiments . . . 62
5.5 Conclusion . . . 64

6 Conclusions, recommendations, and future research **65**
6.1 Conclusions . . . 65
6.2 Contribution to theory . . . 65
6.3 Contribution to practice . . . 66
6.4 Recommendations . . . 66
6.5 Limitations and future research . . . 66

References **68**

Appendix **72**
A Algorithms of the Metaheuristics Discussed in Literature Review . . . 72
A.1 Simulated Annealing Algorithm . . . 72
A.2 Tabu Search Algorithm . . . 72
A.3 Iterated Local Search . . . 73
A.4 Variable Neighborhood Descent . . . 73
A.5 Greedy Randomized Adaptive Search . . . 73
B Customer locations served predominantly by external vehicles (common carriers) . . . 74
B.1 Experiment E1 . . . 74
B.2 Experiment E4 . . . 74

1 Introduction

This chapter contains a description of the business in the case and its problems. Section 1.1's objective is to provide the reader with an overview of the industry in which the firm operates and its products and services. Section 1.2 introduces the problem cluster, together with the core problems which are going to be considered within this thesis. Section 1.3 outlines the goal of this research. Section 1.4 gives the main research question and all sub-questions based on the problem statement and the core problem. Finally, Section 1.5 provides an overview of the research design that is followed within the thesis.

1.1 Company description

Industrial packaging is critical in protecting commodities from damage, moisture, temperature, and contamination. The primary goal of industrial packaging is to preserve industrial commodities during transportation and storage (ITPPackaging, 2020). Because industrial packaging is a crucial component in extending a product's life, this industry has been developing, and in 2022, it was valued at over 6000 million USD, with growth predicted to double in the next five years (TheExpressWire, 2023).

Meilink Borculo B.V. is one of the largest industrial packaging companies in the Netherlands. The company has a family business history, starting in 1874 with sawing planks of oak for local carpenters, and then extending into the industrial packaging sector (MeilinkB.V., nd). Meilink specializes in custom packaging, such as wooden boxes, cardboard packaging, exhibit packaging, and flight cases. Depending on the client's requirements, the package design may differ in terms of durability, handling, material type, and cost. Packaging may occur either at the customer facility, which provides the client with shorter lead times, or at one of Meilink's facilities. To provide better services to the customers, the company engages also in transporting the products from and to one of the production sites.

1.1.1 Meilink Borculo B.V

The company has a total of seven offices in the Netherlands, five of which are production and storage locations as well. This research is conducted at the head office of Meilink, located in Borculo. The Borculo location is the largest out of all five production locations, specializing in various types of industrial packaging, but also in producing standardized boxes and packages. While the other sites rely solely on third-party logistics companies to deliver the items within the Netherlands via road transport, the office in Borculo has its private trucks. The company's vehicles are used mostly for delivery to customers via intra-Netherlands transportation, whereas third-party transportation vehicles are used when a product needs to be sent a long distance, internationally, or in some other special cases. Hereafter, the thesis focuses only on the Borculo location, referred to as Meilink or Meilink Borculo B.V., and the other office locations are out of the scope of this paper.

1.1.2 Transportation system

The transportation process of Meilink involves a number of parties. The company utilizes its own vehicles to transport items to clients throughout the Netherlands every day of the week. Figure 1 depicts a simplified version of the different flows of products from and to the Borculo office.

Meilink is responsible for daily product deliveries within the Netherlands, where it uses either its own trucks or external transportation services. The company seldom picks up items for packaging from customers and transports them to the Borculo warehouse. Most of the time, clients arrange external services in advance. Typically, the raw materials also come in a vehicle arranged by the supplier. However, for specific products, Meilink trucks are scheduled to pick up the raw materials and deliver them at the Borculo warehouse. Meilink Borculo primarily relies on its in-house vehicles as a preferred choice for transporting its products, but often also employes common carrier vehicles.

In the case of a client from abroad, Meilink Borculo B.V. is not responsible for arranging the transportation to the production location. Instead, Varekamp Project Services (VPS), also part of Meilink Group, arranges for the goods to be transported first to ports, and then to the Borculo office with external carriers. Once packed or produced, the order is collected by external carriers and dispatched to the client abroad.

Meilink's offices are disconnected from one another, and each site operates as an independent entity. The trucks at Borculo are not responsible for pick-up from Meilink's other locations. Meilink Borculo rarely delivers some goods to the other offices either through its private vehicles or through external carriers. Other locations may also send some items to Borculo, but those use only external carriers.

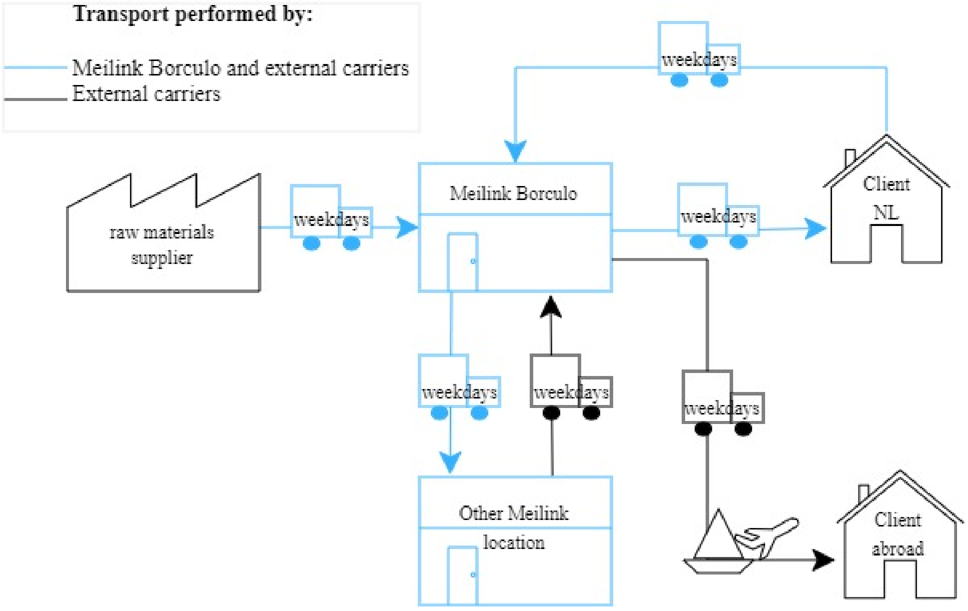

Figure 1. Simplified overview of the different transportation flows between the Meilink offices, clients and suppliers. The flows in blue are performed by both Meilink Borculo in-house trucks and common carriers, while those in black are performed only by external service providers.

1.2 Problem statement

Several interviews with Meilink professionals were held to discover the true problem that has to be handled. Figure 2 depicts the problem cluster. The identified action problem is high transportation costs (red node). All the difficulties raised during the interviews were grouped in a problem cluster with their associated linkages. The white nodes are all the subproblems, which have been identified during the interviews. The core problem (green node) will be tackled in this research.

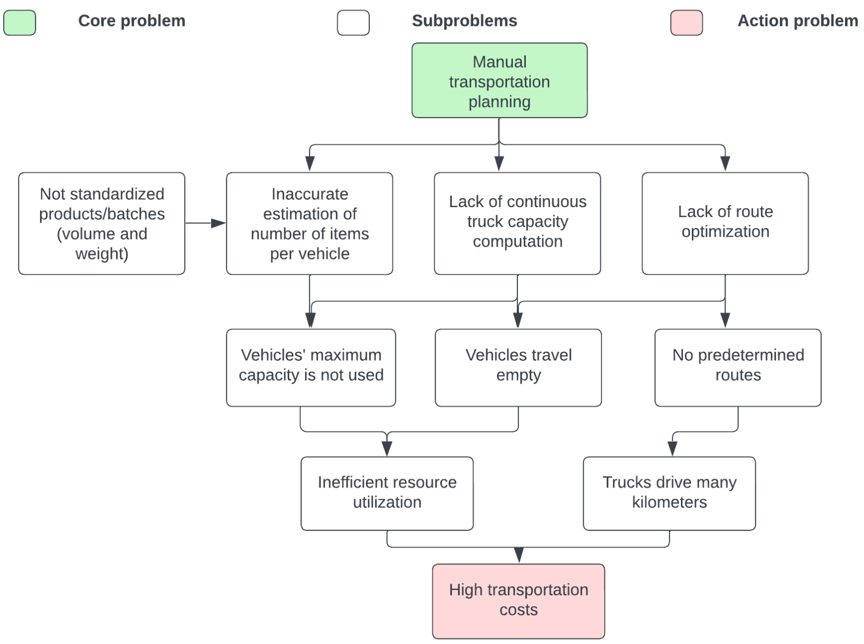

Figure 2. Problem cluster describing the core and subproblems at Meilink Borculo B.V. The core problem to be resolved in this thesis is coloured in green, whereas the action problem is coloured in red.

1.2.1 Action problem

As shown in Figure 1, the transportation process of Meilink consists of several parties (Section 1.1.2). The company has to deliver to customers and other office locations, and sometimes collect raw materials from the suppliers or a product for packing from the clients. This intricate transportation process is performed by both external service providers and the company's own trucks.

The result of the transportation network is the action problem of *`High transportation costs'*. An action problem can be defined as any situation which is not in the way the problem owner wants it to be (Heerkens and van Winden, 2017). In this context, the gap lies in the financial challenge created by transportation planning. This challenge has to be further investigated by studying the underlying issues contributing to it.

1.2.2 Core problems

A problem cluster is formulated with the aim of identifying potential problems and underlying causes, as outlined by Heerkens and van Winden (2017). Core problems are identified by examining issues that lack direct causes. In Figure 2, two core problems are identified: *manual transportation planning* and *non-standardized products/batches (in terms of volume and weight)*.

In this research, the core problem selected is manual transportation planning. This decision is based on its potential for change within a six-month timeframe and the availability of resources. Given that Meilink Borculo B.V. offers unique packaging, standardizing product volume and weight is not feasible. Additionally, manual transportation planning is chosen over non-standardized products/batches due to the former's greater impact on operational efficiency. Manual transportation planning, same as non-standardized products/batches, also leads to inaccuracies in estimating the number of items per vehicle, resulting in underutilized vehicle capacity and potentially high transportation costs. The subsequent lines offer brief descriptions of the two core problems and their associated impacts on the company's transportation process.

1.2.2.1 Not standardized products/batches (volume and weight)

Meilink, as a customer-oriented firm, offers unique packaging, and each of its clients has individual needs in terms of material kind, durability, and size. The items are non-standardized and vary in weight and volume. The personalized character of the items necessitates calculating their volume and determining the optimum method of transportation. Because there is no specialized software for carrying out the planning, the logistics department cannot predict how much space the items will take exactly and how many of them can fit into a single truck. As a result, vehicles are frequently underutilized because of the uncertainty of how much cargo may be carried. This results in inefficient use of resources, particularly Meilink's vehicles, resulting in excessive transportation expenses.

1.2.2.2 Thesis core problem: Manual transportation planning

Since Meilink started as a family business, manual operations still dominate their workflow. This is valid also for transportation planning, where the logistics department relies heavily on employee experience for planning the transportation of goods. Every morning the logistics department manually determines the routes, by considering the accumulated orders and available vehicles. The manual approach to route planning can be very subjective and time-consuming and does not take into account all of the available data. Planning the routes can lead to inaccurate estimation of the number of items which fit within a single truck which results in underutilized vehicle capacity. Since employees are unable to update the truck capacity through the execution of routes and delivery to clients, the vehicles often end up travelling empty. Trucks have to travel longer distances because the routes are not optimized and there is no predetermined order of visiting the customers. This manual approach and the subproblems caused by it can be inefficient and costly. Therefore, manual transportation planning requires attention as a core problem to be addressed within this thesis.

1.3 Research goal

The objective of this research is to address the core problem of manual transportation planning, as defined in the problem statement (Section 1.2). Meilink Borculo B.V. faces significant transportation costs due to its reliance on both private fleet and common carriers. To effectively tackle the issue of high transportation costs, it is essential to optimize the routing process. This optimization involves addressing the complexities inherent in managing various trucks within the private fleet, coordinating with common carriers, and adhering to specific operational constraints.

By delving into the operational-level problem of daily vehicle routing, this research aims to provide tactical guidance to the organization. At the tactical level, strategic decisions regarding the deployment and utilization of resources, such as private fleet and common carriers, can significantly impact overall transportation costs. Therefore, by resolving the underlying Vehicle Routing Problem (VRP) while considering the unique characteristics, criteria, and restrictions of the company, this study attempts to offer actionable insights.

By addressing both tactical and operational levels, this research aims to offer a comprehensive solution that not only optimizes daily routing processes but also provides strategic recommendations for long-term cost reduction and operational efficiency improvements.

1.4 Research Questions

Based on the problem statement and research goal described in Sections 1.2 and 1.3 the research question is formulated as follows:

"How can Meilink Borculo B.V.'s current transportation planning processes be optimized to reduce its transportation costs?"

In order to answer the main research question, several sub-questions are defined. Those act as a path towards getting a systematic answer to the main research question and solving the core and action problems. The sub-questions are divided into five categories and vary from understanding the context of the process and gathering knowledge from the literature to solution generation, implementation, and evaluation. The research sub-questions are explained below.

Current situation and problem context analysis

The first step of this research is analyzing and understanding the current situation. For this purpose, information needs to be collected on different characteristics important for the model formulation. The current logistics structure, number and types of vehicles owned, route planning, and all the existing constraints and limitations, as well as the cost structure and other Key Performance Indicators (KPIs), have to be determined. More specifically, the following research questions have to be answered:

1. What is the current transportation system offered by Meilink Borculo B.V?
   - 1.1. What are the current transportation characteristics of Meilink?
   - 1.2. Which criteria and factors influence the company's route planning decisions?
   - 1.3. What are the associated transportation costs for Meilink Borculo B.V?
   - 1.4. What are the KPIs, constraints, and requirements linked to the logistics process of Meilink?

Literature review and analysis

The second part of the thesis focuses on collecting knowledge for different optimization techniques proposed in the literature, related to the thesis problem. A look into the literature can contribute to understanding the existing models and theories and give insight into how to formulate the most suitable model and create a foundation for the solution approach.

2. Based on the literature, what are the applicable techniques and methods for modelling and solving a vehicle routing problem?
   - 2.1. Which VRP variant aligns with the characteristics of Meilink's transportation process?
   - 2.2. What are the solution approaches for the VRP proposed in the literature?

Design of solution approach

The third part is related to designing the solution approach and selecting a technique to solve the model. This should be done in accordance with the time and resource constraints available. Moreover, the data that is going to be used and the assumptions should be defined. Thus, the questions to be answered are:

3. How should the solution approach for Meilink's transportation process be designed?
    3.1. How can the routing problem and solution approach for Meilink be formulated?
    3.2. What are the requirements which have to be met by the solution approach?
    3.3. What assumptions are underlying the solution approach?

Experimentation and evaluation

Once the model is formulated and the solution approach is designed, the performance of the model has to be evaluated. Therefore, the solution has to be tested under various scenarios and evaluated under each to have a good overview of its performance. The questions below provide an overview of the content of this part:

4. How does the developed solution for optimizing Meilink's transportation process perform compared to the current situation?
    4.1. Which scenarios are interesting to be investigated?
    4.2. How does the solution approach perform in terms of costs and the other KPIs under the different scenarios considered?

Conclusion, recommendations, and limitations

The final step is to provide an answer to the main research question and give recommendations to Meilink based on the solution outcome. This requires answering the following questions:

5. What are the main conclusions and recommendations that can be drawn from the analysed results?
    5.1. What are the main outcomes of the conducted research?
    5.2. What are the recommendations that can be proposed to Meilink from the results of the experiments?
    5.3. What are the research's theoretical and practical implications and corresponding limitations?

1.5 Research design

The study is divided into phases, each aiming to discover a solution to the main research question. Figure 3 depicts the procedure, together with the research questions to be answered in each phase, the required inputs, and the desired output.

The problem identification step is discussed in Chapter 2, along with the problem context, current transportation planning and process. Chapter 3 presents an overview of the literature, focusing on existing VRP models and solution approaches. Chapters 2 and 3 work together to provide the solution's conceptual structure. In Chapter 4 a solution approach for Meilink's transportation problem is developed by establishing a mathematical model, solving and optimizing it by considering existing data and expert opinion. Experiments are carried out and evaluated under different scenarios in Chapter 5. The conclusions and recommendations for Meilink resulting from the experiments are provided in Chapter 6.

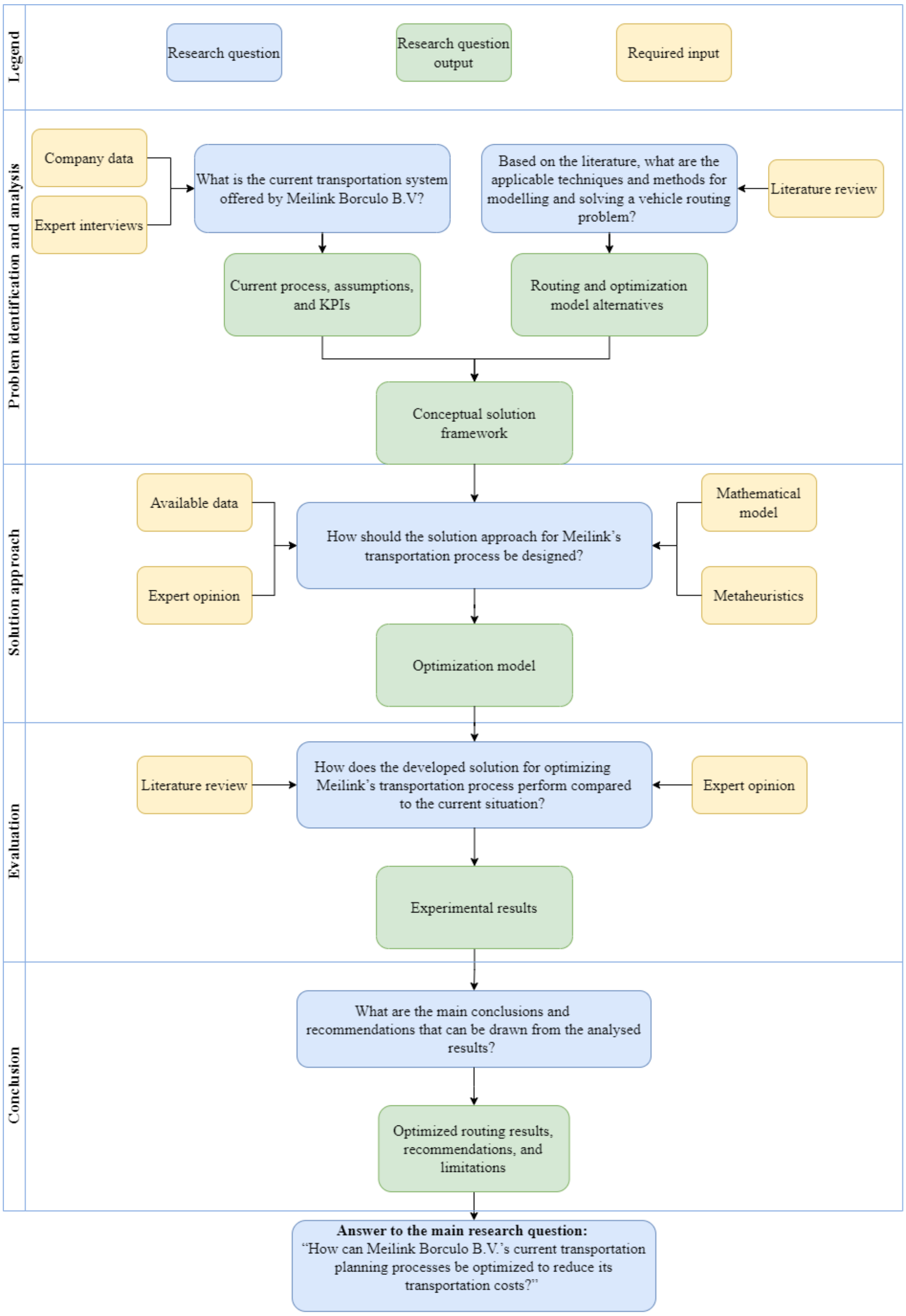

Figure 3. Graphical representation of the research design.

2 Problem context analysis

This chapter is dedicated to answer the research question *\What is the current transportation system o ered by Meilink Borculo B.V?"*. Therefore, the current transportation planning process within Meilink Borculo is introduced in Section 2.1, together with the different factors which influence the employees' decision-making. In Section 2.2 the current situation at Meilink Borculo B.V. is described with the characteristics relevant to the problem. Section 2.3 provides an overview of the current cost structure of the transportation services in Meilink. The KPIs linked to the logistics process of Meilink are outlined in Section 2.4, followed by the problem requirements and limitations in Section 2.5.

2.1 Transportation planning process

The transportation system at Meilink is a complex process that involves weekday deliveries and requires transportation decisions to be made on a daily basis. Transportation flows happen between the company and its customers, suppliers and other company locations (Section 1.1.2). This process operates without a dedicated route planning system but relies on several critical aspects to manage the transportation of goods. Within this section, an overview of the planning process is developed, together with the current decision-making criteria for routing.

2.1.1 Order processing and planning

Currently, Meilink engages in daily item deliveries, necessitating transportation decisions to be made on a daily basis. Notably, there is no unique route planning system in place. However, several factors significantly impact the decision-making. A comprehensive overview of the transportation planning process for order deliveries of the Borculo location is provided in Figure 4. The planning is usually made at the beginning of each day with the accumulated orders. The decision-making chart is followed for each order, for which the deadline is the route planning day and is carried out manually by the department's staff.

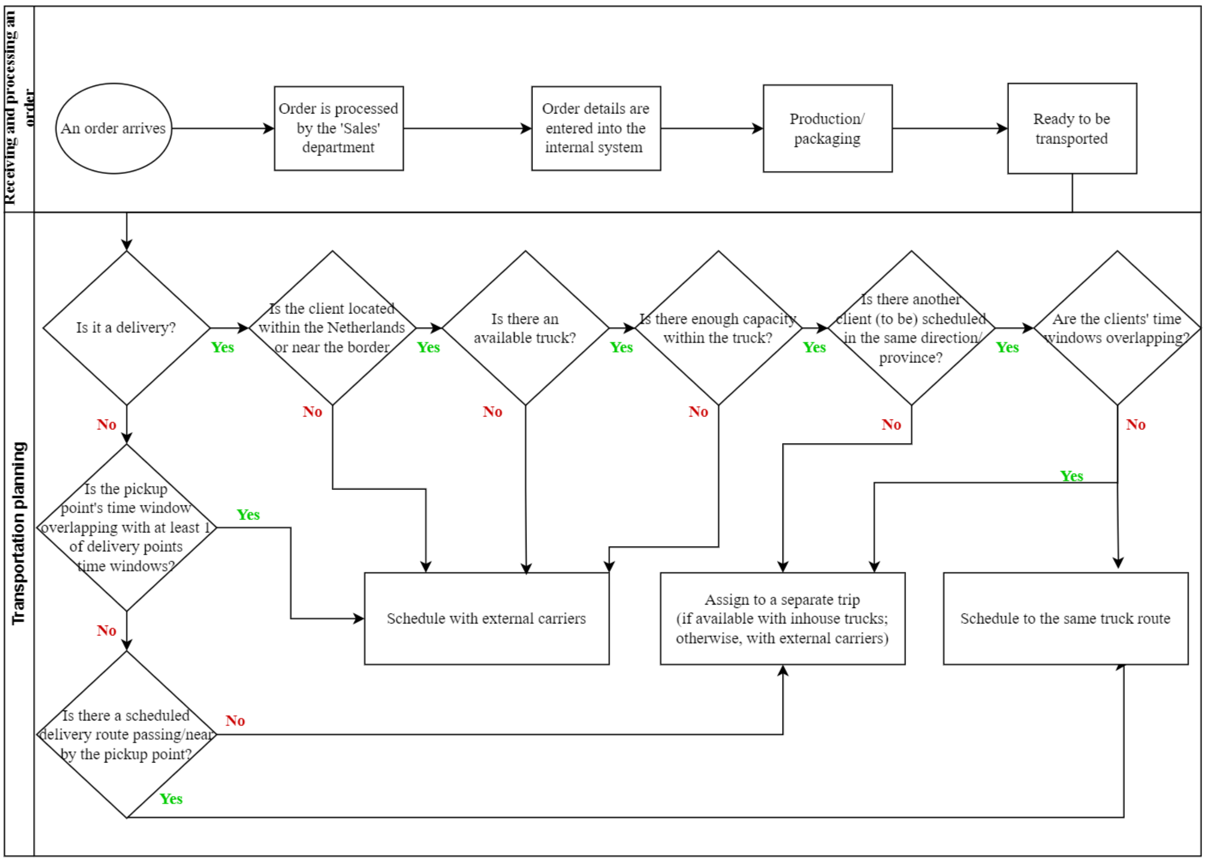

Figure 4. Meilink's transportation planning decision-making diagram.

The process starts when the sales department receives an order and records order-specific characteristics like delivery date and time, single product volume or batch size, and processes it further the chain. In case the order is an item requiring packaging at the Borculo office, external transportation arrangements are made by the customer. On rare occasions, customers may directly request Meilink's transportation services, where the trucks have to both collect the product from and deliver it to the client. Those requests are recorded by the sales department within the system.

Once the order has been processed – either produced or packaged, and it is prepared for transportation, the route planning can begin. Within the logistics department, an internal program is employed that provides an overview of all orders. This includes whether it is pickup or delivery, the type of item, the number of items ordered, the department from which it is ordered (box factory, carton factory, packing hall), current status (arrived, in process, ready, shipped), requested delivery date, client contact information, delivery address, available vehicles, time-windows, product dimensions and weight. The logistics department assigns clients who are located outside the Netherlands to external carriers and the rest are planned for a visit by Meilink's private fleet or common carrier.

2.1.2 Clustering and route optimization

Throughout the week, Meilink Borculo B.V. schedules its trucks to different regions. During the planning process, the primary objective is to cluster together as many clients as possible based on the region and city proximity. The planning process does not rely on any route optimization software or models; rather, everything is done based on employee knowledge and experience. The logistics team checks whether all the products fit within one truck. Additionally, if two or more clients have overlapping and short time windows (e.g., 30 minutes), then they are not included in the same tour. If clients are situated in close proximity, their time windows do not overlap, or those are not too short, and their items can fit within a single truck, then the customers are scheduled in the same tour. In all other cases, clients are assigned either to another available Meilink truck or to external carriers.

When Meilink has to collect goods from clients or suppliers, the decision-making process is outlined in Figure 4. Several factors affect the planning. First, if the pick-up time window overlaps with the delivery time windows, external trucks are scheduled. This is because Meilink's trucks are typically assigned to the delivery first, and the logistics team cannot recalculate the available capacity of the trucks along the delivery route. When the time windows do not overlap and trucks are already scheduled for deliveries on a route that passes by the pickup points, goods are collected on the return trip to the depot, after serving each delivery point. Important to note that Meilink's in-house trucks are primarily used for delivery to clients, and pickups are performed only after deliveries are completed.

In both pickup and delivery requests, the truck drivers receive a list of points which they have to visit and the according time to be there. The transportation department does not specify explicitly the order of visiting the customer or the routes those trucks should undertake.

Meilink trucks can undertake multiple tours per day, provided the drivers' working hours are considered. If they have to make two or more trips on the same day, the drivers are given the closest site first, followed by the farthest. This is done to avoid delays and to adhere to the established time constraints. Drivers are also getting exhausted and want to return home early, therefore they will refuse to undertake two or more excursions if they are assigned to the furthest site initially.

2.1.3 In-house truck ownership vs. utilizing common carriers

Research on the transportation processes of manufacturing organizations discovered that outsourcing transport is preferred over in-house transport, as the latter presents challenges linked to expenses associated with owning and maintaining cars (Bartalero et al., 2020). Furthermore, using third-party logistics may lower environmental expenses since trucks are frequently used more efficiently (Tezuka, 2011). The situation becomes even more difficult since the Dutch government has promised to lower carbon emissions to net zero by 2050, which means companies must take responsibility for the prices of their transportation services and CO2 emissions and endeavour to reduce them.

Outsourced carriers, on the other hand, cannot completely replace in-house transportation, and a balanced solution between outsourcing and owning transportation is required for maximum efficiency (Stojanović, 2017). Meilink uses its vehicles or outsources third-party logistics (3PL) providers to carry to its demand points. However, there are no strict criteria for making a choice between those two options. Also, there is no selection of demand points in the Netherlands, served only by external carriers. Currently, external trucks are used only in the following situations:

- The order must be sent outside of the Netherlands.
- The Meilink trucks are all booked for the day and are unable to serve the whole demand.
- The package dimensions do not fit in Meilink's vehicles or there is not enough capacity available.
- The Meilink trucks drive to a certain location, but an order in the opposite direction must also be satisfied.
- Several orders must be served within the same time window, yet the distance between them is considerable.
- Meilink's truck drivers have off-day(s).

Meilink engages third-party carriers for transportation services in the aforementioned instances. Meilink does not do the route planning of the externally outsourced trucks. They only provide a list of customers, who have to be visited within the day, together with additional characteristics like time-windows and delivery points. The external service provider is then responsible for planning the routes and determining how many trucks are required for the provided client list.

2.2 Characteristics

To develop a comprehensive understanding of Meilink's system, a closer look should be taken into the key aspects that define the logistics process. This includes understanding when demand occurs and how it is distributed, the time windows that must be adhered to, and Meilink's vehicles and their capacities. Those aspects will be explored in the following sections.

2.2.1 Type of orders

Meilink makes trips to its clients every day of the week, except for Saturdays and Sundays. Customers place a wide range of orders, varying in quantity and size. The demand is mostly deterministic, i.e., the transportation department has the information before the route planning begins. Therefore, Meilink can determine how often the client must be visited within a particular time period, and plan the transportation (as described in Section 2.1).

The order requests at Meilink can be categorized into two types: deliveries of goods from the depot to customers and pickups of goods from both customers and suppliers to the depot. Based on the available data for the period between September 2022 and September 2023, Meilink Borculo B.V. handled deliveries to clients 81.27% of the time. In contrast, it has been involved in picking up goods from clients or suppliers in 18.73% of the total orders.

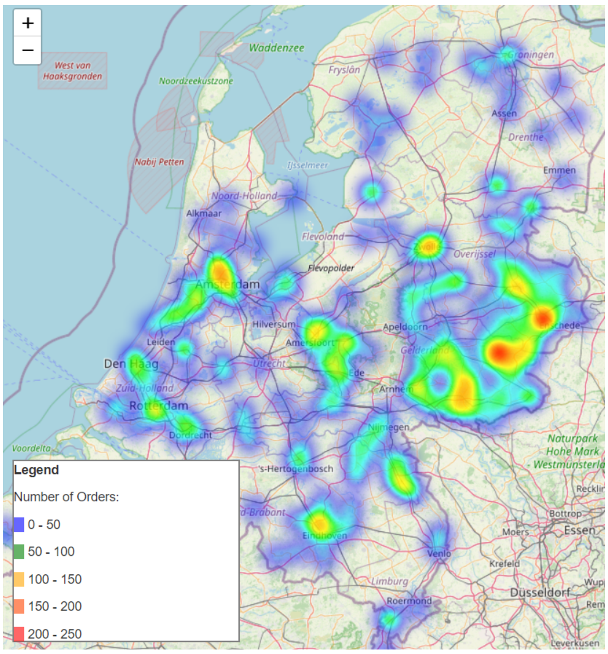

Figure 5. Heatmap of Meilink Borculo's number of transported orders in 2022-23.

2.2.1.1 Delivery to clients

The largest proportion of transportation is linked with delivery of orders (81.27%). Meilink provides transportation services to all of its customers and delivers goods upon their request. The trucks are scheduled for visiting customers nodes from Monday to Friday. Since the deliveries abroad are executed by Meilink's subsidiary, those are out of the scope of this thesis. Therefore, only the demand nodes within the Netherlands and nearby the borders are considered.

In Figure 5 a heatmap can be observed displaying the frequency of order deliveries to different regions across the Netherlands from January 2022 to September 2023. The provinces of Overijssel, Gelderland, Utrecht, and South Holland are visited the most, indicating higher demand from clients in these areas. There, some customers received deliveries more than 200 times throughout the specified time period. Significantly lower is the number of tours made to Limburg, Zeeland, Friesland, Groningen, and Drenthe.

Meilink transports goods also to both Belgium and Germany. Because of toll taxes, those destinations are visited rarely by Meilink trucks, and usually, the products are delivered just near the border. External carriers are responsible for servicing some of the more remote places from Borculo and often they make multiple trips per day.

2.2.1.2 Pickup requests

When products need to be packaged at the Borculo office, the client is generally responsible to transport them to the depot. In some situations, where clients may not be able to arrange transportation, they request that Meilink's trucks pick up the products and bring them to the packaging location. The requests are typically known in advance when the client places an order.

Meilink's usual practice is that suppliers are responsible for arranging transportation and delivering orders using their own vehicles. There are occasional exceptions, such as when the items must be delivered quickly, and the supplier cannot organize transportation. Furthermore, there are some suppliers who do not handle transportation making it Meilink's responsibility to collect the ordered goods. Orders to suppliers are placed irregularly, and the quantity of commodities that Meilink must collect from them is certain once the order is placed.

2.2.2 Time windows

When consumers place orders, they specify a preferred delivery date, expecting their purchases to arrive within the agreed-upon time frame. These are subject to vary with each order, depending on clients' preferences.

The time windows for the transportation planning from September 2022 to September 2023 are presented in Table 1. The table gives an overview of the form of transportation, which may involve delivering to customers or picking up from clients or suppliers. Most clients have broad time spans, such as 8:00 AM to 4:30 PM. However, some clients prefer a specific delivery time, like 8:00 AM, with a 30-minute deviation allowance, creating a shorter time window of 8:00-8:30 AM.

Meilink endeavours to comply with client requirements, even if it means dispatching multiple trucks to the same region or town to fulfil orders for two different clients who share the same delivery date and time preference. These overlapping time windows indirectly increase transportation expenses since both vehicles may return empty after making their deliveries.

Type	Days	Time window	Visits
Delivery	Mon-Fri	7:30-8:00	14
Delivery	Mon-Fri	8:30-9:00	2
Delivery	Mon-Fri	9:30-10:00	2
Delivery	Mon-Fri	10:30-11:00	8
Delivery	Mon-Fri	8:00-13:00	6
Delivery	Mon-Fri	12:30-13:00	2
Delivery/Pick-up	Mon-Fri	8:00-16:00	2154
Delivery/Pick-up	Mon-Fri	8:00-16:30	364
Delivery/Pick-up	Mon-Fri	8:00-17:00	450
Delivery	Mon-Thur	7:30-16:45	4
Delivery	Friday	7:30-12:15	4
Delivery	Mon-Thur	6:00-12:00	2
Delivery	Mon-Fri	7:00-15:00	3
Pick-up	Mon-Fri	7:30-8:00	34
Pick-up	Mon-Fri	10:30-11:00	2
Pick-up	Mon-Fri	11:30-12:00	14
Pick-up	Mon-Fri	8:00-13:00	7
Pick-up	Mon-Fri	8:00-14:00	3
Pick-up	Mon-Fri	8:00-16:00	2

Table 1. Overview of customer and supplier time windows with the number of visits required for the period September 2022 - September 2023.

2.2.3 Vehicles

Meilink Borculo owns a fleet of vehicles that it frequently utilizes for product deliveries. This fleet includes four lorries, six trailers, and one box truck, each designed for specific cargo dimensions. Table 2 shows each type of vehicle and its accompanying cargo dimensions. The box truck is smaller than the trailers. Because the trailers must be attached to a lorry, the six trailers can never be utilized simultaneously. Furthermore, all of the trailers have identical cargo space measurements. Because the materials to be carried are not standardized, the truck dimensions are a critical part of Meilink's transportation.

Common carriers come in a variety of sizes, including massive vehicles like mega-trailers. Because the external transportation company's fleet size is not specified, it is presumed that they can arrange as many trucks as Meilink need per day.

Vehicle model	Type	Dimensions cargo space
Volvo	Box truck	720x240x230cm
Mercedes	Lorry	-
Mercedes	Lorry	-
Mercedes	Lorry	-
Mercedes	Lorry	-
10, vlak. Schmitz	Trailer	1360x240x250 cm
9, zwanehals, Van Hool	Trailer	1360x240x250 cm
11, vlak, Schmitz	Trailer	1360x240x250 cm
12,vlak,Schmitz	Trailer	1360x240x250 cm
7, vlak	Trailer	1360x240x250 cm
8,vlak	Trailer	1360x240x250 cm

Table 2. Types of vehicles currently owned by the company and their dimensions.

2.3 Current transportation cost structure

Meilink Borculo primarily relies on its in-house vehicles as a preferred choice for transporting its products. The daily distance covered by these vehicles is recorded and the expenditures made are evaluated. Clients are charged for the delivery services, generating income for the company. Although clients are charged for delivery services, Meilink's actual transportation expenses, including driver salaries, fuel, maintenance, and external transport costs, are subtracted from this income.

In 2022, Meilink Borculo B.V.'s total labour costs, including truck driver wages, amounted to approximately 40% of the total transportation costs. Private fleet expenses amounted to 21%, with 74% of those for fuel, and the remainder for maintenance, and insurance. Additional costs, such as depreciation, brought the total for the private fleet to 66% from the total transportation costs. The private vehicles travelled approximately 341,000 km in 2022.

Meilink also uses external service providers for transporting its goods. The total costs involved with those external providers are 34% for the year 2022. External carriers charge a fixed fees based on the destination points and not on the total distance travelled. The total distance covered by the external carriers is approximately 237 500 km for the same year.

In summary, Meilink invests a significant amount of money in its transportation services amounting to a total of €683,200[1] for the year 2022. Despite charging clients for transportation, the goal is to reduce the total transportation costs, as they have been substantially high over the past years, which may eventually result in a net loss for the company in the upcoming periods.

2.4 Key performance indicators

The primary Key Performance Indicator (KPI) is the cost associated with transporting items to and from the clients. The costs described in Section 2.3 should be decreased as much as feasible so that the company can keep providing efficient transportation services. Moreover, this KPI is key in resolving the action problem, provided by the company (Section 1.2.1). Optimizing the costs and decreasing the total expenditures from the transportation services can contribute to the company's profitability, ensuring that transportation services remain competitive.

Another critical KPI to examine is the total distance travelled by the trucks. Within the total distance, all the pick-ups to and from clients and suppliers are considered. Beyond the financial aspect, this KPI has an environmental significance to Meilink, as reducing carbon emissions is one of their current goals. Moreover, decreasing the total distance travelled usually leads to lower fuel consumption resulting in a reduction in fuel-related expenses. By managing this KPI effectively, Meilink can decrease its carbon footprint, together with decreasing its fuel consumption expenses and sometimes even the total working hours of the drivers.

The last KPI is the number of trucks assigned to a trip each day. Employee wages are one of the most significant cost components for the company. Optimizing vehicle usage by deploying enough trucks is crucial in decreasing the total labour costs. Employees also require adequate rest days, therefore optimizing the number of vehicles might potentially result in a more ideal work schedule for them. It may also reduce maintenance expenses, as the fewer vehicles there are, the cheaper the total maintenance expenditures. Consequently, this KPI has both financial and human factor aspects, aiming for efficient operations for the company.

2.5 Requirements and limitations

There are several requirements and limitations, which have to be considered when formulating a solution for the identified transportation problem. First, the clients' specified time windows have to be satisfied, even if it necessitates deploying two vehicles on the same route. When there is a need for both client deliveries and collection of raw materials from suppliers on the same day, the priority is given to the client. In case of a pickup, the priority is also given to the client first.

Each truck can make more than one tour per day and there is no constraint on the maximum number of tours per vehicle, as long as the working hours of the company are not exceeded. There are no limitations related to the number of trucks used per day, thus all five trucks can be used simultaneously. Each truck driver can work at most 9 hours a day and only one driver is assigned per truck. Thus, the total time a truck can be used per day is 9 hours. At the end of each tour, trucks must return back to the depot. Tours cannot be planned on Saturdays and Sundays. The route plannings should be made for each day of the year. The model should execute the route planning with an hour.

When planning a route, the vehicle capacity should be considered, together with the order size and dimensions. A limitation is that some of the products are transported folded, thus, their dimensions are different than the one within the system and have to be estimated. A further limitation is that

[1] This cost does not represent the actual costs of the business. The total costs are multiplied by a random non-integer number to maintain the confidentiality of the business. All subsequent costs will also be multiplied by the same random number.

the products are not standard in size, which may not fit in the private fleet. In such situations, bigger vehicles have to be sourced externally.

Lastly, the thesis should mainly focus only on Meilink Borculo, and the orders received there. The demand received from other locations is out of the scope of this report.

2.6 Conclusion

This chapter seeked for an answer to the posed research question ***What is the current transportation system o ered by Meilink Borculo B.V.?***. It became evident that the company's transportation system involves daily transportation services, Monday to Friday, with a complex process that lacks a dedicated route planning system. Orders are processed and planned manually, with decisions influenced by factors such as delivery date and time, product dimensions, and available vehicles. The private vehicles are clustered to serve within a single region, with the idea to optimize routes, primarily based on employee knowledge and experience. Meilink operates a fleet of vehicles, including four lorries, six trailers and a box truck, and occasionally utilizes external carriers for transportation, primarily for international deliveries.

The transportation planning decisions at Meilink Borculo B.V. are influenced by various criteria and factors, including delivery time windows specified by clients, product dimensions, and the capacity of Meilink's fleet. Moreover, the company has to consider client preferences for specific delivery times and the need to comply with overlapping time windows, which altogether add complexity to route planning. Additionally, the type and size of orders, availability of vehicles, and driver working hours are critical considerations in decision-making.

Meilink Borculo B.V. invests significantly in transportation services, with transportation costs amounting to €683,200 for the year 2022. These costs include labor expenses, fuel, maintenance, and external transport charges. Labor costs constitute approximately 40% of the total internal transportation expenses, while private fleet expenses, including fuel and maintenance, amount to 66%. External service providers account for 34% of the total transportation costs. Despite charging clients for transportation services, the company aims to reduce costs, considering them as substantially high and potentially leading to a net loss.

Key Performance Indicators (KPIs) for Meilink's logistics process include total transportation costs, total distance traveled by trucks, and the number of trucks assigned to trips each day. These KPIs are crucial for optimizing operations, reducing costs, and ensuring efficient resource allocation. Constraints and requirements include adhering to client time windows, considering vehicle capacity and order dimensions, and complying with driver working hours and vehicles' availability. The route planning should be executed within an hour, considering vehicle capacity and non-standardized product sizes. Additionally, the focus should mainly be on Meilink Borculo and its orders, with demand from other locations considered out of scope.

3 Literature review

Meilink Borculo B.V. has a routing problem, which requires optimization of their current transportation process by minimizing transportation costs. Therefore, the best model approach and solution methods for routing problems with Meilink's transportation process characteristics will be investigated in the following sections. Namely, the research question *Based on the literature, what are the applicable techniques and methods for modelling and solving a vehicle routing problem?* will be addressed. In Section 3.1 the general VRP is introduced, followed by an extensive literature review on the vehicle routing problem (VRP) variants, which align with the characteristics of Meilink's transportation process (Sections 3.2-3.7). The rest of the chapter provides an overview of the solution approaches, that best fit the described VRP variants (Section 3.8).

3.1 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a complex logistical challenge that involves determining the most efficient routes for a fleet of vehicles to serve a set of customers. The VRP usually has the goal to minimize costs or maximize efficiency. The VRP found its grounds back in 1959 when Dantzig and Ramser introduced the "Truck Dispatching Problem", which is a generalization of the "Traveling-Salesman Problem". The authors aimed to find the optimal routing of homogeneous gasoline delivery trucks between the terminal and multiple service stations (Dantzig and Ramser, 1959). In the later years, Clarke and Wright (1964) generalized the Truck Dispatching Problem by linearizing it and formulated the first Vehicle Routing Problem, which includes heterogenous vehicles with varying capacities, leaving from a central depot and serving geographically dispersed customers. The Vehicle Routing Problem is a non-deterministic polynomial (NP)-hard problem, meaning, finding a solution for this problem is exceptionally challenging and time-consuming (Alridha et al., 2021). For those problems, there is no one quick way to find the solution, rather many possibilities exist especially when the problem gets bigger. One such solution method is Clarke and Wright's savings algorithm for solving VRP, which is a greedy heuristic because it involves step-by-step decisions for achieving a local optimization without searching for or guaranteeing a globally optimal solution (Clarke and Wright, 1964).

1 Type of Study	2.6.3 Mix of both	3.8.2 Uncapacitated vehicles/unlimited capacity
1.1 Theory	2.7 Time horizon	3.9 Vehicle homogeneity (Capacity)
1.2 Applied methods	2.7.1 Single period	3.9.1 Similar vehicles
1.2.1 Exact methods	2.7.2 Multi period	3.9.2 Load-specific vehicles
1.2.2 Classical Heuristics	2.8 Backhauls	3.9.3 Heterogeneous vehicles
1.2.3 Metaheuristics	2.8.1 Nodes request simultaneous pickups and deliveries	3.9.4 Customer-specific vehicles
1.2.4 Hyper-heuristics	2.8.2 Nodes request either linehaul or backhaul service, but not both	3.10 Travel time
1.2.5 Machine learnig	2.9 Node/Arc covering constraints	3.10.1 Deterministic
1.2.6 Simulation	2.9.1 Precedence and coupling constraints	3.10.2 Function dependent
1.2.7 Real time solution methods	2.9.2 Subset covering constraints	3.10.3 Stochastic
1.3 Implementation documented	2.9.3 Recourse allowed	3.10.4 Unknown
1.4 Survey, review or meta-research	**3 Problem Physical Characteristics**	3.11 Object
2 Scenario Characteristics	3.1 Transportation network design	3.11.1 Single
2.1 Number of stops on route	3.3.1 Directed network	3.11.2 combinations
2.1.1 Known (deterministic)	3.3.2 Undirected network	**4 Information Characteristics**
2.1.2 Partially known,partially probabilistic	3.2 Location of addresses (customers)	4.1 Evolution of information
2.2 Load splitting constraint	3.2.1 Customers on nodes	4.1.1 Static
2.2.1 Splitting allowed	3.2.2 Arc routing instances	4.1.2 Partially dynamic
2.2.2 Splitting not allowed	3.4 Number of points of origin	4.2 Quality of information
2.3 Customer service demand quantity	3.4.1 Single origin	4.2.1 Known (Deterministic)
2.3.1 Deterministic	3.4.2 Multiple origins	4.2.2 Stochastic
2.3.2 Stochastic	3.5 Number of points of loading/unloading facilities (	4.2.3 Forecasted
2.3.3 Unknown1	3.5.1 Single depot	4.2.4 Unknown (Real-time)
2.4 Request times of new customers	3.5.2 Multiple depots	4.3 Availability of information
2.4.1 Deterministic	3.6 Time window type	4.3.1 Local
2.4.2 Stochastic	3.6.1 Restriction on customers	4.3.2 Global
2.4.3 Unknown1	3.6.2 Restriction on roads	4.4 Processing of information
2.5 On site service/waiting times	3.6.3 Restriction on depot/hubs	4.4.1 Centralized
2.5.1 Deterministic	3.6.4 Restriction on drivers/vehicle	4.4.2 Decentralized
2.5.2 Time dependent	3.7 Number of vehicles	**5 Data Characteristics**
2.5.3 Vehicle type dependent	3.7.1 Single vehicle	5.1 Data Used
2.5.4 Stochastic	3.7.2 Limited number of vehicles	5.1.1 Real world data
2.5.5 Unknown	3.7.3 Unlimited number of vehicles	5.1.2 Synthetic data
2.6 Time window structure	3.8 Capacity consideration	5.1.3 Both real and synthetic
2.6.1 Soft time windows	3.8.1 Capacitated vehicles/limited capacity	5.2 No data used
2.6.2 Strict time windows	3.8.2 Uncapacitated vehicles/unlimited capacity	

Figure 6. Taxonomy of the VRP literature (Ni and Tang, 2023). The characteristics which match the Meilink's transportation process are highlighted in blue.

Since the development of the first Vehicle Routing Problem, its variations have been evolving and currently, there are various types of models proposed in the literature. Ni and Tang (2023) created a three-level classification of the VRP models. The overview of Ni and Tang (2023) VRP taxonomy is provided in Figure 6. The VRPs usually combine several features from the ones presented in Figure 6 to match the problem they are trying to solve.

Within the deterministic VRPs, the delivery requests from customers are known and are only received during the execution of the vehicle route (Ni and Tang, 2023). This also applies that the number of stops on a route, which is the customer nodes in a VRP, is known. Moreover, the vehicle routing problem with time windows (VRPTW) adds the additional constraint that customer nodes have time constraints, within which they should be visited (Ni and Tang, 2023). A time window is associated with each customer node, within which loading or unloading the goods has to be completed Ombuki (2006). Within the VRPTW, the vehicles have to arrive at the customer node between the earliest and latest time specified (Solomon, 1987). When these time windows are soft, the problem is relaxed to VRP with additional penalty costs imposed (Miranda and Conceição, 2016).

The capacitated vehicle routing problem (CVRP) is one of the most researched and utilized variants of the VRP. This revolves around the idea that all customers have a deterministic demand, which has to be satisfied through deliveries with vehicles having limited capacity (Toth and Vigo, 2002). A generalization of the ordinary capacitated VRP is the vehicle routing problem with backhauls (VRPB) where the goods are both delivered to the customers, but also some goods are brought back to the depot. This problem is also called the pickup and delivery problem, where the vehicles can travel from the depot to nodes, which request either linehaul or backhaul service (Ropke and Pisinger, 2006).

The taxonomy in Figure 6, however, does not consider all variations of the backhaul, such as the Vehicle Routing Problem with Divisible Delivery and Pickup (VRPDDP), where each customer can be visited for both pickup and delivery operations by a single or separate visit (Parragh et al., 2008). Also, goods can be partially picked up within the VRPDDP, allowing for order splitting.

The Split Delivery VRP is another variant of the CVRP, which allows the delivery of goods to a customer to be divided between multiple vehicles (Wassan and Nagy, 2014). As a result, instead of being visited once and receiving the entire demand at once, the client's demand may be supplied across multiple smaller deliveries by separate vehicles.

Although the VRP with private fleet and common carrier (VRPPFCC) is not mentioned in Ni and Tang's classification, it is relevant to the problem of Meilink discussed in Section 2. The VRPPFCC addresses the issue that sometimes the capacity of inhouse vehicle may not be enough to serve all demand nodes (Higino et al., 2018). Also, it may be that some vehicles cannot access certain locations. In both cases, outsourcing vehicles externally is the solution companies utilize. However, the VRPPFCC looks into the demand nodes, which have to be strictly satisfied by either private fleet or by common carriers (Bolduc et al., 2008).

Another characteristic that is not present in the taxonomy of Ni and Tang is distinguishing between single and multi-trip problems. Multi-Trip Vehicle Routing Problem's (MTVRP) primary objective is to optimize the routing of a fleet of vehicles to serve a set of customers over multiple trips (Brandão and Mercer, 1998), compared to the single-trip VRP, where each vehicle completes only a single route. Within the MTVRP, vehicles are allowed to return multiple times to the depot for loading and unloading and continue servicing additional customer nodes until their capacity or time limits are reached (Brandão and Mercer, 1998).

In the following subsection, the mathematical model of the general VRP is provided. In each of the later models dealing with variants of the VRP, the specific changes and adaptations in this general model are provided. This is done for the VRPDDP, VRPTW, MTVRP, VRPPFCC, and SDVRP. These specific variants of a VRP have the characteristics of the problem described in Section 2.

3.1.1 Mathematical model

The general Vehicle Routing Problem is presented in its simplified version by Christofides et al. (1981). The problem considers a case with geographically dispersed customers with known demand, who must be served by a fleet of vehicles from a central depot with a cost minimization objective. This formulation is presented below.

Parameters:

- N: Set of customers, indexed by i and j, including depot node 0.
- M: Set of vehicles, indexed by k.

- Q_k: Capacity of vehicle k.
- q_i: Demand of customer i.
- c_{ij}: Cost of the least cost path from vertex x_i to vertex x_j.

Decision variables:

- x_{ijk}: Binary variable, where

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from customer } i \text{ to customer } j, \\ 0, & \text{otherwise.} \end{cases}$$

- y_i: Sequence indicator variable, where

$$y_i = \begin{cases} 1, & \text{if customer } i \text{ is visited in the route,} \\ 0, & \text{otherwise.} \end{cases}$$

Model:

$$\min z = \sum_{i=0}^{N} \sum_{j=0}^{N} c_{ij} \sum_{k=1}^{M} x_{ijk} \tag{3.1}$$

Subject to

$$\sum_{i=0}^{N} \sum_{k=1}^{M} x_{ijk} = 1 \quad \forall j = 1, \ldots, N \tag{3.2}$$

$$\sum_{i=0}^{N} x_{ipk} - \sum_{j=0}^{N} x_{pjk} = 0 \quad \forall k = 1, \ldots, M, p = 0, \ldots, N \tag{3.3}$$

$$\sum_{i=1}^{N} q_i \sum_{j=0}^{N} x_{ijk} \leq Q_k \quad \forall k = 1, \ldots, M \tag{3.4}$$

$$\sum_{k=0}^{M} \sum_{j=0}^{N} x_{0jk} \leq |M| \tag{3.5}$$

$$\sum_{j=1}^{N} x_{0jk} = 1 \quad \forall k = 1, \ldots, M \tag{3.6}$$

$$y_i - y_j + N \sum_{k=1}^{M} x_{ijk} \leq N - 1 \quad \forall i \neq j = 1, \ldots, N \tag{3.7}$$

$$x_{ijk}, y_i \in \{0, 1\} \quad \forall i, j, k \tag{3.8}$$

The objective function (3.1) minimizes the total costs involved with this transportation problem. Constraint (3.2) states that each customer should be visited once, and the flow conservation constraint (3.3). Constraint (3.5) ensures that the maximum number of vehicles is not exceeded and (3.6) each vehicle may be used exactly once. The sub-tour elimination constraint (3.7) also ensures that each route passes through the depot. This model is also a Capacitated Vehicle Routing Problem (CVRP), as in constraint (3.4), the total capacity of each vehicle (heterogenous) may not be exceeded. Constraint (3.8) ensures the variables are binary.

3.2 VRP with backhauls

In the general pickup and delivery VRP, there are two main subclasses: VRP with backhauls (VRPB) and VRP pickup and delivery, involving transportation to and from a depot and between customer nodes, respectively (Parragh et al., 2008). The pickup and delivery VRP are outside the scope of the thesis, as

Meilink does not distribute products between customers. Instead, the company operates from a depot and focuses on transportation to and from it.

Figure 8 illustrates the four subclasses of the VRP with Backhauls: Vehicle Routing Problem with Clustered Backhauls (VRPCB), VRP with Mixed Linehaul and Backhaul (VRPMB), VRP with Divisible Delivery and Pickup (VRPDDP) and VRP with Simultaneous Delivery and Pickup (VRPSDP). Each of these is shown with the same configuration, namely triangles representing the pickup goods, squares denoting depots, and circles signifying delivery goods (Polat, 2017).

In both VRPCB and VRPMB, a customer requires either a delivery or a pickup but not both. The VRPCB (Figure 7, a) is clustering the linehaul and backhaul customers, prioritizing delivery nodes before pickups (Nagy et al., 2015). The VRPMB (Figure 7, b), on the other hand, does not require the clustering constraint, rather it focuses on restricting the linehauls and backhauls based on the vehicle capacity (Parragh et al., 2008).

In the other two classes, VRPDDP and VRPSDP, delivery customers can also require pickups. The VRPDDP (Figure 7, d) follows the same logic as VRPMB, with the only difference being that all customers are associated with both linehaul and backhaul quantities, allowing each customer to be visited twice, once for delivery and once for pickup service (Polat, 2017). The VRPSDP (Figure 7, c) does not allow each customer to be modelled as two separate entities, therefore, each client can be visited only once for a simultaneous delivery and pickup (Nagy et al., 2015). In contrast, the VRPDDP allows each client to be a delivery node, a pickup node, or both at the same time, allowing the deliveries and pickups to be separated by visiting the customer twice or, if the capacity of the vehicle allows, the deliveries and pickups to be executed simultaneously during the same visit.

In Meilink's problem scenario, each customer serves as both a pickup and a delivery node, which means that some customers may need to be visited twice. This unique classification aligns with the VRP with Divisible Delivery and Pickup (VRPDDP). Unlike the other three classifications, VRPDDP permits load splitting (Polat, 2017), which is a feature necessary for modelling the Meilink scenario. Consequently, the mathematical model for VRPDDP will be presented in the following section.

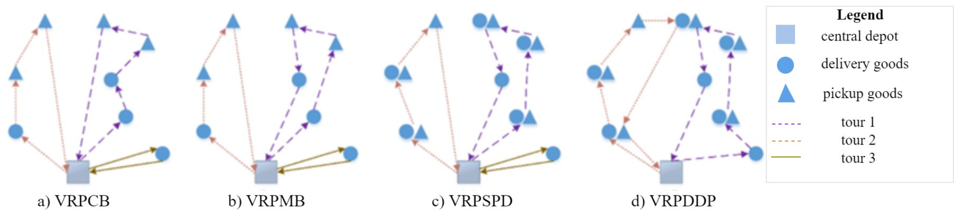

Figure 7. VRP with Backhaul subclasses (Polat, 2017, p.2)

3.2.1 Mathematical model for VRPDDP

Nagy et al. (2015) developed an integer linear programming model for the Vehicle Routing Problem with Divisible Delivery and Pickup(VRPDDP). This model allows more than one visit to be made to the customer: one for pickup and one for delivery. It assumes that all of the delivery to a customer is made in a single visit, and it is the same case with pickup. The parameters and decision variables that have not yet been specified in Section 3.1 have been given below.

Parameters:

- $D = f0g$: the set of depots (consisting of a single depot)
- $B = fn + 1, n + 2, \ldots, 2ng$: the set of backhaul customers (backhaul $n + i$, is a copy of linehaul i)
- $V = D [N [B$: the set of all vertices

Decision variables:

- R_{ij} = the amount of delivery goods on board on arc ij
- P_{ij} = the amount of pickup goods on board on arc ij

Model:

$$\min z = \sum_{i=0}^{V}\sum_{j=0}^{V} c_{ij} \sum_{k=1}^{M} x_{ijk} \tag{3.1a}$$

Subject to:

$$\sum_{i=0}^{V}\sum_{k=1}^{M} x_{ijk} = 1 \quad \forall j \in N \cup B \tag{3.2a}$$

$$\sum_{i=0}^{V}\sum_{k=1}^{M} x_{jik} = 1 \quad \forall j \in N \cup B \tag{3.9}$$

$$\sum_{i=0}^{V} R_{ij} - q_j = \sum_{i=0}^{V} R_{ji} \quad \forall j \in N \tag{3.10}$$

$$\sum_{i=0}^{V} R_{ij} = \sum_{i=0}^{V} R_{ji} \quad \forall j \in B \tag{3.11}$$

$$\sum_{i=0}^{V} P_{ij} = \sum_{i=0}^{V} P_{ji} \quad \forall j \in N \tag{3.12}$$

$$\sum_{i=0}^{V} P_{ij} + q_j = \sum_{i=0}^{V} P_{ji} \quad \forall j \in B \tag{3.13}$$

$$\sum_{i \in L \cup B} P_{0i} = 0 \tag{3.14}$$

$$\sum_{i \in L \cup B} R_{i0} = 0 \tag{3.15}$$

$$R_{ij} + P_{ij} \leq Q_k x_{ijk} \quad \forall i \in V, j \in V, k \in M \tag{3.16}$$

$$\sum_{k=0}^{M}\sum_{j=1}^{V} x_{0jk} \leq |M| \tag{3.5a}$$

$$Q_k \sum_{i=1}^{V} x_{0ik} \geq \sum_{i=1}^{N} q_i \quad \forall k \in M \tag{3.17}$$

$$x_{iik} = 0 \quad \forall i \in V, k \in M \tag{3.18}$$

$$x_{(n+i)ik} = 0 \quad \forall i \in N, k \in M \tag{3.19}$$

$$x_{ijk} + x_{jik} \leq 1 \quad \forall i, j \in N \cup B, k \in M \tag{3.20}$$

$$R_{ij}, P_{ij} \geq 0 \quad \forall i \in V, j \in V \tag{3.21}$$

In this model, the objective function (3.1a) is slightly changed, compared to (3.1), by including also the backhaul nodes in the summation of costs. The same applies to constraints (3.2a) and (3.5a). Equation (3.9) ensures that each customer is served once. Constraints (3.3) and (3.7) are replaced by (3.10) -(3.13), which are flow conservation constraints, also ensuring that the subtours are eliminated. Constraints (3.14) -(3.15) are ensuring that vehicles start with zero pickup load and finish with zero delivery load. The vehicle capacity limit is enforced in (3.16). Constraint (3.4) is replaced by (3.17) sets the minimum number of vehicles required, together with vehicle total capacity. Equation (3.18) ensures there are no loops, and (3.19) that there is no arc from a backhaul to its corresponding linehaul. Constraint (3.20) eliminates subtours for sets of two customers. Constraint (3.21) sets the variables as non-negative.

3.3 VRP with time windows

The Vehicle Routing Problem with Time Windows (VRPTW) is a combination of routing and scheduling problem in which vehicles must reach at their destination within the time window provided (Ni and Tang, 2023). These time frames might be classified as soft, hard, or mixed. The hard time window restriction requires customers to refuse items if delivery exceeds their time window (Kallehauge et al., 2005). When a vehicle arrives earlier than expected, it must wait until the earliest service time. With flexible time windows, the vehicle might arrive at a client location earlier or later, and a penalty for lateness is incurred (Chen et al., 2007). Aside from the client nodes, there may be time window constraints for vehicle usage as well as the depot's opening and closing hours.

3.3.1 Mathematical model for VRPTW

The model for the VRP with time windows is adapted from the formulation of Kallehauge et al. (2005). The model is adapted to the parameters and decision variables in the previous sections. Only parameters and constraints different from those in Section 3.1 are presented.

Parameters:

- a_i = start time of customer i's time window
- b_i = end time of customer i's time window
- t_{ij} = travelling time from customer i to customer j (it may include the service time at customer i)

Decision variables:

- s_{ik} = the time vehicle k starts to service customer i

Constraints:

$$\sum_{i=1}^{V} x_{i0k} = 1 \quad , k \in M \tag{3.22}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall i, j \in V, k \in M \tag{3.23}$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in V, k \in M \tag{3.24}$$

$$s_{ik} + t_{ij} - \text{BigM}_{ij}(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in V, k \in M \tag{3.23a}$$

First, this model adds a limitation that each route should end at the depot (3.22). Constraint (3.23) establishes the relationship between the vehicle departure time from a customer and its immediate successor, and (3.24) ensures the customer time windows are satisfied. Although constraint (3.23) is not linear, it can be linearised through equation (3.23a).

3.4 Split delivery VRP

The Split Delivery (or Split-Load) Vehicle Routing Problem (SDVRP) divides the customer demand into many shipments that may be delivered via different routes and vehicles (Ni and Tang, 2023). Instead of one, several vehicles can visit a single consumer, or with the same vehicle in multiple trips. Dror and Trudeau (1989) are the first to propose the Split Delivery VRP. In their paper it was shown that because the SDVRP is a more flexible variant of the VRP, it can result in considerable savings in overall distance and number of vehicles assigned. Later, a k-SDVRP model was formulated, in which each tour begins at the depot, distributes k units to clients, and returns to the depot (Archetti et al., 2006).

The SDVRP and the VRPDDP (Section 3.2) are both based on the idea that a single visit to the customer may not be enough to meet all of their demands (Wassan and Nagy, 2014). However, the VRPDDP is limited to two trips per client, one for delivery and one for pickup. Furthermore, the deliveries and pickups may not themselves be split (Wassan and Nagy, 2014). The SDVRP, on the other hand, allows deliveries to be split among multiple tours and vehicles. Moreover, in the SDVRP the optimal solution assures that each pair of routes has no more than a single customer in common, which does not apply to the VRPDDP (Wassan and Nagy, 2014).

3.4.1 Mathematical model for SDVRP

The following mathematical formulation is based on the k-SDVRP model by Archetti et al. (2006). It is modified by removing the restriction on the number of k-units supplied, making it a simple SDVRP. The model is adapted to the parameters and decision variables discussed in the preceding sections.

Decision Variables:

- d_{ik} = the quantity of demand i delivered by the kth vehicle

Constraints:

$$d_{ik} \quad q_i \sum_{j=0}^{V} x_{ijk} \quad \forall i \in V, k \in M \tag{3.25}$$

$$\sum_{k=1}^{M} d_{ik} = q_i \quad \forall i \in V \setminus \{0\} \tag{3.26}$$

$$d_{ik} \quad 0 \quad \forall i \in V, k \in M \tag{3.27}$$

Constraint (3.25) assures that the quantity of demand provided to customer i does not exceed the quantity ordered by this customer. Constraint (3.26) assures that the split orders for each demand node satisfy the total quantity demanded by customer i. Finally, constraint (3.27) assures that the delivered quantity is not negative.

3.5 VRP with private fleet and common carrier

The Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPFCC) was first designed to address the problem that enterprises faced when overall customer demand surpassed the total delivery capacity of private vehicles (Chu, 2005). As a result, total customer demand could not to be satisfied, necessitating the use of third-party carriers to resolve the problem. Although the costs of outsourcing are typically higher than the costs of direct service, there are some instances where using a private fleet may be more expensive, such as serving customers in difficult-to-reach areas or when vehicles must travel half-empty and thus a less-than-truckload carrier can be used (Higino et al., 2018). The VRPPFCC is about determining which customers to serve directly with private fleet and which through third-party carriers. The goal of this model is to reduce total expenses.

3.5.1 Mathematical model for VRPPFCC

Bolduc et al. (2008) present an updated version of the VRPPFCC model for heterogeneous fleets of cars. The concept ensures that each client is served precisely once by a private fleet vehicle or by a common carrier, and that each private fleet vehicle performs just one route (Bolduc et al., 2008). The new parameters, variables and constraints, which are unique for the VRPPFCC are shown and discussed below.

Parameters:

- e_i = the cost charged by the external carrier for serving customer i

Decision Variables:

- $w_{ik} = 1$ if customer i is served by the private fleet vehicle $k, 0$ otherwise
- $z_i = 1$ if customer i is outsourced to a common carrier, 0 otherwise

Objective Function:

$$\min z = \sum_{i=0}^{V} \sum_{j=0}^{V} (c_{ij} \quad \sum_{k=1}^{M} x_{ijk}) + \sum_{i=0}^{V} e_i z_i \tag{3.1b}$$

Constraints:

$$\sum_{j=0, j\neq h}^{V} x_{hjk} = \sum_{i=0, i\neq h}^{V} x_{ihk} = w_{hk} \quad \forall h \in V \setminus \{0\}, k \in M \tag{3.28}$$

$$z_i + \sum_{k=1}^{M} w_{ik} = 1 \quad \forall i \in V \tag{3.29}$$

$$\sum_{i=1}^{V} q_i w_{ik} \leq Q_k \quad \forall k \in M \tag{3.30}$$

$$w_{ik} \in \{0, 1\} \quad \forall i \in V, k \in M \tag{3.31}$$

$$z_i \in \{0, 1\} \quad \forall i \in V \tag{3.32}$$

The VRPPFCC adds two more factors related to the decision between a private fleet and an external carrier. As a result, the objective function (3.1b) has been modified to reflect the total costs associated with using the external carrier. Equation (3.28), on the other hand, illustrates the same vehicle k serving and departing from customer h. Constraint (3.29), on the other hand, assigns each demand node to a private fleet or an external carrier. Constraint (3.17) is replaced with (3.30), which ensures that the private fleet's capacity is not exceeded. Finally, constraints (3.31) -(3.32) make sure the variables are binary.

3.6 Multi-trip VRP

The Multi-Trip Vehicle Routing Problem (MTVRP), also known as the VRP with multiple use of vehicles, was first introduced by Fleischmann (1990). The MTVRP is created to address scenarios where vehicles can undertake shorter tours and then, those vehicles can be used again for the second or third time within the same working day. This enables the same vehicle to perform multiple routes during a single day, a significant difference to the Single-Trip VRP, where each vehicle is restricted to making a single tour to serve all of the assigned customers (Cattaruzza et al., 2016). In contrast to the Single-Trip VRP, the MTVRP allows vehicles to return to the depot multiple times for loading, unloading, and serving additional customers on subsequent trips.

3.6.1 Mathematical model for MTVRP

The model for the Multi-Trip VRP is based on the 4-index formulation presented in Cattaruzza et al. (2016)'s work. This model has been modified and suited to the parameters and decision variables described in the preceding sections. Furthermore, numerous constraints from previous models have been adjusted, while new constraints have been added to account for the multi-trip situation.

Parameters:

- T = set of possible trips a vehicle can make.
- $T_{\max}$ = maximum duration of trips.

Decision variables:

- $x_{ijkr} = 1$ if vehicle k travels from customer i to customer j during trip r, otherwise 0.
- R_{ijr} is the amount of delivery goods on board on arc (i, j) during trip r.
- P_{ijr} is the amount of pickup goods on board on arc (i, j) during trip r.
- t_{ikr} is the time vehicle k starts to service customer i in trip r.

Objective function:

$$\min z = \sum_{i=0}^{V} \sum_{j=0}^{V} \left(c_{ij} \sum_{k=1}^{M} \sum_{r \in T} x_{ijkr} \right) + \sum_{i=0}^{V} e_i z_i \tag{3.1c}$$

Constraints:

$$\sum_{i,j \in V} x_{ijkr} \geq \sum_{i,j \in V} x_{ijk(r+1)} \quad \forall k \in M, \forall r \in T \tag{3.33}$$

$$\sum_{i,j \in V} t_{ij} x_{ijkr} \leq T_{\max} \quad \forall k \in M, \forall r \in T \tag{3.34}$$

$$t_{ikr} + t_{ij} - \text{BigM}(1 - x_{ijkr}) \leq t_{jkr} \quad \forall i \in V, \forall j \in N, \forall k \in M, \forall r \in T \tag{3.23b}$$

$$a_i \leq t_{ikr} \leq b_i \quad \forall i \in V, \forall k \in M, \forall r \in T \tag{3.24a}$$

$$t_{ikr} \geq 0 \quad \forall i \in V, \forall k \in M, \forall r \in T \tag{3.35}$$

Within this model a new parameter is added related to the trips, and it has to be included in three of the decision variables. This results in modifying the objective function accordingly (3.1c). All constraints including the decision variables x_{ijkr}, R_{ijr}, and P_{ijr} in the previous models have to be modified to reflect the new index for the trips r. Constraint (3.33) ensures that if a vehicle is making a trip $r+1$, it must have made trip r before that. Constraint (3.34) ensures the total duration of each trip r is not exceeded. Constraint (3.23b) is modified to reflect the new variable t_{ikr} and it ensures customer j is served after customer i by vehicle k in route r and that subtours are eliminated. The customer time windows are considered in the modified constraint (3.24a). Finally, constraint (3.35) ensures the new variable is non-negative.

3.7 Combining multiple features of the VRP

The Vehicle Routing Problem models and formulations described in Sections 3.1-3.6 investigate specific aspects such as time-windows, capacity, load-splitting, multiple trips, simultaneous delivery and pickup, and case with common carrier and private fleet. However, in real-life situations, businesses encounter a mix of the features.

The multi-trip pickup and delivery problem with split loads and multiple time windows (VRPSDP-SLTW) is one such combination developed by Wang et al. (2013) to handle a real-world construction scenario. The situation consists of a single depot, a set of clients who may request both pickup and delivery services within a certain time frame, and a fleet of vehicles to serve clients. Because simultaneous pickup and delivery may cause the vehicle capacity to be exceeded, the authors approach the problem as a split delivery situation, in which each client can be visited many times by a single vehicle on a single route or numerous vehicles along different routes. Wang et al. (2013) model the VRP with the goal of minimizing both the number of vehicles used and overall travel expenses.

Suprayogi and Priyandari (2017) also discussed the VRP with multiple trips, time windows, and simultaneous pickup and delivery with the weighted objective of minimizing the total tour duration time and number of vehicles used. The model also uses capacity constraints, and time windows for both depot and customer nodes.

A real-world instance requiring combining a number of VRP elements happened in the construction industry, where building material delivery and construction garbage disposal were needed. Jaballah and Cherif-Khettaf (2021) applied the multi-trip pickup and delivery problem with split loads, profits, and multiple time windows characteristics to this construction company case. This problem investigates various time frame specifications for some of the demand nodes, as well as a heterogeneous fleet of cars and demand splitting. Because of the limited number of vehicles, this problem solution could not serve all clients; therefore, customers were chosen based on the revenue contribution to the company (Jaballah and Cherif-Khettaf, 2021).

Furthermore, Zhang et al. (2023) suggested a split-demand multi trip vehicle routing problem with simultaneous pickup and delivery, which, while focused on scheduling luggage transit trains, may be extended to truck routing. The tugs can visit a flight once every trip, however this can be done numerous times with a set maximum number of trips (Zhang et al., 2023). During those trips, a pickup and delivery service might be provided concurrently. They also include capacity limits and loading time, as well as time windows for luggage release and return, making the scenario much more difficult. The goal is to reduce the total cost of the route.

The heterogeneous fleet vehicle routing problem with split deliveries, multiple products, and multiple trips with time window limitations was stated in a gasoline distribution instance (Nugroho et al., 2020). The study was prompted by the increasing demand for gasoline distribution, which resulted in specific demands not being met (Nugroho et al., 2020). Within the customer's working hours, a fleet of trucks is required to visit and serve multiple gas stations with either a single cargo or several deliveries. The study's goal is to reduce overall distribution costs.

Some models, such as VRP with private fleet and common carrier, are new and have not yet been integrated with many other model characteristics. However, Dang et al. (2022a) considered the vehicle routing problem with common carriers and time constraints due to the recognition of its vast application in real-life instances. They proposed a mixed integer linear programming formulation with customer time frames and truck capacity as constraints. The goal was to keep total transportation costs as low as possible. Dang et al. (2022a) applied the concept to a large-scale scenario and saved money by carefully selecting outsourcing deliveries.

Table 3. Literature review of VRP variants and their solutions.

Article	VRP Features						Objective	Solution
	MT	C	SD	TW	DDP	PFCC		
Jaballah and Cherif-Khettaf (2021)	✓		✓	✓	✓		Minimize the number of used vehicles, total duration time, total distance, and maximize the number of priority customers	Score Based Heuristic (SBH)
Wang et al. (2013)		✓	✓	✓	✓		Minimize the number of vehicles required and total travel costs	Hybrid Heuristic Method (HHM), Construction Heuristic Algorithm (CHA), Reactive Tabu Search (RTS)
Suprayogi and Priyandari (2017)	✓	✓		✓	✓		Minimize total tour duration time and number of vehicles used	Tabu Search (TS), Local Search (LS), Genetic Algorithm (GA)
Zhang et al. (2023)	✓	✓	✓	✓	✓		Minimize total costs	Adaptive Large Neighbourhood Search (ALNS)
Nugroho et al. (2020)	✓		✓	✓			Minimize total costs	Sequential insertion
Dang et al. (2022a)		✓		✓		✓	Minimize total costs	Red-black Ant Colony Search (RB-ACS)

Table 3 contains an overview of the articles referred to in this section, and the solution approaches presented in each of those. Despite the fact that multiple authors have merged four to five features, no model in the literature covers all of this thesis' problem aspects. Most articles concentrate on the VRP with Simultaneous Pickup and Delivery, but not on the VRP with Divisible Delivery and Pickup. Furthermore, no research has been conducted on the VRP with private fleet and common carrier in combination with other elements such as simultaneous pickup and delivery or multi-trip case.

3.8 Heuristics

The Vehicle Routing Problems are about finding the shortest tours visiting all nodes precisely once, also known as a Hamiltonian cycle in graph theory. As the number of nodes to visit increases so does the problem's complexity, making it an NP-complete problem (Ni and Tang, 2023). Exact approaches, such as direct tree search methods, dynamic programming, and integer linear programming, are used to solve VRPs (Laporte and Nobert, 1987). However, in a real-world setting, the problem gets considerably more complex because the VRP involves many different constraints, making exact techniques highly difficult and time-consuming to solve. Heuristic and metaheuristic algorithms can be used to solve such large-scale issues since they can deal with a large number of constraints and nodes while obtaining near-optimal solutions in a reasonable amount of time.

Heuristics are categorized into constructive and improvement. The constructive heuristic seeks an initial solution to the problem, whereas the improvement heuristic aims to improve the solution to the greatest extent possible. Metaheuristics, on the other hand, can be divided into two types: local search methods that explore the solution space by moving from one solution to the next in each iteration, and population-based metaheuristics that evolve a collection of potential solutions (population) to explore the solution space by mimicking biological evolution (Laporte et al., 2014).

The nearest-neighbour algorithm, savings algorithm, and two-stage heuristics are some of the oldest and very widely used constructive heuristics for finding an initial solution to the VRP (Ni and Tang, 2023; Laporte et al., 2014; Blocho, 2020). These are frequently used in conjunction with improvement heuristics operators such as reinsertion, swap, and 2-opt (Laporte et al., 2014). The constructive heuristics are discussed in Section 3.8.1.

The most common solution approaches for VRP with simultaneous pickup and delivery variations, according to Wassan and Nagy (2014), include Tabu Search algorithms, Ant Colony Optimization, and Adaptive Large Neighbourhood Search. In general, local search metaheuristics are used more than population-based ones for most VRP variants. Simulated Annealing, Iterated Local Search, Large Neighbourhood Search, and Greedy Randomized Adaptive Search Procedure are frequently used in VRPs with multiple characteristics (Wang et al., 2013; Suprayogi and Priyandari, 2017; Zhang et al., 2023; Ni and Tang, 2023). As a result, those are discussed in the sections that follow, along with the algorithm for each (Sections 3.8.2-3.8.8).

3.8.1 Constructive heuristics

Constructive heuristics develop the routing solution from scratch by making empirical judgments (Laporte et al., 2014). These heuristics are typically employed as the first step toward improvement and metaheuristics. The Clarke and Wright savings algorithm is one such heuristic, in which initial routes from the depot and back to the depot are constructed for each customer, savings are calculated, and customers are merged based on the highest savings until the route is complete (Clarke and Wright, 1964).

Solomon suggested an insertion-based approach for solving VRPTW in 1987. Customers were repeatedly put into the initially empty list of routes based on factors such as trip cost savings, time frame consideration, and a combination of the two until all customers were served (Solomon, 1987). Solomon (1987) presented the Time-Oriented Nearest-Neighbor heuristic, where every route starts by discovering the consumers which are not routed and are closest to the depot and the algorithm looks for the unassigned client closest to the last customer added at each iteration. Many other parameters, such as vehicle capacity limits, vehicle arrival and departure times, and client time frames, can be added to the Nearest-Neighbor.

The cluster-first route-second heuristic is a two-stage heuristic that divides consumers into viable sets before determining the optimal route inside each cluster (Fisher and Jaikumar, 1981). The algorithm begins by selecting initial points (seed points) from the customer nodes and allocating each customer to those seeds based on cost savings. A Generalised Assignment Problem must then be solved to assign all customers into clusters while ensuring total vehicle capacity is met (Laporte et al., 1999). The final step is to solve the problem for each acquired cluster. The method was then used to solve various VRP variations, such as multi-trip VRP (Garside and Laili, 2019).

3.8.2 Simulated Annealing

The Simulated Annealing (SA) optimization was introduced in 1983 by drawing its roots from the statistical mechanics (Kirkpatrick et al., 1983). The Simulated Annealing starts with an initial solution and iteratively explores the solutions space by accepting both improving and non-improving solutions based on a temperature parameter to escape from the local optima. Osman (1993) was one of the first to apply and evaluate the Simulated Annealing to the VRPs. The SA approach has proven to be producing near-optimal solutions in a relatively short time (Osman, 1993). Since then, the Simulated Annealing has begun to be widely applied for many variants of the VRPs. For example, a SA algorithm is proposed with a mechanism of repeatedly cooling and raising the temperature to solve the CVRP (Wei et al., 2018). The algorithm can also be used for VRPTW and is capable of solving big problems with up to 350 customers and 50 vehicles in a short time (Mohammadi et al., 2022). The algorithm for the simulated annealing for capacitated VRP is presented in Algorithm 13 (see Appendix A.1), based on Wei et al. (2018).

3.8.3 Tabu Search

Tabu Search (TS) is a metaheuristic algorithm developed in 1986 as an iterative procedure to guide other heuristics to escape the local optima and explore the solution space (Glover, 1986). Osman (1993) applied the Tabu Search algorithm to a CVRP and found out that it is more robust and outperforms the Simulated Annealing algorithm in solution quality and computation time. In the later years, the Tabu Search was applied to many VRP variations, including the VRPBTW presented in the study of Duhamel et al. (1997). The TS starts with an initial solution and explores the neighbouring solutions by applying local moves like 2-opt, swap, and Or-opt. The solutions are evaluated based on the importance of the problem KPIs or objective function, such as distance travelled or costs. It then checks whether the solution is in the Tabu List, which is a list of recent moves aiming to prevent revisiting them. If the solution is in the Tabu List, then it is not accepted, unless it does not improve the best solution found until this moment. The solution and Tabu List are updated, and the algorithm continues until a stopping criterion is met (David J. Rader, 2010). The Tabu Search heuristic as presented in the book of David J. Rader (2010) is given in Algorithm 14 (see Appendix A.2).

3.8.4 Iterated Local Search

An iterated local search algorithm (ILS) is a metaheuristic which focuses on a smaller subset of solutions, instead of considering the entire solution space. The ILS algorithm, developed by Cuervo et al. (2014) is shown on Algorithm 15 (see Appendix A.3). It begins with an initial solution, which is improved by using a local search algorithm to find a local optimum. Following, the solution is perturbed by using methods like removing randomly selected customer and inserting it back in a different randomly selected position to escape from the local optima and explore the solution space (Cuervo et al., 2014). Then, a decision is made whether to use the new solution found or the old one, or even a combination of both and the process is repeated iteratively from perturbing a solution, until a stopping criterion is met. The ILS has been used to solve the VRP with Backhauls and due to its simplicity, the algorithm usually has been proven to be fast to execute (Cuervo et al., 2014). Moreover, a multi-start ILS has been developed for the split delivery VRP with limited fleet, making use of perturbation mechanism called Multiple-k-split, which removes randomly between 5 and 7 customers and reinserts them into other places (Silva et al., 2015).

3.8.5 Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) is a metaheuristic for solving optimization problems by exploiting the idea of a neighbourhood change to escape from local optima. It explores distant neighbourhoods of the current incumbent solution and changes to a new solution only if an improvement is made (Mladenovic and Hansen, 1997). This ensures that most variables which are already at their optimal value will be kept and only the ones which are not, will be updated through exploring neighbouring solutions. The algorithm for the basic VNS based on Hansen and Mladenovic (2005) is given in Algorithm 1.

Algorithm 1 Variable Neighborhood Search (VNS)

$x \leftarrow$ FindInitialSolution()
Define neighborhood structures $N_k, k = 1, \ldots, k_{\max}$
while stopping condition not met **do**
  $k \leftarrow 1$
  while $k \leq k_{\max}$ **do**
    $x' \leftarrow$ Shake(x, N_k)
    $x'' \leftarrow$ LocalSearch(x')
    if Cost(x'') $<$ Cost(x) **then**
      $x \leftarrow x''$
      $k \leftarrow 1$
    else
      $k \leftarrow k + 1$
    end if
  end while
end while
return x

3.8.6 Variable Neighbourhood Descent

The variable neighbourhood descent (VND) is a specific type of local search methods, which can be used within the VNS framework. VND is a deterministic local search method for exploring neighbourhoods to escape from the local optima (Hansen et al., 2010). The VNS explores different neighbourhoods and execute local search to find better solutions, whereas VND focuses on improving the current solution without a further improvement of any of the neighbourhoods. In recent year, variations of the VND algorithm have been applied to solve VRPs such as the random variable neighbourhood descent (RVND), which does not follow a predefined order of the neighbourhood structures and randomly selects the order in each execution (Higino et al., 2018). The RVND has been mainly applied to VRP with profits and VRP with private fleet and common carrier (Doan et al., 2021; Higino et al., 2018). The algorithm for the basic VND is presented on Algorithm 16 (see Appendix A.4).

3.8.7 Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic that mimics the behaviour of real ants, who communicate with one another by leaving pheromone depots on the ground for other ants to follow (Bell and McMullen, 2004). ACO builds a solution by probabilistically selecting pathways based on pheromone levels. In VRP, an ACO is employed by building a complete tour for the initial routing while adhering to restrictions such as vehicle capacity, split deliveries, and time windows (Çatay, 2009; Rizzoli et al., 2007). Once an ant has completed its journey and returned to the depot, it may begin a new route until all clients have been served. Once all of the ants have completed their routes, the pheromone levels are updated, and the shortest or most cost-effective routes deposit more pheromones, encouraging future ants to follow (Dang et al., 2022b; Bell and McMullen, 2004). Iteratively, the procedure is repeated until a near-optimal solution is found for the VRP.

3.8.8 Greedy Randomized Adaptive Search Procedures

Greedy Randomized Adaptive Search Procedures (GRASP) are one of the most promising strategies for tackling optimization problems. It is an iterative procedure with two phases: construction and local search. The goal of the construction phase is to design a feasible solution by picking elements based on a greedy function. A subset of the top solution candidates is then saved in a Restricted Candidate List (RCL). The heuristic is adaptive since the benefits are changed in each iteration of the initial phase to highlight the differences with the selected elements. The items are chosen at random from the list, introducing variety and allowing for the exploration of alternative solutions at each iteration. When the solution is constructed, a local search is applied to explore for improvements. The GRASP has been applied widely to VRPB, CVRP, and VRPTW (Haddadene et al., 2016; Tütüncü et al., 2009; Marinakis, 2017). The generic GRASP algorithm is provided in Algorithm 17 (see Appendix A.5) and construction phase in Algorithm 18 (see Appendix A.5), adapted from Feo and Resende (1995).

3.9 Conclusion

This chapter answers the research question ***Based on the literature, what are the applicable techniques and methods for modelling and solving a vehicle routing problem?*** Despite the extensive exploration of VRP variants in the literature, a gap remains in addressing a model that precisely matches Meilink's combination of requirements, including split-load, multi-trip capabilities, divisible pickup and delivery, the integration of a private fleet with common carriers, and adherence to time windows.

Among the VRP variants reviewed, the closest alignment was found with the works of Zhang et al. (2023), which, while sharing several characteristics with Meilink's scenario, does not incorporate the private fleet and common carrier aspect. This highlights the novelty and complexity of Meilink's routing problem, showcasing the necessity for a tailored solution approach.

It became evident that a metaheuristic approach for solving Meilink's Vehicle Routing Problem (VRP) should be employed. This stems from the inherent complexity and scale of the transportation problem. Traditional exact methods, while capable of finding optimal solutions, are impractical for large-scale VRPs due to their computational intensity and the exponential growth of the solution space with the addition of more variables and constraints. Meilink's VRP, characterized by its combination of multiple trips, split deliveries, simultaneous pickup and delivery, and the integration of private fleets and common carriers, presents a highly complex problem that surpasses the capabilities of exact algorithms within reasonable computational times. Metaheuristics offer a balance between solution quality and computational efficiency, providing near-optimal solutions within a feasible timeframe. Notably, the Variable Neighborhood Search (VNS) algorithm emerges as a particularly promising method for addressing Meilink's VRP.

The decision to employ VNS is motivated by several factors. While SA and TS are powerful in exploring the solution space, they are generally constrained to a single or a limited set of neighborhood structures. VNS, on the other hand, systematically explores multiple neighborhood structures, allowing for a more comprehensive search of the solution space. This makes VNS particularly successful at adapting to the complex nature of Meilink's VRP, which combines multiple trips, split deliveries, and the use of both private fleets and common carriers. VNS's strategy of changing neighborhood structures ensures a broader and more diverse exploration, increasing the likelihood of finding globally optimal or near-optimal solutions when compared to ILS. ACO and GRASP are effective for specific problem types, however, they may not always guarantee a thorough exploration of the solution space due to their reliance on pheromone trails and greedy constructions, respectively. VNS's sequential approach to exploring different neighborhoods allows for a more thorough search, which is crucial for addressing the complex constraints and objectives of Meilink's VRP.

While existing literature provides valuable insights into various VRP variants and solution methodologies, the unique combination of requirements in Meilink's transportation process necessitates a novel approach. The choice of VNS as the solution method is justified by its adaptability, effectiveness in large-scale scenarios, and capability to integrate complex VRP features, making it a suitable candidate for developing an optimized routing strategy for Meilink Borculo B.V. This study aims to bridge the gap in the literature by tailoring the VNS algorithm to Meilink's specific needs, contributing to the body of knowledge on VRPs and offering practical solutions to complex routing challenges.

4 Solution Approach

After a thorough literature review, it has become evident that Meilink's problem is complex and consists of several characteristics found in literature research related to Vehicle Routing Problems (VRPs). This chapter is dedicated to presenting the design of the solution to the core problem addressed in this research. More specifically, this chapter aims to address the posed research question *How should the solution approach for Meilink's transportation process be designed?* Consequently, the chapter begins by describing the transportation problem model of Meilink in Section 4.1, including a discussion of the model's requirements (4.1.1) and assumptions (4.1.2). The mathematical formulation of the model is provided in Section 4.2. In Section 4.3 a toy illustration of the model and a solution to it is provided. Section 4.4 introduces a metaheuristic solution through the Variable Neighbourhood Search algorithm to tackle larger instances of the problem.

4.1 Model description

In the Meilink Borculo B.V. transportation problem can be described as a Multi-Trip Capacitated Vehicle Routing Problem with Divisible Delivery and Pickup, Time Windows, Private Fleet, and Common Carriers (MTCVRPDDPTWPFCC) and can be defined as follows. Let $G = (V, A)$ be a directed graph, where $V = L \cup B \cup \{0\}$ is the node set that contains all customers and the depot nodes, and A is the arc set. There is a set $L = \{1, 2, 3, \ldots, n\}$ of customers with delivery requests (linehaul) and a set $B = \{n+1, n+2, n+3, \ldots, 2n\}$ of customers with pickup requests (backhaul) where both sets of customers are to be served by Meilink. Each customer can be either a linehaul, or backhaul, or both at the same time. The Borculo location has only one depot, indicated by $\{0\}$ and the depot has working hours during which it can be visited by the vehicles, denoted by $[a_0, l_0]$. A distance $Dist_{ij}$ is defined for each arc between each pair of vertices (i, j), $i \neq j$, $i, j \in V$. It is assumed that travel distances are symmetric and satisfy the triangle inequality. Each customer $i \in L \cup B$ has a demand (either pickup or delivery) d_i and a time window with start time a_i and end time l_i. A customer can only be served within their time window $[a_i, l_i]$. In Meilink, the number of private vehicles is limited, and there is a set K containing the private fleet of the company. Each vehicle from the private fleet has a corresponding volume capacity Q_k and a unit cost u_k for each kilometer travelled, which includes the variable costs of each vehicle, such as fuel expenses. Moreover, each vehicle has a fixed cost c_k, including employee wages, maintenance, and depreciation, accumulated only if the vehicle is used. As defined in the requirements, all private vehicles can be used for at most 9 hours ($T_{\max}$) a day. Meilink also relies on the services of externally employed vehicles. Those usually charge a fixed cost e_i for serving a particular client location. The MTCVRPDDPTWPFCC aims to minimize the total transportation costs while satisfying the requirements provided in Section 4.1.1 and considering the model assumptions (4.1.2). These requirements include visiting the customer and depot nodes only within the time window, allowing the vehicles to satisfy customer demand within multiple trips, and ensuring the vehicles' volume capacity is not exceeded.

4.1.1 Requirements

The solution to the VRP in the problem must adhere to several requirements. These requirements are specified below and will be included as constraints to the model formulation.

- Each demand node can be visited more than once.
- A vehicle cannot transport more than its maximum volume per trip.
- Each customer can be a delivery, a pickup location or both.
- A vehicle must start and end each trip at the depot location.
- Each vehicle can travel at most 9 hours per day.
- The customers have expected delivery date and time, thus, the time windows have to be adhered to. This includes also the depot working hours.
- Each client can be served by either a private fleet vehicle or by common carrier per trip.

- Each vehicle starts with no pickup load and should return to the depot with no delivery load. This is to ensure that all of the loaded items are delivered and that there are no items to unload upon starting the trips.
- The total demand of the customers can be satisfied in multiple trips, however, ensuring that the time windows are met.

These requirements will be considered in the mathematical model formulation as well as in the developed solution approach.

4.1.2 Assumptions

In order to formulate the Vehicle Routing Problem (VRP) of Meilink Borculo, several assumptions have been made. These crucial for the effective modelling assumptions are outlined below:

- The items to be transported vary in volume and weight. The volume of the objects is computed based on their dimensions and used as input to the model. It is assumed the total weight of the items is always within the allowable limit for each vehicle. Therefore, the weight of the items is not considered a limitation and will not be formulated as a constraint.
- It is assumed 80% utilization of each vehicle's volume capacity to reflect real-world loading constraints. This pragmatic approach accounts for the diverse type of cargo, including items that cannot be stacked or require special handling, ensuring flexibility and safety in transport. The 20% buffer is strategically chosen to accommodate these variations, enabling a more accurate and feasible routing solution that mirrors the real-world logistic challenge.
- Deliveries and pickups are scheduled on their last expected delivery day.
- Each driver is assigned to a single vehicle. The drivers do not form a limitation and their working hours are considered by the maximum travel time of the vehicles.
- The service time for loading and unloading at each demand location is fixed at 30 minutes. However, at the depot the service time is extended to an hour, considering the larger quantities of products are being loaded or unloaded there.
- For customers without specified time windows, their working hours are considered as their available time windows. Travel times are assumed to be deterministic, and waiting at a location is allowed. However, starting service before the beginning of a time window is not allowed, meaning vehicles may have to wait if they arrive early.
- All pickup and delivery requests are considered deterministic. The demand is assumed to remain constant overtime.
- The travel time is a ratio between distance in kilometers and average speed of the vehicle.
- Outsourced vehicles are not a subject to capacity constraints, as various vehicles can be used for transportation purposes. Costs associated with outsourced vehicles do not depend on the type of vehicle used for and there are no constraint on the number of external vehicles available.
- Given that the depot is closed on Saturdays and Sundays, any delivery dates originally scheduled for the weekend are adjusted to the preceding Friday.
- Split deliveries are no longer considered, as each item is formulated as a new demand node to ensure the items are transported in their complete volume.

4.2 Mathematical formulation

The mathematical formulation of Meilink Borculo B.V's transportation problem is based on the models of Bolduc et al. (2008) and Wassan et al. (2017). Bolduc et al. (2008) focuses on formulating a VRP with private fleet and common carriers, whereas Wassan et al. (2017)'s model is on the MT-VRPB. This formulation below combines these two models and expands on those by adding the time-window constraints. Moreover, some of the constraints and parameters of the literature mathematical formulations are modified to incorporate Meilink's transportation problem aspects. Split deliveries are not considered in the model, as each item is formulated as a new demand node to ensure the items are transported in their complete volume. This reduces the problem to a multi-trip capacitated VRP with divisible delivery and pickup, time windows, private fleet and common carriers (MTCVRPDDPTWPFCC). The Mixed Integer Linear Programming (MILP) formulation begins with setting the parameters of the model, based on the transportation problem of Meilink Borculo B.V, followed by the decision variables, objective function and the constraints, which were previously described under the model requirements.

4.2.1 Parameters

Meilink Borculo B.V.'s transportation problem has the following parameters:

$L = \{1, 2, 3, \dots, n\}$: set of linehaul customers (delivery requests), indexed by i and j
$B = \{n+1, n+2, n+3, \dots, 2n\}$: set of backhaul customers (pickup requests), indexed by i and j
$V = L \cup B \cup \{0\}$: set of the depot and customers
$K =$ set of private fleet vehicles indexed by k
$Q_k =$ volume capacity of vehicle k
$d_i =$ demand (volume) of customer i such that $i \in B$ for pickup demand and $i \in L$ for delivery demand
$Dist_{ij} =$ distance (in km) from node i to node j
$\beta =$ travel speed multiplier
$a_i =$ start time of customer i's time window
$l_i =$ end time of customer i's time window
$\gamma =$ time for serving customer i
$e_i =$ cost charged by external carrier for serving customer i
$u_k =$ unit cost per km travelled by vehicle k
$c_k =$ fixed cost for using vehicle k
$T_{\max} =$ maximum travel time of vehicles
$M =$ a large positive constant number

4.2.2 Decision variables

The decision variables of the model are the following:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels directly from customer } i \text{ to customer } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$R_{ij} = \text{volume of pickup or delivery on board on arc } ij$$

$$w_{ik} = \begin{cases} 1 & \text{if private fleet vehicle } k \text{ serves customer } i, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{if customer } i \text{ is served by an outsourced carrier,} \\ 0 & \text{otherwise.} \end{cases}$$

$$s_{ik} = \text{the time when vehicle } k \text{ starts serving customer } i$$

$$y_k = \begin{cases} 1 & \text{if private fleet vehicle } k \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$$

4.2.3 Objective function

The total cost minimization objective can be given as:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} u_k Dist_{ij} x_{ijk} + \sum_{i \in V} e_i z_i + \sum_{k \in V} c_k y_k \tag{4.1}$$

The objective function is minimizing the total routing costs, which are composed of three distinct types of expenses. The first component of the summation represents the costs incurred by the private fleet vehicles. This includes the unit costs for the total distance travelled in kilometers, accounting for all trips made to demand locations and the return to the depot. The second summation component represents the charges imposed by common carriers for serving customers. The final element accounts for the fixed costs associated with deploying any vehicle k from the private fleet.

4.2.4 Constraints

The constraints, aligned with the assumptions and requirements of Meilink Borculo B.V.'s transportation problem are as follows:

$$\sum_{j \in N} x_{0jk} \quad y_k \qquad \forall k \in K \tag{4.2}$$

$$\sum_{j \in N} x_{j0k} \quad y_k \qquad \forall k \in K \tag{4.3}$$

$$\sum_{j \in V} \sum_{k \in K} x_{jik} = 1 \quad z_i \qquad \forall i \in N \tag{4.4}$$

$$\sum_{j \in V} x_{jik} = \sum_{j \in V} x_{ijk} = 0 \qquad \forall i \in N, \forall k \in K \tag{4.5}$$

$$\sum_{i \in M \cup L} R_{ij} \quad d_j = \sum_{i \in V} R_{ji} \qquad \forall j \in L \tag{4.6}$$

$$\sum_{i \in L \cup B} R_{ij} + d_j = \sum_{i \in M \cup B} R_{ji} \qquad \forall j \in B \tag{4.7}$$

$$\sum_{i \in V} x_{ijk} \quad w_{jk} \qquad \forall j \in N, \forall k \in K \tag{4.8}$$

$$R_{ij} \quad Q_k \quad x_{ijk} \qquad \forall i \in N, \forall j \in N, \forall k \in K \tag{4.9}$$

$$\sum_{i \in V} \sum_{j \in V} \beta \quad \mathrm{Dist}_{ij} \quad x_{ijk} + \sum_{i \in V} \sum_{j \in V} \gamma_i \quad x_{ijk} \quad T_{\max} \qquad \forall k \in K \tag{4.10}$$

$$R_{ij} = 0 \qquad \forall i \in L, \forall j \in B \cup M \tag{4.11}$$

$$x_{ijk} = 0 \qquad \forall i \in B, \forall j \in L, \forall k \in K \tag{4.12}$$

$$x_{0jk} = 0 \qquad \forall j \in B, \forall k \in K \tag{4.13}$$

$$\mathrm{M} \quad y_k \quad \sum_{i \in N} w_{ik} \qquad \forall k \in K \tag{4.14}$$

$$s_{ik} + (\beta \quad \mathrm{Dist}_{ij} + \gamma_i) \quad s_{jk} \quad (1 \quad x_{ijk}) \quad \mathrm{M} \qquad \forall i, j \in V, i \neq j, \forall k \in K \tag{4.15}$$

$$a_i \quad s_{ik} \quad l_i \qquad \forall i \in V, \forall k \in K \tag{4.16}$$

$$\sum_{k \in K} w_{ik} + z_i = 1 \qquad \forall i \in N \tag{4.17}$$

$$x_{iik} = 0 \qquad \forall i \in V, \forall k \in K \tag{4.18}$$

$$x_{ijk} + x_{jik} \quad 1 \qquad \forall i, j \in N, \forall k \in K \tag{4.19}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in V, \forall k \in K \tag{4.20}$$

$$R_{ij} \quad 0 \qquad \forall i, j \in V, \forall k \in K \tag{4.21}$$

$$w_{ik} \in \{0, 1\} \qquad \forall i \in V, \forall k \in K \tag{4.22}$$

$$z_i \in \{0, 1\} \qquad \forall i \in V \tag{4.23}$$

$$s_{ik} \quad 0 \qquad \forall i \in V, \forall k \in K \tag{4.24}$$

$$y_k \in \{0, 1\} \qquad \forall k \in K \tag{4.25}$$

Constraints (4.2) and (4.3) ensure that each vehicle's tour begins and ends at the depot location, provided the vehicle is used. Constraint (4.4) guarantees that each customer is served either by an internal vehicle or by an outsourced carrier. In (4.5), the degree balance of each node is ensured. The delivery flow conservation constraint (4.6) ensures that the total quantity of goods delivered to each linehaul customer by vehicle k meets customer demand. Similarly, constraint (4.7) ensures flow conservation at the backhaul nodes.

The number of vehicles from the private fleet responsible for the objective function is determined through constraints (4.8) and (4.14). Constraint (4.9) ensures that the vehicle capacity on each arc is less than the vehicle's total capacity. The time spent on the road servicing customers across various routes must not exceed the maximum permitted driving time per vehicle per day, as stated in (4.10). Constraint (4.11) ensures that there is no load from a linehaul customer to either a backhaul customer or to the depot. Constraints (4.12) and (4.13) prohibit a vehicle from traveling from a backhaul to a linehaul customer and directly from the depot to a backhaul customer, respectively.

Time window constraints for both customer and depot nodes are established in (4.15)-(4.16). In (4.17), each customer must be served either by a private vehicle or by a common carrier. Constraint (4.18) prevents loops, and (4.19) ensures route continuity. Finally, the integrality constraints for the decision variables are formulated in (4.20)-(4.25).

4.3 Toy problem illustration

An illustrative solution to the defined Vehicle Routing Problem (VRP) discussed in the preceding pages is presented in Figure 8. This toy problem encompasses various constraints. It involves seven customer nodes labeled A to G, with each node having either a linehaul (l) or/and a backhaul (b) demand denoted by volume in cubic cm. Additionally, each customer node, including the depot, is assigned a time window within which service can be provided.

Distances between the nodes are represented in hours. The vehicles utilized in this scenario consist of two types: private vehicles (Vehicle 1 and 2) and an external vehicle, which functions as a common carrier. The private fleet vehicles possess a loading capacity of 1000 cubic centimeters (cm^3), while the external vehicle is not constrained by capacity limitations in the model, as trucks with varying dimensions can be outsourced.

Service time at each customer node is fixed at 30 minutes, and loading/unloading time at the depot is set at 1 hour. Vehicles have the flexibility to make multiple trips. Private fleet vehicles always begin and end their trips at the depot, ensuring route continuity. In contrast, external vehicles provide service to the customers but may not necessarily return to the depot.

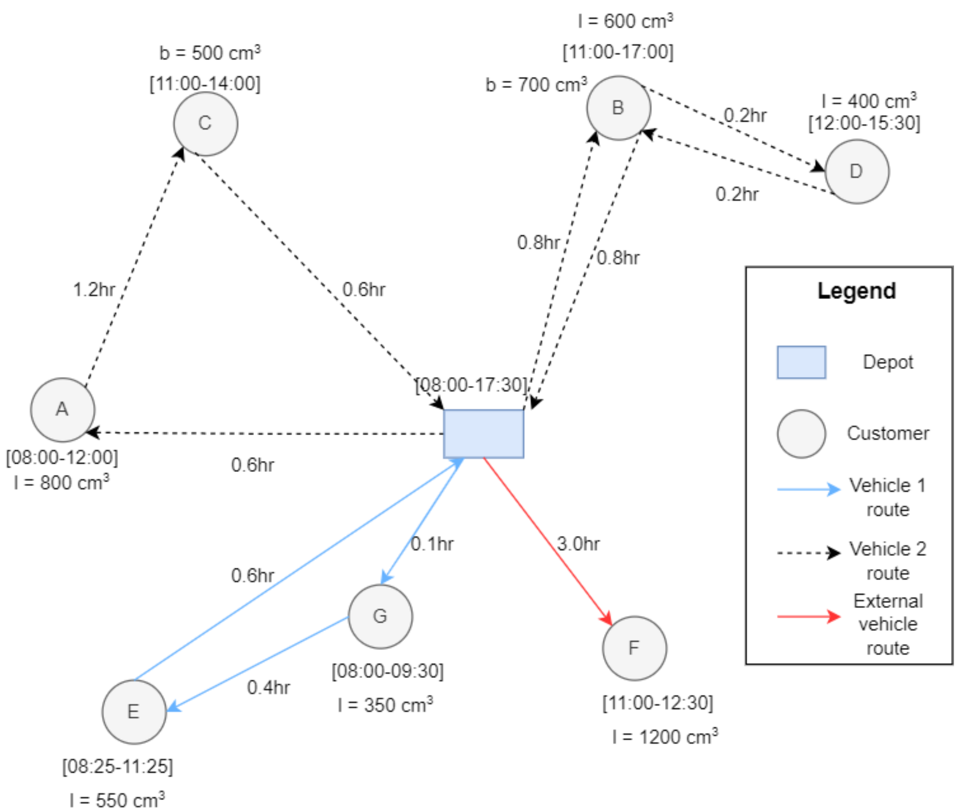

Figure 8. An illustration of a toy instance solution of a MTCVRPDDPTWPFCC.

In this solution, Vehicle 2 departs from the depot at 8:00 AM, coinciding with the depot's opening hours. The closest nodes to the depot are A and C; however, since both require the same traveling time of 0.6 hours, the time windows of the customer nodes need to be considered. Prioritizing the earliest time window ensures that the vehicle does not have to wait until the client's time window is open. In this scenario, the earliest time window belongs to customer node A. Consequently, Vehicle 2 leaves the depot and proceeds to node A. Node A is designated as a linehaul node, with a customer order volume of $l = 800$ cm³. Given that each vehicle has a volume capacity of 1000 cm³, Vehicle 2 departs from the depot carrying 800 cm³ to fulfill client A's order volume. Only after serving customer A can it attend to a backhaul customer, in this case, customer C. After arriving at customer A at 8:36 AM (due to the 0.6 hours traveling time), Vehicle 2 serves the customer for a fixed service time of 30 minutes, leaving customer A with a capacity of 1000 cm³ at 9:06 AM. Subsequently, the vehicle proceeds to the nearest node, which in this example is node C.

A backhaul demand totaling 500 cm³ needs to be serviced at node C, leaving vehicle 2 with a capacity of 500 cm³. The travel time from node A to node C is 72 minutes, leading to Vehicle 2's arrival at node C at 10:18 AM. Consequently, the vehicle must wait for 42 minutes before it can serve customer C. Upon serving customer C, the vehicle returns to the depot at 12:06 PM, accounting for both service and travel time. Although the nearest customer from node C is node B, continuing the route at this point is deemed suboptimal as customer B has both linehaul and backhaul demands that would exceed the vehicle's capacity on the initial tour from the depot. Thus, the vehicle returns to the depot to be loaded for an hour with the loads of the next customers it can serve, namely, customer B and D, with a total linehaul demand of 1000 cm³.

The vehicle embarks on a second tour to customer node B, the nearest customer. It services its linehaul demand first within its time window, as there is no spare capacity for pickup. Even after servicing customer B, the remaining capacity of the vehicle is 600 cm³, whereas the backhaul volume of the customer is 700 cm³. Consequently, the vehicle cannot simultaneously fulfill both the pickup and delivery demand of node B and must continue with its route. Vehicle 2 arrives at node D at 14:36, serves the linehaul demand, and departs from the customer at 15:06 with a spare capacity of 1000 cm³.

Subsequently, the vehicle returns to customer B to service its backhaul demand within the time window. It then returns to the depot, completing its second tour and the routing for the day. The depot closes at 17:30, leaving no time to continue serving customers as the depot loading time is 1 hour.

Vehicle 1 also departs from the depot and serves customers G and E, both of whom are linehaul customers. It starts from G and proceeds to E, as G shares the same opening time as the depot, thus excluding the need for the vehicle to wait. Following this, it continues to node E and subsequently returns to the depot. However, Vehicle 1 from the private fleet cannot serve customer F due to the node's distance from the depot and other nodes, which exceeds 3 hours, and its limited time window. Moreover, customer F's demand is 1200 cm², surpassing the capacity of the private fleet vehicles. Given these constraints, outsourcing a common carrier with larger trucks capable of accommodating customer F's demand volume and servicing the customer within its time window is necessary.

The objective function of this toy instance solution can be computed by considering the variable costs (per km traveled) of vehicle 1 and 2 from the private fleet, along with the fixed costs per day of utilizing those vehicles. The external truck incurs a fixed cost for visiting a customer location.

For vehicle 1 and 2 from the private fleet, the variable cost is 0.38 cents per km, and they traveled for 5.5 hours at a speed of 50 km/h, covering a total distance of 275 km. Additionally, there is a fixed cost of 50 euros for using each of the private fleet vehicles (Vehicle 1 and 2). Moreover, the external truck charges a fee of 240 euros for visiting customer node F. The total costs for this problem can be calculated as follows:

$$0.38 \quad 275 + 2 \quad 50 + 240 = €444.5$$

This includes the variable costs for the distance traveled by the private fleet vehicles, the fixed costs for their utilization, and the fee charged by the external truck for visiting customer node F.

4.4 Solution Methodology

Finding a feasible solution for larger instances of the formulated Mixed Integer Linear Programming (MILP) problem may be challenging, especially within a reasonable computation time. The MILP model comprises 25 distinct types of constraints, five binary variables, and one integer variable. Consequently, for a higher number of customer nodes, the complexity of the MILP model increases significantly. For instance, with a setup of a single depot, 30 customer nodes, and 10 vehicles — 5 of which are part of a private fleet — the model includes 9,610 x_{ijk} variables (calculated as 31 31 10), 300 s_{ik} variables, 150 w_{ik} variables, 961 R_{ij} variables, 30 z_i variables, and 5 y_k variables. This results in a total of 10,095 variables for this scenario. Due to the increased complexity in higher number of instances, the Variable Neighborhood Search metaheuristic is considered more efficient. This choice is motivated by the novelty of applying VNS to problems involving private fleets and common carriers, contributing to the academic literature. Moreover, VNS is recognized for its flexibility and efficacy in managing Vehicle Routing Problems with a large number of demand nodes.

4.4.1 Variable Neighbourhood Search

Algorithm 2 outlines the structure of the VNS algorithm employed in the solution approach. The main difference with Algorithm 1 is the stopping criterion: while Algorithm 1 stops when a certain time limit or specific target cost is reached, Algorithm 2 is instead setting maximum number of iterations without including the time as a stopping criterion. The VNS below diversifies the search process by systematically altering the neighborhood structure. This is achieved through the "Shake" function, which perturbs the current solution to explore new regions of the solution space, potentially uncovering better solutions. Following this, a "LocalSearch" procedure refines the new solution, seeking improvements within a neighbourhood. The algorithm iterates through a predefined set of neighborhood structures, dynamically adjusting based on the improvement observed in solution quality until the maximum number of iterations is reached.

Algorithm 2 Variable Neighborhood Search (VNS)

Input: InitialSolution, maxIterations, K_max
Output: bestCost, bestSolution
$bestSolution \quad InitialSolution$
$bestCost \quad$ CalculateCost($InitialSolution$)
$currentSolution \quad InitialSolution$
$iteration \quad 0$
while $iteration < maxIterations$ **do**
    $k \quad 1$
    while $k \quad K\_max$ **do**
        $newSolution \quad$ Shake($currentSolution, k$)
        $newCost \quad$ LocalSearch($newSolution$)
        if $newCost < bestCost$ **then**
            $bestSolution \quad newSolution$
            $bestCost \quad newCost$
            $k \quad 1$
        else
            $k \quad k + 1$
        end if
    end while
    $iteration \quad iteration + 1$
end while
return $bestCost, bestSolution$

4.4.2 Initial solution

Initial solution is needed for the VNS algorithm to have a basis from which it can begin its exploration. In this case, the initial solution is divided in three different functions, due to the complexity of the VRP in case and the constraints involved with it.

Algorithm 3 FindNextCustomerFunction()

```
Input: vehicle_state, customer_data, distance_matrix, unvisited_customers
Output: j_prime (ID of the next customer to visit)
current_location   vehicle_state[′current_location′]
min_distance   1
earliest_start_time   1
j_prime   None
for each customer j in unvisited_customers do
    distance_to_j   distance_matrix[current_location][j]
    start_time_j   customer_data[j][′start_time′]
    if distance_to_j < min_distance or (distance_to_j == min_distance and start_time_j < earliest_start_time) then
        min_distance   distance_to_j
        earliest_start_time   start_time_j
        j_prime   j
    end if
end for
return j_prime
```

Algorithm 3, the "FindNextCustomerFunction", is designed to identify the closest customer to a vehicle's current location by using a distance matrix that calculates Euclidean distances between customer nodes. In scenarios where the distances between vehicle's location and more than one customer are equal, then the customer with the earliest start time of the time window is prioritized.

Algorithm 4 ServeCustomerFunction()

```
Input: vehicle_state, customer_id, customer_data, time_matrix, unvisited_customers
Output: Boolean indicating if the customer was successfully served
current_location   vehicle_state[′current_location′]
Determine is_delivery or is_pickup for customer_id
Retrieve order_volume from customer_data
travel_time   time_matrix[current_location][customer_id]
arrival_time   vehicle_state[′time′] + travel_time
if is_delivery and remaining_capacity + order_volume > max_capacity then
    return False
end if
if is_pickup and remaining_capacity < order_volume then
    return False
end if
Adjust for waiting time if early
Calculate service time
Check total time against daily limit and time window constraints
if time constraints not met then
    return False
end if
Update vehicle_state based on is_delivery or is_pickup
Mark customer_id as visited in unvisited_customers
return True
```

The second function, "ServeCustomerFunction", is presented in Algorithm 4. Its primary purpose is to assess whether a customer can be served by taking into account the vehicle's capacity based on the type of demand (pickup or delivery). Additional constraints, such as the early arrival time of a vehicle to a customer node and waiting until the beginning of the customer time window, the depot opening and closing hours, the service times at various locations, and maximum travel time of vehicles are all taken into consideration. This function returns a boolean value indicating whether the customer can be successfully served. If the customer cannot be served, the function proceeds to the next customer in the list of unvisited customers.

Algorithm 5, the "RouteVehiclesFunction", plays a central role in creating the routes for vehicles. To include randomness in the initial solution routing, the function begins by shuffling the list of vehicles. Although the common carriers are with the same capacity and cost characteristics, this list also includes the private fleet vehicles, which are different in terms of both capacity and costs. This line with shuffling the list of vehicles can be removed and then, the vehicles will be chosen sequentially in the order they are stored in the dataframe. For each vehicle, the "FindNextCustomerFunction" is employed to identify a closest customer. The vehicle then attempts to serve the customers from the list of unserved ones based on the constraints in "ServeCustomerFunction". In cases where a customer is cannot be served, it is categorized as such and returned back to the list of unvisited customers. The function ends once all customers are served either by private fleet or common carriers, or when the maximum iteration limit is reached, indicating that routing constraints could not be met, resulting in incomplete routing. In case scenarios have to be generated to include only either private vehicles or common carriers, this function can easily be modified by removing the corresponding vehicles from the list of vehicles.

Algorithm 5 RouteVehiclesFunction

Input: vehicle_states, unvisited_customers
Output: Updated vehicle states with finalized routes
Initialize iteration count and set max_iterations
Randomly shuffle vehicle_states
while unvisited_customers and iteration max_iterations **do**
    Initialize attempted_customers
    for each vehicle_state in vehicle_states **do**
        Determine earliest start time for each vehicle based on customer time windows
        while unvisited_customers not empty **do**
            j_prime FindNextCustomer(vehicle_state, all_nodes_expanded, time_matrix, distance_matrix_expanded, unvisited_customers attempted_customers)
            if *j_prime* is None **then**
                Break from the loop; no viable next customer
            end if
            if ServeCustomer(vehicle_state, *j_prime*, all_nodes_expanded, time_matrix, unvisited_customers) is False **then**
                Add *j_prime* to attempted_customers for this vehicle
                Continue to next iteration
            end if
            Remove *j_prime* from unvisited_customers
        end while
    end for
    Increment iteration
end while
Finalize routes for each vehicle, including return to depot
if iteration > max_iterations **then**
    Print incomplete routing warning
end if
return vehicle_states

4.4.3 Shaking

After the initial solution is generated, the VNS continues with the shaking phase (Algorithm 6), which is crucial for exploring different regions of the solution space. The shaking phase within this implementation considers five distinct operators, described in Section 4.4.5. One of the operators is randomly selected to create a modified solution. This exploration allows for the subsequent local search to potentially discover better solutions by escaping the local optima.

Algorithm 6 Shake Function

Input: vehicle_states, customer_nodes
Output: new_solution, new_cost
new_solution copy of *vehicle_states*
selected_operator Randomly choose from list of operators
if success **then**
    new_solution Updated solution from *selected_operator*
    new_cost Updated cost from *selected_operator*
else
    new_cost calculateCosts(*new_solution*, *private_vehicles*, *all_nodes_expanded*)
end if
Return *new_solution*, *new_cost*

4.4.4 Local Search

The Local Search outlined in Algorithm 7 starts with an initial solution and attempts to improve it by applying a set of operators to the current solution iteratively. The best solution found until this point is stored and the algorithm iterates through the different operators to check if applying the operator results in a lower cost. If the costs are improved, then the best cost is updated and the improvement process continues until no further improvements can be made.

Algorithm 7 Local Search Algorithm

Input: vehicle_states, customer_nodes
Output: best_route, best_cost
Initialisation *best_route*, *best_cost*
improvements False
while *improvements* **do**
    for *operator* in *operators* **do**
        new_solution Updated solution from *operator*
        new_cost Updated cost from *operator*
        if *success* **and** *new_cost* < *best_cost* **then**
            best_cost *new_cost*
            best_route *new_route*
            improvements True
            break
        end if
    end for
end while
Return *best_route*, *best_cost*

4.4.5 Operators

The operators play a crucial role in the formulation of the VNS algorithm, as they enable the meta-heuristic to explore various neighbourhood structures and exploit the current solution space effectively. In this solution approach five different operators are incorporated based on the research of Hansen and Mladenovic (2005) and McNabb et al. (2015). The operators include Swap, Reinsertion, 2-Opt, Move and Swap Vehicles, each of which is individually described in the following sections. Before making any modifications of the routes and the overall solution, the feasibility of the solution after applying the operator is ensured, along with evaluating the potential cost improvement. The solution is updated only if there is a reduction in the best-found costs upon applying the operator. Additionally, solutions with identical costs are not stored to prevent looping.

4.4.5.1 Swap customer

The Swap operator outlined in Algorithm 8 begins by checking if there are at least two vehicles available for the swap and at least three customers per chosen vehicle. Without these condition the swap cannot be performed. If these two conditions are met, the operator randomly chooses two vehicles, each with one customer randomly selected from them. The swap is executed through exchanging the selected customers between the chosen vehicles. In case the newly generated routes are infeasible or the costs are not lower than the best costs found, the swap is reverted, ensuring that only valid and cost-improving solutions are considered.

Algorithm 8 Swap Customer Operator

```
Input: best_cost, best_route
Output: best_route, best_cost
if Number of vehicles > 2 then
    Randomly select vehicle1 and vehicle2
    if Both vehicles have at least 3 customers then
        Randomly select customer1 from vehicle1 and customer2 from vehicle2
        Swap customer1 and customer2
        Calculate the new_cost after the swap
    end if
end if
if solution feasible and new_cost < best_cost then
    Update best_cost and best_route
else
    Revert the swap
end if
return best_cost, best_route
```

4.4.5.2 Reinsertion

The Reinsertion operator begins by ensuring there is at least one vehicle which has a route from the initial solution with three or more customers (Algorithm 9). If this condition is met, the operator randomly selects one of the vehicles which have a route and randomly chooses a customer from this vehicle's route. This customer is removed from its current position and a new position is selected, ensuring it is different from the original position. The operator is not allowed to change the depot locations in the route. The customer is then reinserted into the new position and feasibility and cost checks are performed.

Algorithm 9 Reinsertion Operator

```
Input: best_cost, best_route
Output: best_route, best_cost
if Number of customers   3 then
    Randomly select vehicle and customer from its route
    Store original_position of customer
    Remove customer from its current position in vehicle
    Choose a new position for customer (not equal to original_position )
    Insert customer into the new position in vehicle
    Calculate the new_cost
end if
if solution feasible and new_cost < best_cost then
    Update best_cost and best_route
else
    Revert the reinsertion
end if
return best_cost, best_route
```

4.4.5.3 2-Opt

The 2-Opt operator begins by verifying whether at least one of vehicle's route contains four or more nodes (including the depot at the beginning and end of the route). The operator proceeds as described in Algorithm 10 by randomly selecting one of the available vehicles and randomly choosing two edges *a* and *b* within the vehicle's route. The edges must be non-adjacent, and edge *b* should not follow *a* immediately in the route sequence. The order of nodes between those two edges is swapped within the vehicle's route and feasibility and costs are assessed.

Algorithm 10 2-Opt Operator

Input: best_cost, best_route
Output: best_route, best_cost
if Number of customers in a vehicle 4 **then**
  Randomly select *vehicle* and two non-adjacent edges a and b in the route, ensuring b is not $a + 1$
  Reverse the order of the nodes between a and b in *vehicle*'s route
  Calculate the *new_cost*
end if
if solution feasible and *new_cost* < *best_cost* **then**
  Update *best_cost* and *best_route*
else
  Revert the 2-Opt
end if
return *best_cost*, *best_route*

4.4.5.4 Move

The Move operator begins with verifying that the number of vehicles with routes exceeds two (Algorithm 11). Then, two different vehicles are selected randomly.It is checked if the first vehicle selected has at least three customers to perform the move operator. Based on random selection a customer is removed from the first vehicle and inserted in a randomly determined position in the second vehicle. The new position has to be between the two depot nodes.

Algorithm 11 Move Operator

Input: best_cost, best_route
Output: best_route, best_cost
if Number of vehicles 2 **then**
  Randomly select *vehicle_from* and *vehicle_to*
  Randomly select a customer *customer* from *vehicle_from*'s route
  Remove the *customer*'s position from *vehicle_from*'s route
  insert_position to *vehicle_to*'s route
  Calculate the *new_cost*
end if
if solution feasible and *new_cost* < *best_cost* **then**
  Update *best_cost* and *best_route*
else
  Revert the Move
end if
return *best_cost*, *best_route*

4.4.5.5 Swap Vehicles

The Swap Vehicle operator, outlined in Algorithm 12, begins by verifying that the available number of vehicles exceeds two. Once the condition is met, two different vehicles are selected randomly from the list of vehicles, including both private fleet and common carriers. The customer nodes are swapped between the selected vehicles, creating two new routes for each of them. At the end, the feasibility and cost improvement of the solution is assessed.

Algorithm 12 Swap Vehicles Operator

Input: best_cost, best_route
Output: best_route, best_cost
if Number of vehicles 2 **then**
  Randomly select $vehicle1$ and $vehicle2$
  Swap the customer nodes between $vehicle1$ and $vehicle2$
  Calculate the new_cost
end if
if solution feasible and $new\_cost < best\_cost$ **then**
  Update $best\_cost$ and $best\_route$
else
  Revert the Swap
end if
return $best\_cost$, $best\_route$

4.5 Conclusion

This chapter provides a comprehensive overview of the solution methodologies applied to address the transportation challenge faced by Meilink Borculo B.V. More specifically, this chapter provides an answer to the research question *How should the solution approach for Meilink's transportation process be designed?* First, Meilink's routing problem is formulated as a Multi-Trip Capacitated Vehicle Routing Problem with Divisible Delivery and Pickup, Time Windows, Private Fleet, and Common Carriers (MTCVRPDDPTWPFCC). Following, the requirements and assumptions are defined.

The model must adhere to several requirements, including the ability for vehicles to make multiple trips, vehicle capacity limits, servicing of customers as either delivery or pickup locations, and the necessity for vehicles to start and end trips at the depot. Additionally, vehicles are constrained by a maximum daily travel time, and customer service must occur within specified time windows. The model also ensures that each customer can be served by either a private fleet vehicle or a common carrier per trip, and vehicles must return to the depot without any remaining delivery load.

Moreover, several assumptions underpin the solution approach to ensure its practical applicability and feasibility. These include the consideration of volume as the primary constraint over weight, an 80% utilization of vehicle capacity to account for loading constraints, and the deterministic nature of customer demands and travel times. The model assumes fixed service times, allows for waiting at locations if vehicles arrive early, and does not consider split deliveries. Additionally, it is assumed that outsourced vehicles are not subject to capacity constraints, and adjustments are made for deliveries initially scheduled on weekends.

Lastly, the solution approach for Meilink's complex routing problem incorporates a Mixed Integer Linear Programming (MILP) model for smaller instances and a Variable Neighborhood Search (VNS) metaheuristic for larger instances, focusing on minimizing total transportation costs while adhering to operational constraints. The MILP model is an exact method, which once solved generates the optimal solution to the VRP. However, the disadvantage of the exact method is the computation time required. The VNS, on the other hand, stands out for its adaptability and efficiency in handling extensive problem sizes through a strategic combination of initial solution generation, a diversifying shaking phase, an exploitative local search, and a set of operators (Swap Customers, Reinsertion, 2-Opt, Move, and Swap Vehicles). These elements work together to navigate the solution space effectively, escaping local optima and progressively refining solutions. This approach not only addresses Meilink's routing problem but also contributes to the broader field of vehicle routing problems by demonstrating the practical application and benefits of metaheuristic algorithms in complex logistical scenarios.

5 Evaluation

This chapter evaluates the two developed solution approaches, Mixed Integer Linear Programming (MILP) and Variable Neighbourhood Search (VNS), aimed at addressing the transportation problem of Meilink Borculo B.V. Thus, 10 real-world instances are developed and analyzed through a series of experiments. The primary goal of this chapter is to assess the performance of the developed solutions in optimizing Meilink's transportation process relative to the current cost structure. This evaluation is framed around the research question: *"How does the developed solution for optimizing Meilink's transportation process perform compared to the current situation?"*. The chapter unfolds by outlining the experimental design in Section 5.1, detailing the development of artificial and real-world data instances in Section 5.2, tuning the parameters for MILP and VNS in Section 5.3, and presenting the results from scenario tests in Section 5.4.

5.1 Experiment Design

The experimental framework contains five distinct experiments, labeled from *E0* to *E4*, as provided in Table 4. The initial experiment (*E0*) is dedicated to fine-tuning the parameter settings for both the VNS algorithm and MILP model. Proper parameter configuration is crucial for enhancing the efficiency and accuracy of the algorithms, thereby ensuring their capability to identify near-optimal solutions within a feasible timeframe. This phase involves adjusting parameters for the initialization, shaking, and local search stages of the VNS, in addition to setting the maximum computation time for both MILP and VNS.

Table 4. Experimental design.

Experiment ID	**Title**	**Goal**
E0	Parameter tuning	This experiment aims to identify the optimal settings for the VNS algorithm and the MILP to enhance solving efficiency by using artificial experimental data. For the VNS algorithm, the initialization method, the maximum number of iterations without improvement, the sequence of local search operators, and the shaking phase operators are determined. For the MILP, the maximum execution time is selected.
E1	Mixed fleet utilization	This experiment evaluates the performance of both private fleet and common carrier vehicles in the generated real-world data instances using the two solution methods (VNS and MILP). Common carriers are not routed, meaning external trucks can only serve customers directly from the depot without continuing to other customers before returning. The objective is to assess key performance indicators (KPIs)—cost, distance, running time, number of vehicles, and the percentage of unserved customers—across various instances and compare the efficacy of the two solution methods.
E2	Exclusively private fleet	Here, only private fleet vehicles are subjected to real-world instances tests using both solution methods. The goal is to analyze the KPIs in different instances and compare the solution methods' effectiveness.
E3	Common carriers only	In this setup, only common carrier vehicles are tested in real-world instances with both solution methods. Similar to Experiment *E1*, common carriers are not routed; they serve from the depot and do not continue their routes to serve other customers. The aim is to evaluate and compare KPIs across instances, focusing on the performance of the VNS and MILP methods.
E4	Routing common carriers	This experiment allows common carriers to be routed, enabling them to serve multiple customers per departure from the depot, similar to private fleet vehicles, using both solution methods. The objective is to assess and compare KPIs under various instances. Additionally, this experiment facilitates a comparison with Experiments *E1* and *E3*, which do not route common carriers, offering insights into how routing affects KPI improvements.

To conduct these parameter tuning experiments, artificial data instances are created to ensure the VNS and MILP's general applicability, thus avoiding overfitting to the specific context of Meilink Borculo B.V. These datasets include both fewer and more instances than those observed in real-world instances. The instances are generated using data available from Meilink and are further detailed in Section 5.2.

Following the algorithms' parameter tuning, several real-world instances are developed and evaluated in Experiments *E1* through *E4*. Utilizing Meilink Borculo B.V.'s customer data and the existing cost structure, these instances are designed to examine the company's actual situation. The instances aim to evaluate the solution methods' effectiveness in optimizing transportation processes to reduce costs. Additionally, artificial data instances reflecting higher demand than the company's current situation are tested in Experiments *E1* through *E4* to assess the solution methods' outcomes should the company face increased demands.

The first three experiments (*E1* through *E3*) do not allow routing for the common carriers, reflecting a real-world constraint faced by the company. In Experiment *E4*, common carriers are routed, modifying the MILP and VNS approaches from *Experiments E1* and *E3* to include this additional routing capability. This experiment will determine the extent to which routing common carriers affects the solutions and the KPIs.

5.1.1 Technical details

To perform the experiments and evaluate the data instances, a computer with Windows 10 Pro operating system, 16GB RAM, Intel(R) Core(TM) i7-8750H and 2.20GHz is used. For both VNS algorithm and the MILP model, Spyder 5.4.3 IDE is used integrating Python 3.11.5 64-bit. For the MILP, the MIP package of Python is used and the model is solved with the optimization tool Gurobi version 11.0.0 with an academic license.

5.2 Data Instances

Within this section, artificial data instances for parameter tuning (Section 5.2.1) are developed, together with real-world instances (Section 5.2.2). The data instances for parameter tuning consist of artificial scenarios that do not directly reflect the actual demand faced by Meilink Borculo B.V (Table 5). These instances feature a varied number of customers, ranging from 5 to 100, to demonstrate the solution methods' applicability to both smaller and larger demand cases. The use of artificial instances creates a controlled environment, facilitating the analysis of the VNS algorithm's response to diverse problem characteristics. This approach helps in identifying the algorithm's strengths and weaknesses across different instances, which may be hindered by the complexity and variability of real-world data. Moreover, evaluating the VNS across a range of artificial instances ensures the algorithm's robustness and its ability to generalize across various problem types, a critical aspect for real-world applications where problem characteristics can significantly differ. A well-tuned algorithm, based on artificial instances, can contribute to its effective performance on real-world problems, avoiding overfitting to a particular dataset. Hence, parameter tuning on artificial instances is crucial in optimizing algorithm performance by identifying the optimal parameter settings for a good solution quality and computational efficiency. This tuning process, detailed in Section 5.2.1, is exclusive to Experiment *E0*, focusing on the VNS model tuning and MILP model computation time.

Conversely, real-world instances (Section 5.2.2) validate the algorithm's practical applicability, examining the feasibility and efficiency of the proposed solutions within an operational context. These real-world instances reflect the actual daily pickup and delivery demands encountered by the company, incorporating actual time windows, locations, and vehicle specifications (Table 6). Additionally, four artificial instances are included within the real-world context to showcase potential future demand increases and their impact on company's KPIs. The evaluation of real-world instances, as outlined in Section 5.2.2, is used in Experiments *E1* to *E4*.

Furthermore, Section 5.2.3 elaborates on the cost structure of Meilink Borculo B.V.'s transportation challenge. These cost considerations are integral to both the parameter tuning phase and the evaluation of real-world instances, thereby influencing Experiments *E0* through *E4*.

5.2.1 Artificial data instances for parameter tuning

For the tuning of the Variable Neighbourhood Search (VNS) parameters, data instances ranging from 5 to 100 customer nodes are prepared, based on Meilink Borculo B.V.'s dataset for the years 2022-23. These artificial data instances are used only for Experiment *E0*.

The instances include randomly selected customer demand nodes, together with their linehaul and backhaul requests. Those do not represent the actual daily demands of the company, however, they are randomly chosen, and for larger number of customer nodes, demands from multiple days are merged to test the VNS's performance on more complex instances. The data contains client demands, their preferred time windows, and specific delivery and pickup locations from Meilink Borculo's records, although they do not exactly mirror the company's daily demands. The instances vary, with pickup nodes ranging from 0 to 47 and delivery nodes from 5 to 80. Client demands were randomly chosen from Meilink's dataset, and time windows were set according to the most commonly requested periods by clients.

The composition of the private fleet for each data instance reflects the actual fleet at Meilink, consisting of one box truck and four larger trucks, with their dimensions provided in the problem context analysis. Each instance ensures the combined capacity of the private fleet and common carriers meets all delivery and pickup requirements. The number of common carriers is adjusted based on the increase in customer nodes and demand levels.

Distances between demand nodes are determined utilizing the 'geodesic' method from the 'geopy' library within the Spyder environment. This method calculates the shortest path between two points on an ellipsoid's surface based on their latitude and longitude coordinates. The selection of this method ensures consistency across both the Variable Neighborhood Search (VNS) and Mixed Integer Linear Programming (MILP) models, where distances between nodes are computed similarly. Compared to the Euclidean distance, the 'geodesic' method offers enhanced accuracy by accounting for the Earth's curvature, a critical consideration for calculating longer distances. Nonetheless, it is important to acknowledge that this method does not guarantee absolute precision in reflecting the actual road distances between nodes.

Vehicle speed varies by instance, affecting the conversion of distances into travel time. For example, a speed factor of 0.02 means travel time is calculated as 0.02 times the distance in kilometers or in other words, it corresponds to a speed of 50 km/h. A comprehensive overview of the artificial data instances used for parameter tuning is provided in Table 5.

Table 5. Overview of the 10 artificial datasets for parameter tuning. Each dataset represents a day with number of different customer locations, number of backhaul customers, and the speed of the vehicles.

ID	Number of Customers	Number of Pickup Nodes	Speed
D1	5	0	0.02
D2	10	4	0.05
D3	20	5	0.1
D4	30	18	0.02
D5	40	12	0.05
D6	50	7	0.1
D7	60	0	0.02
D8	70	27	0.05
D9	80	47	0.1
D10	100	20	0.02

5.2.2 Real-world instances

To assess the optimal transportation strategy for Meilink Borculo B.V., 10 real-world instances are used from company's 2022-23 data, which includes daily customer demands, pickup requests, time windows, vehicle capacities, and associated costs (Section 5.2.2).

In this context, "demand" refers to the number of distinct customers requiring service within a single day. Demand varies both daily and seasonally, with certain days of the week and months of the year experiencing peak or significantly lower demands. Given the daily execution of transportation, instances are tailored to specific weekdays, reflecting anticipated customer delivery dates.

Three weekdays are selected for detailed evaluation: Monday, with typically the lowest demand; Thursday, experiencing the highest demand; and Friday, showing moderate demand levels throughout the year. To accurately represent variability, both a low-demand and a high-demand Monday are chosen from the dataset, ensuring they reflect typical demand patterns without being extreme outliers. This approach is similarly applied to Thursdays and Fridays.

As outlined in Section 2, there is considerable variation in both pickups and deliveries on any given day. Consequently, the instances include diverse pickup and delivery nodes to accurately mirror actual demand. Each dataset's total demand nodes, and whether they include pickup nodes, are detailed in Table 6, noting that high customer numbers on certain days might not equate to a high total order volume (and vice versa) due to product and packaging diversity.

Given the real-world instances involve only 7 to 46 customer nodes, four artificial instances are developed and evaluated. These artificial instances, created from the Meilink datasets by merging the customer demand over several days, help explore potential impacts on KPIs if the company experiences growth and an increase in daily orders.

A comprehensive overview of all instances is provided in Table 6. The real-world instances which are representative of the actual demand of the company are *S1-S6* and the artificial ones are *AS7-AS10*. Each instance undergoes analysis using both Mixed Integer Linear Programming (MILP) and Variable Neighborhood Search (VNS) under Experiments *E1* to *E4*.

Table 6. Overview of real-world daily data instances for Meilink Borculo B.V.'s transportation problem. Each instance is a specific day of the week, where a certain level of demand is observed. The customer nodes vary per day and insome days there are both linehaul and backhaul customers.

ID	Day of the Week	Demand	Customer Nodes	Pickup
S1	Thursday	low	11	Yes
S2	Thursday	high	46	Yes
S3	Friday	low	10	Yes
S4	Friday	high	37	No
S5	Monday	low	7	No
S6	Monday	high	34	Yes
AS7	-	-	60	No
AS8	-	-	70	Yes
AS9	-	-	80	Yes
AS10	-	-	100	Yes

5.2.3 Cost data for numerical experiments

The cost breakdown[2] for each vehicle of Meilink Borculo B.V. is detailed in Table 7, with data sourced from 2022. As described in Section 2.3, the fixed costs associated with using a vehicle include employee wages, depreciation, maintenance, insurance, and taxes. It is assumed that each vehicle is driven by one employee, so the average wage for an employee is incorporated into the fixed costs for each vehicle. Notably, some vehicles have reached the end of their depreciation period, resulting in a depreciation value of zero. Maintenance costs, which varied for each lorry in 2022, are summed up annually. The maintenance costs for the six trailers are also aggregated and then evenly distributed across the four lorries. The total maintenance costs per vehicle are presented in the fourth column of the table. The box truck does not make use of the trailers and, therefore does not incur the maintenance costs associated with them. After calculating the total annual fixed costs per vehicle, these costs are divided by 260 days – the number of working days from Monday to Friday in a year – to determine the daily fixed costs per vehicle, as shown in column six.

Fuel is considered a variable cost. To calculate fuel consumption for each vehicle, information from the respective vehicle brand and model websites was utilized. The average between the minimum and maximum fuel consumption per vehicle is computed and is presented in the table, since it is very hard to estimate the exact consumption of a vehicle, as it changes based on many factors, including loading, speed, traffic conditions, and the age of the vehicle. The fuel price per litre for Diesel cars within the Netherlands is taken as 1.504 euros average for the year 2022-23. The cost per km travelled per vehicle is found by the consumption rate of a vehicle times the price of fuel. Those are presented in column 7.

[2]These cost does not represent the actual costs of the business. All costs are multiplied by a random non-integer number to maintain the confidentiality of the business. All costs in subsequent pages are also be multiplied by the same random number.

Table 7. Cost breakdown of company's private vehicles.

ID	Model	Depreciation	Maintenance	Consumption	FC	VC
1	Mercedes Benz	13559.2	5330.34	20.42	210.68	0.307
2	Mercedes Benz	13559.2	5071.56	20.42	209.68	0.307
3	Volvo	6320	2455.03	16.69	170.49	0.251
4	Mercedes Benz	0	8633.86	20.42	171.99	0.307
5	Mercedes Benz	0	6948.66	20.42	165.51	0.307

The common carriers usually charge a fixed fee based on the customer node they are visiting. However, in case there are more than two customers who can be served on the same trip by common carriers, the cost for the furthest customer applies. However, the third-party logistics has not provided the fee for all the possible destinations that can be visited by them. Since this is the case, a cost per kilometre charged by the common carriers has to be estimated to make it feasible for the model and not restrict the common carriers to visit only specific cities. Therefore, the distance between the Borculo location and all the cities in the provided list by the common carrier is found by using the 'geodesic' method from the 'geopy' library in Spyder. After finding the geodesic distance between the nodes in the common carrier list, the fixed fee for each node is divided by the distance between the Borculo and that node. Then the found costs per km are averaged and the approximate cost per km charged by common carriers is approximated as €2.778.

5.3 Parameter Tuning

In this section the parameters of the VNS are tuned and MILP's maximum computation time is set by using the artificial data instances as described in Section 5.2.1. Firstly, the initialisation method used for the VNS is determined, followed by the maximum number of iterations without an improvement of the algorithm is set. Moreover, the order of the local search operators is chosen, as well as the strategy to be used in the shaking phase. These parameters are tested against multiple predefined data instances. Besides, the MILP's maximum computational time is determined to find an optimal solution within reasonable time.

5.3.1 Initialisation

As described in the solution methodology (Section 4.4) the initialisation can be performed in two ways. The first method assigns vehicles based on their order in the vehicle list. The second method involves randomizing the selection of vehicles from the list, enabling the solution to choose randomly the vehicle, either a private or external, from the list of vehicles. The choice of initial solution is crucial for selecting the vehicles used to transport goods to and from customers, as it influences the range of possible outcomes when applying the Variable Neighborhood Search (VNS). Consequently, determining the most effective initialization method between the two is essential and is achieved through experimentation:

Experiment I1: Sequential Fleet Allocation. This initialization method consistently produces the same result due to the ordered assignment of vehicles. It is conducted only once in the experiments.

Experiment I2: Randomized Vehicle Selection. This method of initialization yields varying results since it assigns vehicles randomly each time. Therefore, it is conducted 10 times to ensure the output is not an extreme outlier, which is then compared to *Experiment I1*. The minimum, maximum, and average values of the experiment for each data instance are reported, but only the minimum is compared to *I1*.

The experiments test the solution on all possible ways of assigning the vehicles. The results from the experiment are provided in Table 8. The first column provides the artificial data instances ID experimented with, the second column is split into cost and time, where the costs generated from *Experiment I1* and the associated computation time are provided. The last columns provide the minimum, average and maximum costs and time values from *Experiment I2* with each data instance.

Table 8. Comparison of initialization methods. The lowest cost objective value and lowest computation time compared between *I1* and *I2* are given in bold.

ID	*I1*		*I2*					
	Cost	Time	Cost			Time		
			Min	Avg	Max	Min	Avg	Max
D1	252.42	0.008	**207.25**	252.42	377.53	**0.008**	0.014	0.039
D2	509.14	0.004	**442.14**	621.64	802.90	**0.002**	0.008	0.010
D3	**1873.74**	0.049	1927.80	2387.45	2622.41	**0.02**	0.023	0.031
D4	2635.28	0.09	**2117.74**	2532.86	2693.80	**0.02**	0.028	0.032
D5	6468.85	0.142	**5291.71**	5427.52	5977.94	**0.092**	0.111	0.142
D6	6358.44	0.085	**4438.11**	5799.69	6224.0	**0.043**	0.058	0.091
D7	7333.84	0.264	**5819.73**	6509.67	6740.73	**0.167**	0.188	0.202
D8	7061.73	0.678	**5764.70**	6622.28	6885.07	**0.303**	0.459	0.849
D9	8127.69	0.377	**6428.92**	6778.49	6944.47	**0.213**	0.256	0.313
D10	9421.88	0.811	**7134.84**	8611.49	9809.08	**0.344**	0.393	0.554
Average	5004.3	0.251	**3957.29**	4554.35	4907.79	**0.121**	0.154	0.226

Table 8 reveals a variable performance comparison between the two initialisation experiments across the data instances. The Sequential Fleet Allocation (*Experiment I1*) and Randomized Vehicle Selection (*Experiment I2*) methods offer distinct approaches to initializing the VNS algorithm, impacting the solution's quality and the time required to reach it.

The Randomized Vehicle Selection method (*Experiment I2*) consistently outperforms the Sequential Fleet Allocation method (*Experiment I1*) in terms of achieving lower cost solutions across almost all data instances, as evidenced by the minimum cost values being lower for *Experiment I2* in the majority of cases. This suggests that introducing randomness in vehicle selection allows the VNS algorithm to explore a more diverse set of initial solutions, increasing the likelihood of finding more cost-effective routes. For instance, in data instance *D1*, the minimum cost achieved through *Experiment I2* is significantly lower than the cost from *Experiment I1*, highlighting the potential benefits of a randomized approach to vehicle assignment.

Furthermore, the Randomized Vehicle Selection method also shows advantages in computational efficiency, with lower minimum computation times reported for several data instances. This indicates that not only does the randomized approach potentially lead to better solutions, but it can also do so in a more time-efficient manner. For example, in data instances *D2* and *D5*, the minimum computation times for *Experiment I2* are notably lower than those for *Experiment I1*, suggesting randomized approach offers quicker allocation of vehicles to demand nodes.

Lastly, the Randomized Vehicle Selection method (*Experiment I2*) appears to be a better initialization strategy for the VNS algorithm in the context of this thesis. Its ability to consistently achieve lower costs and, in many cases, do so more quickly than the Sequential Fleet Allocation method (*Experiment I1*), makes it a more effective approach for initializing the VNS algorithm. This analysis suggests that incorporating randomness into the initial vehicle selection process can significantly enhance the performance of metaheuristic algorithms in solving Meilink's problem.

5.3.2 Number of iterations without improvement

The number of iterations without an improvement is a crucial parameter in the Variable Neighborhood Search (VNS) algorithm, significantly impacting the optimality of the result it produces. Higher number of iterations does not necessarily guarantee better results; instead it could lead to increased computational time, making the solution method inefficient. Thus, carefully selecting the number of iterations allows for a balance between computation time and quality of the produced results.

To explore the optimal number of iterations, a series of experiments are conducted, as detailed below. These experiments range from 10 to 100 iterations, with each experiment applied to the artificial data instances outlined in Section 5.2.1. The initial solutions for these experiments are generated using the Randomized Vehicle Selection method. From these, the best cost solution across 100 iterations of the initial solution is identified and utilized for subsequent experimentation on iteration numbers. Given the inherent randomness in the VNS's shaking phase, each experiment-data instance combination is executed

10 times. The maximum improvement, leading to the lowest cost objective value across these runs is calculated and presented in Table 10.

Experiment I1: This experiment limits the VNS to a maximum of 10 iterations without improvement.
Experiment I2: The maximum number of iterations is set to 25 in this experiment without improvement.
Experiment I3: This setup increases the limit to 40 iterations without improvement.
Experiment I4: In this experiment, the iteration cap is extended to 75 without improvement.
Experiment I5: The final experiment allows up to 100 iterations without improvement.

Table 9. Best improvements and time for Experiments N. In bold is the best improvements, percentage wise, out of the 10 runs per experiment. The times are in seconds and represent the computation time for the run with the highest improvement.

ID	Initial objective	*I1*		*I2*		*I3*		*I4*		*I5*	
		Improvement (%)	Time (s)	Improvement (%)	Time (s)	Improvement (%)	Time (s)	Improvement (%)	Time (s)	Improvement (%)	Time (s)
D1	204.6	5.10	1.67	5.10	4.21	5.10	6.47	5.16	12.16	**6.70**	16.19
D2	426.28	8.29	2.30	8.27	5.19	9.03	7.79	9.19	14.12	**9.67**	19.74
D3	1956.32	41.74	3.06	42.10	6.61	44.06	7.74	44.73	14.52	**44.95**	21.02
D4	1656.94	32.30	2.22	34.31	5.66	35.24	7.77	38.15	13.75	**40.29**	23.01
D5	5541.31	46.41	2.98	53.21	6.00	56.56	8.69	56.65	15.74	**58.80**	19.47
D6	5656.29	57.27	3.29	57.88	6.23	60.16	8.04	63.85	15.71	**64.89**	19.95
D7	6095.43	45.61	3.02	47.72	6.04	49.28	9.92	51.66	15.34	**53.05**	19.87
D8	5275.4	36.68	3.92	39.96	6.49	40.14	9.38	41.41	15.75	**41.46**	20.54
D9	6178.05	43.11	3.08	43.11	6.88	43.16	9.15	44.59	16.97	**44.59**	20.77
D10	9075.35	43.62	3.27	45.50	19.58	46.25	19.66	46.88	21.97	**47.23**	27.52
Average	4206.6	36.01	2.88	37.72	7.29	38.90	9.46	40.23	15.60	**41.16**	20.81

The data in Table 10 gives an overview of the initial solution generated for the VNS and the best improvement percentage for each pair of experiment-data instance out of the 10 runs. It indicates that increasing the number of iterations generally leads to better solution quality, as indicated by the higher percentage of improvement in the objective value. This trend is consistent across all data instances, with the highest average improvement observed in Experiment *I5*, where the maximum number of iterations without improvement was set to 100. This suggests that allowing the algorithm more iterations to explore the solution space can indeed yield better solutions.

However, this improvement in solution quality comes at the cost of increased computational time. As the maximum number of iterations without improvement rises, so does the time required to complete the experiment. This increase is substantial, moving from an average of approximately 2.88 seconds in Experiment *I1* (10 iterations) to 20.81 seconds in Experiment *I5* (100 iterations). This highlights a trade-off between solution quality and computational efficiency.

Given these observations, the choice of the maximum number of iterations without improvement should be guided by this thesis problem requirements. Given that the priority is to achieve the best possible solution within 30 minutes of computation time, it became evident that the VNS is quite robust in that sense and the maximum computation time per run is not more than 30 seconds with 100 iterations in Experiment *I5*. Although setting the number of iterations without improvement to 100 increases the computation time of the algorithm with a third, it improves on average the objective only by 1%. It is questionable whether setting the number of iterations of a VNS to 100 is reasonable, especially in instances with high customer demand.

While higher iteration counts can yield improved solutions, the implications on computational time must be weighed carefully. Setting the number of iterations around 40 as observed in Experiment *I3*, could serve as an effective compromise between solution quality and computational efficiency for various practical scenarios. This number seems to provide substantial improvements over the lower iteration counts while keeping the increase in computational time within reasonable limits within high demand instances. For this thesis, however, the VNS algorithm's maximum number of iterations without improvement is established at 100 (Experiment *I5*), since the computation time per run remains below 30 seconds, well within the required maximum of 30 minutes.

5.3.3 Order of local search operators

Thought the previous experiments, the order of operators in the local search phase is 'Swap vehicles' ! '2-Opt' ! 'Swap customers' ! 'Move' ! 'Reinsertion'. However, the order in which the local search operators are applied in the VNS can make a difference in terms of both quality of the solution and the computational time required. The choice of the operators order determines the sequence in which different explorations and perturbations are performed on the initial and current solution during the search process. Therefore, the arrangement of operators affects both the efficiency of the algorithm and the ability to produce high-quality results.

In Section 4.4.5 five operators are provided and explained. Those include swap, reinsertion, 2-opt, move, and swap vehicles. If all the possible ways to order those five operators are tested, this will result in 5! or 120 unique experiments. This will require a very extensive time to be spent on evaluating the experiments. To address this challenge, an alternative approach to determine the optimal operator order is adopted. The strategy involves initially evaluating each operator independently, running 100 iterations for each (in alignment with the chosen VNS iteration count). Based on the evaluations, the first operator in the order is selected, based on its historical performance—specifically, the one exhibiting the highest improvement in terms of cost.

This selection of an operator leverages past performance data to prioritize the most promising operator as the initial step. Starting with the operator, which has shown the highest improvement on its own, enhances the probability of generating a high-quality solution. This operator is expected to make substantial improvements to the current solution early in the search, potentially leading to faster convergence and shorter computation times. It acts as an efficient starting point for the search process.

The experiments for evaluating each operator individually are shown below. It is chosen to evaluate those on only five of the data instances *D2, D4, D6, D7, and D9* for better manageability purposes and easier evaluation of the experiments. These instances are chosen due to their diversity in terms of factors such as the demand, pickup and delivery nodes, distances, time windows, and vehicle speed. The Randomized Vehicle Selection initial solution is ran five times and the solution with best costs from it is chosen for testing the operators' performance. Each of the operators is ran for 100 iterations. The outcomes of these experiments are summarized in Table 10.

Experiment O1: In this experiment the 'Swap' operator is used.
Experiment O2: In this experiment the 'Reinsertion' operator is used.
Experiment O3: In this experiment the '2-Opt' operator is used.
Experiment O4: In this experiment the 'Move' operator is used.
Experiment O5: In this experiment the 'Swap vehicles' operator is used.

Table 10. Improvements and time for Experiments O. In bold are the best improvements per data instance, percentage wise.

ID	Initial objective	*O1*		*O2*		*O3*		*O4*		*O5*	
		Improvement (%)	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**
D2	426.28	0.00	0.00	0.00	0.005	0.00	0.004	0.00	0.001	**21.32**	0.005
D4	1656.94	0.00	0.0002	0.00	0.003	0.00	0.002	0.00	0.001	**10.54**	0.006
D6	5656.29	0.00	0.0002	0.00	0.002	0.00	0.002	**6.16**	0.002	3.30	0.004
D7	6095.43	0.00	0.001	0.00	0.003	0.00	0.002	**6.91**	0.002	3.96	0.005
D9	6178.05	0.00	0.001	0.03	0.003	0.00	0.003	0.00	0.001	**6.07**	0.006
Average	4002.59	0.00	0.0004	0.01	0.003	0.00	0.003	2.61	0.001	**9.04**	0.005

The results in Table 10 show that *Experiment O5*, 'Swap vehicles' operator, has the largest average improvement for data instances *D2, D4, and D9*. For instances *D6, and D7 Experiment O4* generates the highest improvement. In order to determine which experiment shows the highest improvement, the average over all data instances is computed and can be seen in the last row of Table 10. Judging by the overall average, *Experiment O5* has the highest average improvement percentage (9.04%) across the selected data instances. Operator 'Swap vehicles' appears to be an efficient starting point for quickly enhancing the initial solution. Moreover, the computation times are very low, indicating efficiency of the operator.

Since *Experiment O5* generated the best average improvement over the selected instances, the 'Swap vehicles' operator is selected as initial operator. To select the order of the other four operators, 24 different experiments will be performed to test all possible orders and evaluate the improvements those generate. Once again, the Randomized Vehicle Selection initial solution is ran five times and the solution with best costs from it is chosen for testing the operators' performance. Each of the below-described experiments is ran for 100 iterations to ensure consistency. The results of those are presented in Table 11.

Experiment L1: Swap vehicles ! Move ! Swap ! 2-opt ! Reinsertion.
Experiment L2: Swap vehicles ! Move ! Swap ! Reinsertion ! 2-opt.
Experiment L3: Swap vehicles ! Move ! 2-opt ! Swap ! Reinsertion.
Experiment L4: Swap vehicles ! Move ! 2-opt ! Reinsertion ! Swap.
Experiment L5: Swap vehicles ! Move ! Reinsertion ! Swap ! 2-opt.
Experiment L6: Swap vehicles ! Move ! Reinsertion ! 2-opt ! Swap.
Experiment L7: Swap vehicles ! Swap ! Move ! 2-opt ! Reinsertion.
Experiment L8: Swap vehicles ! Swap ! Move ! Reinsertion ! 2-opt.
Experiment L9: Swap vehicles ! Swap ! 2-opt ! Move ! Reinsertion.
Experiment L10: Swap vehicles ! Swap ! 2-opt ! Reinsertion ! Move.
Experiment L11: Swap vehicles ! Swap ! Reinsertion ! Move ! 2-opt.
Experiment L12: Swap vehicles ! Swap ! Reinsertion ! 2-opt ! Move.
Experiment L13: Swap vehicles ! 2-opt ! Move ! Swap ! Reinsertion.
Experiment L14: Swap vehicles ! 2-opt ! Move ! Reinsertion ! Swap.
Experiment L15: Swap vehicles ! 2-opt ! Swap ! Move ! Reinsertion.
Experiment L16: Swap vehicles ! 2-opt ! Swap ! Reinsertion ! Move.
Experiment L17: Swap vehicles ! 2-opt ! Reinsertion ! Move ! Swap.
Experiment L18: Swap vehicles ! 2-opt ! Reinsertion ! Swap ! Move.
Experiment L19: Swap vehicles ! Reinsertion ! Move ! Swap ! 2-opt.
Experiment L20: Swap vehicles ! Reinsertion ! Move ! 2-opt ! Swap.
Experiment L21: Swap vehicles ! Reinsertion ! Swap ! Move ! 2-opt.
Experiment L22: Swap vehicles ! Reinsertion ! Swap ! 2-opt ! Move.
Experiment L23: Swap vehicles ! Reinsertion ! 2-opt ! Move ! Swap.
Experiment L24: Swap vehicles ! Reinsertion ! 2-opt ! Swap ! Move.

Table 11. Improvements and time for Experiments L. In bold is the best improvement per data instance and over all instances.

Experiment	*D2*		*D4*		*D6*		*D7*		*D9*		**Average**
	Initial cost	461.13	Initial cost	1656.94	Initial cost	5656.29	Initial cost	6095.42	Initial cost	6178.05	
	Improvement (%)	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**	**Improvement (%)**	**Time (s)**	
L1	2.41	0.016	3.35	0.019	2.58	0.022	2.72	0.017	3.83	0.020	2.98
L2	2.34	0.015	**6.02**	0.021	2.59	0.017	2.80	0.019	2.09	0.018	3.17
L3	0.97	0.015	3.59	0.015	1.40	0.019	2.41	0.015	3.15	0.023	2.30
L4	2.29	0.015	3.92	0.016	2.26	0.019	2.72	0.018	3.91	0.021	3.02
L5	1.09	0.014	3.70	0.017	2.47	0.017	2.82	0.016	3.06	0.019	2.63
L6	3.40	0.015	1.89	0.018	1.99	0.019	3.64	0.016	3.70	0.019	2.92
L7	2.51	0.014	3.56	0.015	**5.16**	0.020	3.38	0.016	3.59	0.021	3.64
L8	1.67	0.014	3.67	0.016	3.52	0.017	2.81	0.017	3.18	0.023	2.97
L9	2.05	0.016	3.77	0.018	2.79	0.014	**3.90**	0.0199	3.08	0.018	3.12
L10	2.39	0.015	3.96	0.016	1.69	0.016	3.89	0.035	4.52	0.022	3.29
L11	2.31	0.022	3.41	0.019	3.62	0.021	2.54	0.034	3.94	0.021	3.16
L12	0.13	0.020	2.76	0.016	3.41	0.018	2.38	0.033	3.50	0.018	2.44
L13	3.47	0.022	5.94	0.015	2.81	0.016	3.25	0.040	4.07	0.020	3.91
L14	2.32	0.024	5.71	0.0159	2.05	0.017	3.23	0.034	2.16	0.022	3.09
L15	**4.71**	0.021	5.39	0.017	3.46	0.020	2.87	0.036	3.46	0.018	**3.98**
L16	1.34	0.020	4.00	0.018	3.19	0.018	2.95	0.034	3.93	0.017	3.08
L17	1.17	0.019	3.89	0.018	0.66	0.018	2.83	0.021	3.49	0.021	2.41
L18	2.18	0.019	0.83	0.016	3.21	0.017	2.29	0.020	2.60	0.019	2.22
L19	2.18	0.020	3.00	0.016	0.69	0.017	2.09	0.020	**4.54**	0.020	2.50
L20	3.62	0.019	2.42	0.015	2.22	0.019	3.06	0.030	2.73	0.018	2.81
L21	4.70	0.021	2.41	0.019	3.10	0.017	3.33	0.022	2.92	0.019	3.29
L22	2.31	0.024	3.42	0.019	3.41	0.022	1.56	0.031	1.33	0.017	2.41
L23	2.06	0.201	3.02	0.021	1.39	0.020	3.84	0.044	3.45	0.019	2.75
L24	1.80	0.023	3.82	0.020	2.32	0.017	1.61	0.040	4.03	0.021	2.72

Table 11 provides a comprehensive overview of the results for various combinations of experiments and data instances. The cost improvements are presented as percentages, representing the average improvement observed over 25 iterations for each combination. Additionally, the reported time reflects the average execution duration for a single iteration of each experiment.

In general, there isn't a single experiment that consistently outperforms others across all data instances. For instance, *Experiment L2* yields the most remarkable results, demonstrating a significant 6.02% improvement in *D4*, establishing it as the optimal choice of operators for this particular dataset. However, *Experiments L13* and *L14* also exhibit substantial improvements for *D4*.

For *D6*, the experiments *L7* and *L8* generate the most substantial improvements, while for data instance *D7* Experiments *L9* and *L10* have the best cost improvement. Moreover, *Experiment L10* achieves a significant improvement for *D9*, while *Experiment L15* performs best for data instance *D2*. Due to the variability and inconsistency in experiment performance across data instances, the average improvement for each experiment over all considered datasets is calculated, and these results are presented in the last column of Table 11.

Based on the average, the highest improvement over all data instances is 3.98% and it is produced by *Experiment L15*, followed by the second-best performer with 3.91% of *Experiment L13*.

5.3.4 Adaptive VNS

For all the previous experiments, a random operator is chosen within the shaking phase. However, in certain cases, using Adaptive Variable Neighbourhood Search (AVNS) may improve the objective value and lead to higher improvement. This method is a variation of the standard Variable Neighbourhood Search in a sense that it adapts dynamically the search in the shaking phase in order to improve the ability of the algorithm in finding high-quality solutions. In the shaking phase, AVNS selects a shaking operator and applies it to the current solution to generate a new solution, which helps the algorithm escape from the local optima. When an improvement is not found, it switches the operator every 25% of the maximum iterations (100 iterations). The AVNS is tested with a predefined order of operators, which showed to generate highest improvement in the LS phase of the algorithm, namely 'Swap vehicles' ! '2-Opt' ! 'Swap customers' ! 'Move' ! 'Reinsertion'.

To test the effectiveness of using AVNS in the solution generation, several experiments are defined, where each experiment aims to modify the current solution in a search for better routes. The experiments are listed below (*Experiment A1-A6*), where *Experiment A1* uses the adaptive shaking phase that applies the operators from *Experiment L15*. Experiments *A2* through *A6* are using a single shaking operator through the search process. The experiments are tested on some of the artificial data instances (*D2, D4, D6, D7, D9*) by generating an initial solution through running the Randomized Vehicles Selection for 100 iterations and storing the best solution. Then, each experiment is run 10 times with 100 iterations without improvement to keep it consistent with the previously tuned parameters.

Experiment A1: Adaptive shaking phase: 'Swap vehicles' ! '2-Opt' ! 'Swap customers' ! 'Move' ! 'Reinsertion'
Experiment A2: Only the 'Swap vehicles' operator is used.
Experiment A3: Only the 'Move' operator is used.
Experiment A4: Only the 'Reinsertion' operator is used.
Experiment A5: Only the 'Swap customers' operator is used.
Experiment A6: Only the '2-opt' operator is used.

Table 12. Average improvement percentages from Experiments A over 10 runs with 100 iterations. The highest improvement percentage per data instance is given in bold.

ID	**Initial objective**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**
D2	423.44	**9.43**	7.87	7.87	9.04	7.85	9.41
D4	2121.69	**47.78**	36.22	37.49	44.98	37.90	45.79
D6	4669.66	**60.81**	51.36	46.79	55.03	49.60	54.35
D7	6984.54	**52.80**	44.72	45.01	51.52	43.87	46.29
D9	6912.25	**49.17**	39.24	40.21	42.32	41.11	42.95
Average	4222.32	**44.00**	35.88	35.47	40.58	36.07	39.76

The results from the experiments are summarized in Table 12 for each data instances and in the last row the average over all artificial data instances is computed. Adaptive Shaking Phase (*Experiment*

A1) consistently outperforms the single-operator experiments across all artificial data instances, with the highest average improvement percentage of 44.00%. This indicates the effectiveness of dynamically switching operators based on the stage of the iteration process, allowing the algorithm to escape local optima more effectively.

Single-operator experiments show varying performance, with 'Swap vehicles' (*A2*) generally performing the least effectively and '2-opt' (*A6*) being the most effective single operator, but still less effective than the adaptive strategy.

The adaptive shaking phase (*Experiment A1*) demonstrates a clear advantage in leveraging multiple neighborhood structures to enhance the search process's exploratory capabilities, outperforming the single operators with at least 5% higher improvement on average. By adapting the operator based on the search's progress, AVNS can more effectively navigate the solution space, leading to higher quality solutions. Therefore, for Experiment *E1* to *E4*, the adaptive shaking phase is utilized with a predefined sequence of operators ('Swap vehicles' *!* '2-Opt' *!* 'Swap customers' *!* 'Move' *!* 'Reinsertion'). If no improvement is found within each 25% of the maximum iterations (100 iterations in this case), AVNS switches to the next operator in the sequence.

5.3.5 Maximum running time of MILP

To identify a (near) optimal solution without requiring extensive computation time, it is crucial to establish an appropriate maximum runtime for the MILP. The following experiments are designed to explore the balance between solution optimality and computation time. Three distinct experiments—*M1*, *M2*, and *M3*—with varying computation times from 3 minutes to 1 hour, are conducted.

Experiment M1: considers a short computation time of 3 minutes (180 seconds).
Experiment M2: allocates a moderate computation time of 30 minutes (1800 seconds).
Experiment M3: allows for an extended computation time of 60 minutes (3600 seconds).

Table 13. Comparison of objective value, computation time, and optimality gap across models M1, M2, and M3. The average best results are given in bold.

ID	M1			M2			M3		
	Result	**Time (s)**	**Gap**	**Result**	**Time (s)**	**Gap**	**Result**	**Time (s)**	**Gap**
D1	189.97	0.08	0.00%	189.97	0.08	0.00%	189.97	0.08	0.00%
D2	358.07	1.09	0.00%	358.07	1.09	0.00%	358.07	1.09	0.00%
D3	770.68	104.09	0.00%	770.68	104.09	0.00%	770.68	104.09	0.00%
D4	1203.17	1800	2.36%	1119.85	1800	0.66%	1119.85	3600	0.59%
D5	3091.95	1800	7.29%	3060.98	1800	1.30%	3060.98	3600	1.29%
D6	2085.18	1800	22.19%	2081.78	1800	2.72%	2081.78	3600	2.72%
D7	4672.9	1800	13.49%	4666.44	1800	0.12%	4666.45	3470.42	0.00%
D8	7546.95	1800	10.77%	7544.47	1800	5.98%	7539.86	3600	0.79%
D9	9086	1800	9.03%	9084.4	1800	0.98%	9084.4	3600	0.98%
D10	13179.28	1800	6.16%	13163.44	1800	4.19%	13161.68	3600	4.19%
Average	4218.42	1360.53	7.13%	**4211.51**	**4210.87**	**1.60%**	4210.87	2517.57	1.05%

The outcomes of these experiments are detailed in Table 13, which includes the cost results, performance gaps, and computation times for each experiment-data instance combination. For data instances *D1*, *D2*, and *D3*, the results are consistent across all experiments, showing no performance gap. Thus, even the shortest computation time of 180 seconds (*M1*) suffices to achieve an optimal solution for these instances. For data instances *D4* through *D10*, the performance gap narrows as computation time increases, indicating that longer computation times enable the MILP to approach the optimal solution more closely, especially in instances with a larger number of customer nodes. In specific cases, like *D7*, the extended computation time in *Experiment M3* (3470.42 seconds) closes the performance gap (0.00%), suggesting that the optimal solution has been attained. This demonstrates the model's complexity and indicates that for data instances with a significant number of pickups, a (nearly) optimal solution can be achieved within a shorter computation time.

In general, while a 3600 seconds computation time can improve the solution quality and even close the optimality gap for certain data instances, the computation time is very high, given that it does not close that gap for all instances. For a company like Meilink, it's essential to perform routing within a feasible time of 30 minutes. Given that the average costs of Experiment *M2* is 4211.51 and the gap is 1.60% , it is more reasonable to set the maximum computation time to 1800s as the optimality gap in Experiment *M3* decreases with only 0.55% and the average computation time in the same experiment doubles.

5.3.6 Final parameters

The experiments detailed in Section 5.3 are crucial for the final tuning of the VNS and determining the maximum computation time of MILP. By sequentially fine-tuning the parameters, the most effective settings are identified, which were then applied in subsequent experiments. For the initialization of the VNS, the Randomized Vehicle Selection will be executed for 100 iterations, retaining the minimum cost solution for further stages of the algorithm. It became clear that setting the maximum number of iterations to 100 offers the best improvement, and since the computation time of the VNS is only up to 30 seconds for the highest demand instances, the maximum number of iterations is set. The 'Swap vehicles' operator emerged as the most effective for the local search phase and, therefore, is prioritized in the sequence of operators. After evaluating various sequences, the order that provided the highest improvement was Swap vehicles ! 2-opt ! Swap ! Move ! Reinsertion. Consequently, this sequence will be adopted. Furthermore, the Adaptive VNS approach will be utilized, as it demonstrated superior improvement potential, reducing the need to rely on a single shaking operator. When no improvement is found within each 25% of the maximum iterations (100 iterations in this case), AVNS switches to the next operator in the sequence of operators, which is the same as the one for LS.

Regarding the MILP's maximum runtime, 1800 seconds are allocated. While this duration does not guarantee optimal solutions for all data instances, it produces promising outcomes, particularly for instances with fewer customer nodes. An extended computation time of 3600 seconds did not significantly enhance the solutions' optimality across all instances, nor did it justify the doubled computational effort. Therefore, the 1800-second timeframe is considered the most practical, given also the requirements of the company.

5.4 Scenario Evaluation

This section presents the outcomes of applying MILP and VNS across the real-world data instances as presented in Section 5.2.2 within Experiments *E1* to *E4*. Initially, the mixed fleet utilization scenario (*E1*) is explored, followed by scenarios focusing exclusively on the private fleet (*E2*) and on common carriers (*E3*), where only private fleet vehicles are routed. Subsequently, the scenarios under *E4* are examined, where both vehicle types are routed in the MILP and VNS models. The section concludes by comparing the two solution methods, highlighting their similarities, differences, and the implications of their results.

5.4.1 Mixed Fleet Utilization

Experiment *E1* assesses the performance of mixed fleet utilization, where both private fleet vehicles and common carriers are available to meet demand nodes, but only private fleet vehicles are routed. This setup mirrors Meilink Borculo B.V.'s current operational model, where internal vehicles are routed, and external carriers are utilized in high-demand instances. The MILP and VNS are tested for the ten instances outlined in Section 5.2.2 and their results are presented in Table 14 and 15. The parameter settings as defined in the parameter tuning apply.

The results from Experiment *E1* using MILP, detailed in Table 14 for each instance, include the data instance IDs and labels, with labels indicating the day of the week and demand level (e.g., TH = Thursday, high demand). The third column lists the number of demand nodes, while subsequent columns present the model's costs ('Objective'), computation time in seconds, gap to optimality in percentage, and the total distance traveled by all vehicles. Additional columns provide insights into the number of nodes served by private fleets versus common carriers, the IDs of private vehicles utilized, and the percentage of unserved nodes.

Table 14. Summary of MILP results for each data instance with mixed fleet utilization (Experiment *E1*).

ID	Label	Nodes	Objective	Time (s)	Gap (%)	Distance (km)	Nodes with CC	Nodes with PF	PF ID	Unserved %
S1	TL	11	252.04	0.21	0.00	113.82	1	10	3	0
S2	TH	46	5277.41	1800.00	1.36	2143.11	18	28	all	0
S3	FL	10	375.56	0.05	0.00	329.02	2	8	3	0
S4	FH	37	1062.33	1800.00	1.98	826.16	1	36	all	0
S5	ML	7	385.07	0.10	0.00	364.42	4	3	3	0
S6	MH	34	3938.93	1800.00	32.53	1874.36	12	22	all	0
AS7	-	60	11867.24	1800.00	31.45	4411.21	36	24	all	0
AS8	-	70	13944.47	1800.00	34.68	5422.81	51	19	all	0
AS9	-	80	12284.40	1800.00	26.91	5480.72	44	36	all	0
AS10	-	100	22763.44	1800.00	34.27	8558.39	72	28	all	0
Average	-	-	7215.09	1260.04	16.28	2952.89	24	22	-	0

The MILP model demonstrated efficiency in solving low-demand instances within reasonable computation times. However, as the data complexity increases — from *S1*, with 11 nodes, to *AS10*, with 100 nodes—a significant rise in the objective cost is observed, increasing from 252.04 to 22763.44. This substantial cost increase primarily results from the larger distance traveled by all vehicles, which ranges between 113.82 km and 8558.39 km. Such findings underscore the model's scalability and its adeptness at managing larger datasets and more complex instances.

In the MILP analysis, instances *S1*, *S3*, and *S5* reached optimality. Conversely, for the remaining instances, the MILP failed to identify the optimal solution within the allocated 1800 seconds, exhibiting an average optimality gap of 16.28%. This indicates the MILP model's limitations in achieving optimality within constrained computation times, particularly for increased number of data instances.

For instances characterized by low demand, the MILP model predominantly selects vehicle ID 3 from the private fleet to fulfill most customer demands, and relies on common carriers for the unserved nodes. An overview of the nodes, served predominantly by common carriers, is provided in Appendix B.1. During high-demand instances, the model engages all five vehicles from the private fleet. Given the capacity and time constraints coupled with a high number of customer nodes, these five vehicles prove insufficient for servicing all nodes. Consequently, as the demand points increase, reliance on external fleets by the MILP increases as well.

The Variable Neighbourhood Search (VNS) outcomes for Experiment *E1* are detailed in Table 15 across the 10 data instances. Given the randomness in the VNS algorithm's initial solution, the initialization function is executed 100 times to capture the minimum cost solution. Thus, the initial solution objective can be observed in the fourth column of the table, together with the improved objective value, the VNS computation time, and the percentage improvement from the initial solution. It also specifies the total distance traveled by all vehicles and the distribution of nodes served by common carriers versus the private fleet. Information on the IDs of private vehicles used and the percentage of unserved customers is also included in the last columns.

Table 15. Summary of VNS results for each data instance with mixed fleet utilization (Experiment *E1*).

ID	Label	Nodes	Initial solution	Objective	Time (s)	Improvement (%)	Distance (km)	Nodes with CC	Nodes with PF	PF ID	Unserved %
S1	TL	11	664.81	442.12	18.22	33.50	376.14	0	11	3, 5	0
S2	TH	46	7789.83	6270.28	20.64	19.51	3015.12	22	24	all	0
S3	FL	10	420.24	375.56	13.82	10.63	329.02	2	8	3	0
S4	FH	37	1493.52	942.87	20.27	36.87	778.42	1	36	2,3,4,5	0
S5	ML	7	451.79	385.14	18.75	14.75	364.42	0	7	3, 5	0
S6	MH	34	6679.7	1913.74	19.04	71.35	1297.59	2	32	all	0
AS7	-	60	14066.93	9362.06	18.06	33.45	4153.87	36	24	all	0
AS8	-	70	14420.43	9362.06	18.25	19.40	4868.07	26	44	all	0
AS9	-	80	15584.78	11622.75	19.37	25.26	4932.43	47	33	all	0
AS10	-	100	23037.40	20397.65	18.86	11.46	8182.05	69	31	all	0
Average	-	-	8460.94	6336.09	18.53	27.62	2829.71	20	25	-	0

The Variable Neighbourhood Search (VNS) algorithm consistently demonstrates low computation times across all instances, efficiently improving solutions with an average of 18.53 seconds even as problem complexity increases. This performance demonstrates VNS's efficacy in solving VRPs, and the improvement percentages highlight significant enhancements from initial solutions, showcasing VNS's capability to optimize the objective function effectively. Scenarios such as *S4* and *S6* show the largest improvement percentages.

In low-demand instances, VNS prioritizes the allocation of private-fleet vehicles, while high-demand instances see the inclusion of external carriers. Across the mixed fleet experiment, VNS successfully serves all customers, achieving an average improvement of 27.62% and maintaining an average computation time of 18.53 seconds.

Results comparison mixed fleet utilization

In Experiment *E1*, the Meilink Borculo B.V. transportation problem is addressed using both MILP and VNS methods, focusing on mixed fleet utilization. This setup allows private fleet vehicles to have route continuity, while common carriers directly served a customer from the depot or not at all.

Given the VRP's complexity, the MILP solution could not resolve high-demand instances to optimality within the 1800-second maximum runtime. Conversely, the VNS method identified solutions within an average computation time of 18.53 seconds for these instances. Although the VNS, as a metaheuristic, didn't match the MILP in cost performance, it achieved nearly the same objective costs for instances *S3* and *S5*, which are two out of the three instances solved to optimality by the MILP. However, for larger data instances, there's a significant cost discrepancy between VNS and MILP, particularly in artificial instances *AS7, AS8, AS9, AS10*. The reasoning behind that could be that as the size of the solution space grows, finding near-optimal solutions becomes challenging within a short computation time. Moreover, the performance of the VNS depends on the quality of the initial solution and in high-demand instances generating a good initial solution that can lead to a near-optimal final solution becomes more difficult. Notably, in all high demand instances, the VNS outperformed the MILP, likely because the MILP didn't achieve optimality. These instances underscore the VNS's capability to uncover (nearly) optimal solutions and have an average cost over the 10 data instances amounting to 6336.09 compared to the costs of the MILP equal to 7215.09.

To compare the cost outcomes of the MILP and VNS methods with the current operational costs at Meilink Borculo B.V., the costs per instance were multiplied by the number of working days in a year (=260 days). Since Meilink's current transportation is performed with the accumulated orders, the daily cost records of the company are non-comparable with the model costs.

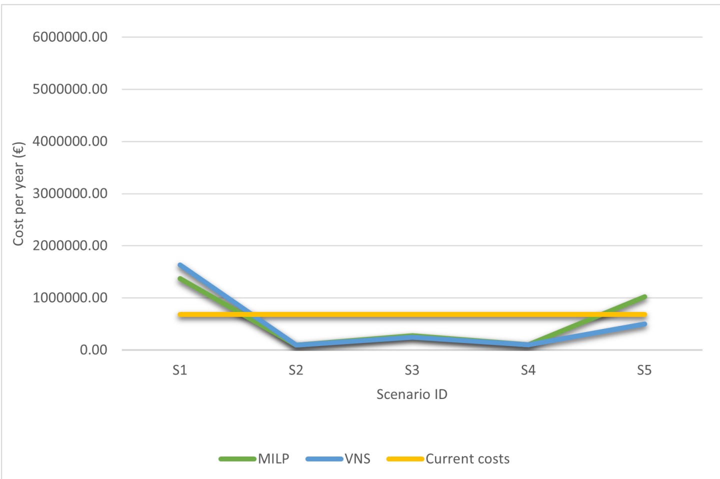

Figure 9. Comparison of yearly costs between MILP, VNS and current company costs from Experiment E1 (assuming 260 working days).

The yearly costs under each instance with the MILP and VNS are plotted on Figure 9. In this figure, the yellow line denotes the current yearly costs, the green line represents the MILP results, and the blue line indicates the VNS outcomes. Assuming Meilink Borculo B.V. faces a high demand similar to that on Fridays (data instance *S4*) throughout the whole year, both VNS and MILP project lower operational costs than the current figures, suggesting a potential cost reduction of 27 to 64% if Meilink's transportation process are optimized. However, this scenario represents an exaggerated demand model, as Meilink's actual demand fluctuates between low and high throughout the week, with seasonal variations also affecting yearly demand. As demand increases, so do the associated costs, as demonstrated in all artificial instances. In instances with very high demand, projected costs may surpass current expenses, which aligns with expectations given that these artificial instances simulate demand levels at least double those currently experienced by Meilink.

Overall, both the MILP and VNS methods present improvements over the current cost framework for Experiment *E1* at Meilink Borculo B.V. The MILP solution comes with the drawback of longer computation times. Conversely, the VNS method significantly reduces computation time, offering a more time-efficient solution approach. Moreover, the VNS outperforms the MILP for the high-demand instances by generating a better objective value and reducing the overall costs for Meilink. For larger or more complex instances, the computational time of the MILP method may become impractical, positioning the VNS as a good alternative despite it not leading to the optimal solution.

5.4.2 Exclusively private fleet

In this section, Experiment *E2* s explored, focusing solely on the use of Meilink Borculo B.V.'s private fleet to meet customer demand. The company's private fleet consists of five vehicles, with their specifications detailed in Section 2.2.3. Initially, the MILP results, followed by the VNS outcomes are provided. The results from both solution methods against the current cost situation of the company are compared.

The MILP results for Experiment *E2* are detailed in Table 16 across the 10 instances. The instance ID and label are provided in the first two columns, the number of demand nodes are provided in the third column.The 'Objective' column lists the model's costs, followed by computation time in seconds, the percentage gap to optimality, and the IDs of the private vehicles utilized. The final column reflects the percentage of unserved nodes.

As it can be observed on the table, the MILP is not able to solve high-demand instances, including *S2, S4, S6, AS7, AS8, AS9, AS10*. This challenge originates from the MILP model's requirement that all customers must be served, a condition unmet due to the limited number and capacity of the company's private fleet vehicles. Consequently, these high-demand instances are infeasible for the MILP. However, in instances with lower demand, the MILP demonstrates a good performance, solving these cases to optimality with a low computation time. In every instance, the model consistently utilizes the same private fleet vehicles, favoring those with IDs 3 and 5 for their lower fixed costs.

Table 16. Summary of MILP results for each data instance with private fleet only (Experiment *E2*).

ID	Label	Nodes	Objective	Time (s)	Gap (%)	Distance (km)	PF ID	Unserved %
S1	TL	11	365.48	0.15	0.00%	108.09	3 and 5	0
S2	TH	46	-	-	-	-	-	-
S3	FL	10	422.45	0.05	0.00%	332.80	3 and 5	0
S4	FH	37	-	-	-	-	-	-
S5	ML	7	422.01	0.14	0.00%	336.09	3 and 5	0
S6	MH	34	-	-	-	-	-	-
AS7	-	60	-	-	-	-	-	-
AS8	-	70	-	-	-	-	-	-
AS9	-	80	-	-	-	-	-	-
AS10	-	100	-	-	-	-	-	-
Average	-	-	403.32	0.11	0.00%	259.00	-	0

The VNS method's outcomes for Experiment *E2* are detailed in Table 17 for each of the 10 instances. Due to the randomness in the initial solution of the VNS algorithm and for consistency across all experiments, the initialisation function was run 100 times and the minimum cost objective is stored for applying the VNS. The table presents the initial costs, objective values, VNS computation times, and the percentage improvements over initial solutions. Additionally, it records the total distance covered by all vehicles and the number of nodes serviced by the private fleet. Information on the IDs of the private vehicles used and the percentage of unserved customers is also provided in the concluding columns.

The VNS algorithm in Experiment *E2* demonstrates notable improvements on the initial solution particularly in low-demand instances such as *S1* with a 29.5% improvement. The computation times remain low across all data instances. While the algorithm effectively optimizes objectives in low-demand instances, it faces challenges in instances with an increased number of customer nodes. This difficulty arises because the private fleet consists of only five vehicles, limiting the application of effective operators like 'Swap vehicles'. Therefore, the high-demand instances have a very low improvement percentage compared to the initial solution. Additionally, high-demand instances reveal a percentage of unserved customers, increasing with the number of customer nodes. This shows the disadvantage of relying solely on the private fleet, especially in high-demand instances.

In instances with lower demand, vehicles with IDs 3, and 5 are preferred, whereas the other internal vehicles are utilized only in high-demand instances. When comparing the objective values of the three low demand instances *S1, S3* and *S5* of VNS with a private fleet only (Expriment *E2*) to those with mixed fleet utilization (Expriment *E1*), the VNS results for both experiments are very close.

Table 17. Summary of VNS results for each data instance with private fleet only (Experiment *E2*).

ID	**Label**	**Nodes**	**Initial solution**	**Objective**	**Time (s)**	**Improvement (%)**	**Distance (km)**	**PF ID**	**Unserved %**
S1	TL	11	627.12	442.12	5.79	29.50	376.14	3, 5	0
S2	TH	46	1099.27	873.17	11.41	20.57	780.09	all	34.5
S3	FL	10	441.54	422.45	5.63	4.32	332.80	3, 5	0
S4	FH	37	945.51	944.73	11.32	0.08	784.46	all	2.7
S5	ML	7	435.01	422.02	7.85	2.99	336.09	3, 5	0
S6	MH	34	1187.44	1176.59	9.92	0.91	860.75	all	6.25
AS7	-	60	1035.88	1035.60	9.04	0.03	363.57	all	48.63
AS8	-	70	1047.30	1044.77	28.50	0.24	395.57	all	52.32
AS9	-	80	1057.08	1051.73	11.95	0.51	424.44	all	40.11
AS10	-	100	1061.17	1060.71	10.62	0.04	444.35	all	49.53
Average	-	-	893.73	847.39	11.20	5.92	509.82	-	23.40

Results comparison on exclusively private fleet

The MILP method outperformed in terms of both computation time and objective value for the instances it could solve. However, it failed to solve high-demand instances due to the private fleet's limited capacity. This limitation, requiring all customer demands to be met, is also restrictive for the VNS. However, the metaheuristic is more advantageous, solving high-demand scenarios despite the unserved customers, due to the way the metaheuristic is formulated. It stores a list of unserved customers and iterates until a solution of those is found or the stopping criterion of maximum iterations is reached. Since the internal vehicles are a low number, the VNS serves only the customers it could given the constraints of the model, and offers routes for those that could be served with the available trucks. Therefore, in instances with low number of customer nodes, the MILP is more advantageous in terms of both computation time and finding the optimal routing. However, the VNS can also be used when it is unknown if all the constraints can be met, such as instances with higher number of nodes, where it can still provide a partial solution to the VRP.

Moreover, a comparison between the costs of the VNS and MILP in Experiments *E1* and *E2* reveals insightful findings. Based on the MILP results, which optimally solved data instances *S1, S3* and *S5*, the mixed fleet utilization experiment *E1* yielded better outcomes. This experiment allowed for the use of both private vehicles and external trucks, with only the internal vehicles being routed. For the VNS, data instances *S3* and *S5* showed better costs results under *E1*, whereas instance *S1* generated the same results in both experiments.

It is important to note that Meilink Borculo B.V. does not face only low-demand situations throughout the year. The necessity for external trucks indicates that the conditions under Experiment *E1*, which permits a mixed fleet approach, are more favorable for addressing the company's transportation needs. To understand why this is caused, the unserved nodes in high-demand instances by the VNS are plotted in Figure 10. As it can be seen, those customers are mostly located at the furthest points from the depot. This suggests that to serve clients in distant locations such as Apeldoorn, Dedemsvaart, Oude Meer, 'S-Hertogenbosch, Maassluis, and Dordrecht, the employment of external trucks is essential. This experiment underscores the inefficiency of the private fleet's capacity to meet all of Meilink's daily demand nodes, particularly on days with high demand levels.

Figure 10. Locations with unmet demand nodes by the VNS under Experiment E2.

5.4.3 Common carriers only

Within this section, Experiment *E3* is performed, where only the common carrier vehicles can be used to respond to customers' demand. This setup requires adjustments to the MILP model to exclusively incorporate external vehicles for serving demand nodes. Unlike private fleet vehicles, common carriers operate without route continuity, meaning each vehicle serves a customer directly from the depot. The VNS approach adheres to the same conditions, with an unlimited number of common carriers available for deployment.

The MILP results from Experiment *E3* are provided in Table 18 for each of the 10 instances. The instance ID and label are provided in the first two column, the number of demand nodes are provided in the third column, followed by the objective value, computation time, and distance. The last column shows the percentage of unserved nodes.

Table 18. Summary of MILP results for each data instance with common carriers only (Experiment *E3*).

ID	Label	Nodes	Objective	Time (s)	Gap (%)	Distance (km)	Unserved %
S1	TL	11	2098.07	0.05	0	755.79	0
S2	TH	46	22700.32	2.92	0	8177.34	0
S3	FL	10	1026.33	0.02	0	369.71	0
S4	FH	37	9295.52	0.54	0	3348.54	0
S5	ML	7	651.70	0.05	0	234.64	0
S6	MH	34	13420.48	1.04	0	4834.47	0
AS7	-	60	21061.84	9.21	0	7587.13	0
AS8	-	70	23571.68	12.69	0	8491.24	0
AS9	-	80	24026.88	6.41	0	8655.20	0
AS10	-	100	32858.08	10.33	0	11836.48	0
Average	-	-	15071.09	4.33	0	5429.05	0

The MILP solves relatively quickly the instances under the third experiment. The reason is that there are no constraints involved with sending the external trucks to the demand nodes. This efficiency originates from the absence of constraints related to the routing of external trucks to demand nodes. The model disregards the capacity and customer time window constraints for common carriers, assuming direct service from the depot to each node will inherently satisfy these conditions. Consequently, while computation times are minimal, the model's objective costs significantly escalate compared to *E1*. Despite these higher costs, the experiment ensures no customer is left unserved, and there is no gap from optimality.

For the VNS, solving the 10 instances under the common carriers only experiment, results are provided in Table 19. The objective value, computation time, and distance travelled can be observed on the table.

In this experiment, where each customer is served directly from the depot without considering route continuity or varying costs among common carriers, the solution space is very limited. This impacts the VNS's ability to find improvements after the initial solution, as the initial solution is the only possible configuration under the given constraints. Since there is only one feasible solution, the algorithm could not explore any neighbourhoods. The computation time of 6-7 seconds per instance is quite high, given that the solution cannot be improved. The objective values from the VNS are very high across all instances, which reflects the high distance travelled. This is again a consequence from the experiment constraint that necessitate serving each customer directly from the depot.

Table 19. Summary of VNS results for each data instance with common carriers only (Experiment *E3*).

ID	Label	Nodes	Objective	Time (s)	Improvement (%)	Distance (km)	Unserved %
S1	TL	11	2098.07	7	0	755.79	0
S2.1	TH	46	22700.32	7.46	0	8177.34	0
S3.1	FL	10	1026.33	7.70	0	369.71	0
S4.1	FH	37	9295.52	6.59	0	3348.54	0
S5.1	ML	7	651.70	9.18	0	234.64	0
S6.1	MH	34	13420.48	7.08	0	4834.47	0
AS7.1	-	60	21061.84	6.70	0	7587.13	0
AS8.1	-	70	23571.68	6.23	0	8491.24	0
AS9.3	-	80	24026.88	6.52	0	8655.20	0
AS10.1	-	100	32858.08	5.94	0	11836.48	0
Average	-	-	15071.09	7.04	0	5429.05	0

Results comparison on common carriers only

In Experiment *E3* only common carriers are utilized to fulfill all demand nodes, with external trucks travelling directly from the depot to each customer, without adhering to a route continuity constraint. Given the high per-kilometer cost associated with these trucks, the overall objective values, as determined by both the VNS and MILP methods, are significantly high across all data instances. This experiment's constraints lead to a single feasible solution, resulting in identical cost outcomes for both the VNS and MILP models. However, there is a notable difference in computation times between the two methods. The MILP demonstrates overall a lower computation time compared to the VNS. This discrepancy arises because the MILP method does not engage in optimizing or exploring neighboring solutions, thus quickly identifying the only feasible solution. Furthermore, when comparing average computation times, the MILP method significantly outperforms the VNS, averaging 4.33 seconds.

In comparing the outcomes of Experiment *E3* with those of *E1* and the current costs of the company, it becomes evident that the costs associated with using only common carriers (as seen in Experiment *E3*) exceed both the current operational expenses and the results obtained from Experiment *E1*. Therefore, it is preferable for the company to use its own vehicles, in combination with the non-routed common carriers.

Experiment *E3* serves as an illustration of the limitations imposed by specific operational constraints, such as the requirement for common carriers to serve customers directly from the depot without route optimization. The findings from this experiment suggest that in instances with a predetermined or highly constrained solution space, simpler or more straightforward solution methods, like the MILP model, may offer greater efficiency than complex metaheuristic approaches in terms of computation time. This insight is particularly relevant when the optimization problem is characterized by a single feasible solution.

5.4.4 Routing common carriers

In this section, Experiment *E4* is conducted, which introduces the capability for common carriers to be routed, enabling them to serve multiple customers sequentially after departing from the depot. This adjustment necessitates modifications to both the MILP and VNS solution methods to ensure they accommodate constraints related to customer time windows (TWs), vehicle capacity, and route continuity for both internal and external vehicles. The experiment is divided into two parts:

Firstly, both private fleet and common carriers are utilized, with routing applied to all vehicles. The outcomes of this approach are comparable with those from Experiment *E1*, where a mixed fleet is similarly employed but without routing for common carriers.

Secondly, the focus shifts to exclusively utilizing and routing common carrier vehicles. The performance of the MILP and VNS under these conditions is comparable against the results from Experiment *E3*, which also exclusively uses external vehicles but does not apply route continuity.

5.4.4.1 Using both private fleet and common carriers

The VNS and MILP models are allowed to use mixed fleet, however, all vehicles in the fleet are routed. This is achieved through having a merged list of vehicles (both internal and external) for the MILP model and removing all constraints and variables related to the common carriers only.

The MILP results from Experiment *E4* using mixed fleet vehicles are provided in Table 20 for each of the 10 real-world data instances. The table outlines each instance by ID and label, the number of demand nodes, objective value, computation time, optimality gap, and the total distance traveled.

For instances with lower demand (*S1, S3, S4*, and *S5*), as well as the high-demand Friday (*S5*), the model achieves optimality within a reasonable time. Notably, in these instances, the model exclusively utilizes the private fleet, avoiding the use of common carriers. This decision underscores the cost implications of routing common carriers, especially in low-demand instances where the return trip to the depot incurs additional expenses. The customer nodes to which common carriers are scheduled can be observed in Appendix B.2.

In contrast, instances with a higher number of demand nodes (*S2, S6, AS7, AS8, AS9*, and *AS10*) rely on the outsourcing of common carriers. Despite achieving a solution in these more complex data instances, the model exhibits a significant optimality gap, averaging 8.31% across all instances. This gap highlights the challenges in finding optimal solutions within the constraints of higher demand and the specified routing requirements.The experiment ensures all customers are served, with an average computation time of 1214.68 seconds.

Table 20. Summary of MILP results using mixed fleet for each scenario with routing common carriers (Experiment *E4*).

ID	Label	Nodes	Objective MILP	Time (s)	Gap (%)	Distance (km)	Nodes with CC	Nodes with PF	Number of CC	PF ID	Unserved %
S1	TL	11	365.48	3.09	0.00	108.09	0	11	0	3, 5	0
S2	TH	46	4107.74	1800.00	16.72	2451.46	32	14	10	1,2, 3	0
S3	FL	10	422.45	2.43	0.00	332.80	0	10	0	3, 5	0
S4	FH	37	983.63	1340.26	0.00	319.59	0	37	0	all	0
S5	ML	7	422.01	1.05	0.00	336.09	0	7	0	3, 5	0
S6	MH	34	4446.93	1800.00	23.64	2330.35	18	16	13	1,3,4,5	0
AS7	-	60	4330.18	1800.00	5.07	2094.15	36	24	9	1,2,3,4	0
AS8	-	70	4516.86	1800.00	17.64	1946.78	47	23	11	all	0
AS9	-	80	6150.4	1800.00	18.51	2779.39	54	26	16	all	0
AS10	-	100	6788.08	1800.00	21.56	3474.51	65	35	12	all	0
Average	-	-	3253.38	1214.68	8.31	1617.32	25	20	7	-	0

For the VNS method, addressing the 10 real-world data instances under the mixed fleet experiment with routing all vehicles, the results are summarized in Table 21. This table details the objective values, computation times, distances traveled, and the utilization of vehicle types across instances, along with the distribution of nodes served by each vehicle type.

In this experiment, the initial solution for the VNS is executed 100 times for each instance to find the lowest cost initial solution, which is later optimized through the VNS by taking the highest cost improvement. The results showcase significant improvements across all instances, with optimization percentages ranging from 24% to 56%. This range of improvement underscores the VNS's capability to enhance initial solutions effectively across diverse instances. Notably, the computation times vary, with the more complex high-demand and artificial instances requiring the most time. Despite this variation, the average computation time stands at 20.59 seconds, which is considerably lower than the MILP model's average. Additionally, the VNS method successfully served all customers in every instance, further demonstrating its efficiency and effectiveness in optimizing Meilink Borculo B.V.'s transportation planning under mixed fleet utilization with routing capabilities.

Table 21. Summary of VNS results for each instance with mixed fleet utilisation and routing all vehicles (Experiment *E4*).

ID	**Label**	**Nodes**	**Initial solution**	**Objective VNS**	**Time (s)**	**Improvement (%)**	**Distance (km)**	**Nodes with CC**	**Nodes with PF**	**Number of CC**	**PF ID**	**Unserved %**
S1	TL	11	1081.07	472.56	19.16	56.29	294.77	8	3	2	5	0
S2	TH	46	4745.03	2106.21	23.78	55.61	1021.74	29	17	6	all	0
S3	FL	10	496.19	375.46	15.29	24.33	147.4	1	9	1	5	0
S4	FH	37	1601.05	782.107	19.76	51.15	430.56	9	28	1	3, 4, 5	0
S5	ML	7	533.87	321.82	26.77	39.72	205.52	5	2	2	3	0
S6	MH	34	2596.59	1327.15	20.08	48.89	744.1	18	16	3	2, 3, 4, 5	0
AS7	-	60	6654.4	2986.6	21.08	55.12	1417.73	49	11	11	all	0
AS8	-	70	5989.24	3099.24	19.58	48.25	1485.21	50	20	10	all	0
AS9	-	80	6848.46	3527.1	20.48	48.5	1632.13	64	16	10	all	0
AS10	-	100	9947.14	5357.98	19.95	46.14	2348.51	88	12	16	all	0
Average	-	-	4049.3	2035.62	20.59	47.4	972.77	32	13	6	-	0

In the results table, it is observed that the VNS consistently employs at least one common carrier across all instances, demonstrating the critical role of external carriers in fulfilling the demand. Additionally, the VNS frequently selects vehicles with IDs 3, 4, and 5 for almost every instance, indicating a preference for specific vehicles within the private fleet based on their operational efficiency or cost-effectiveness. On average across all instances, there are 32 nodes served by the common carriers, compared to 13 nodes from the private vehicles. This shows the importance of using the common carriers, since otherwise, the demand could not be satisfied solely by the internal vehicles.

When comparing the Mixed Integer Linear Programming (MILP) objective and computation time with the Variable Neighbourhood Search (VNS) under the mixed fleet utilization case with routing all vehicles, it becomes evident that the VNS outperforms the MILP in both cost efficiency and computation speed. This difference arises because the MILP struggles to find optimal solutions for high-demand instances, whereas the VNS demonstrates robustness in generating low cost objectives. The VNS efficiently optimizes the initial solution, with an average computation time of just 20.59 seconds across the 10 data instances, showcasing its speed and effectiveness.

Conversely, the MILP, while achieving optimal solutions for low-demand instances, requires significantly more computational resources, averaging 1214.68 seconds in computation time. This is notably higher compared to the VNS. When comparing the average costs between the MILP and VNS, the latter shows a highly optimized performance with costs of 2035.62 compared to the costs of MILP equal to 3235.28. Given this context, the extended computation time required by the MILP does not justify over the VNS solutions, especially when considering the VNS's efficiency and the quality of solutions it provides under this experiment.

5.4.4.2 Using common carriers only

In this section, the MILP and VNS models are allowed to use common carriers only and the vehicles are routed.

The Mixed Integer Linear Programming (MILP) results from Experiment *E4*, focusing on the routing of common carriers only, are detailed in Table 22. This table outlines the objective value for each instance, alongside computation time, optimality gap, and total distance travelled.

In this experiment, the MILP successfully solved only the instances with lower demand, specifically *S1*, *S3*, and *S5*. A large contrast in computation times is evident between these low-demand instances and the high-demand ones, including the artificial instances designed to test the system under very higher demand conditions. For instances with increased demand, the optimality gap approaches 20%, with an average gap of 6.04% across all instances. This indicates a significant challenge in achieving optimal solutions within the defined computational limits for these more demanding cases.

Computation times are notably high, averaging 1260.40 seconds across all instances. This extended computation time underscores the complexity and computational demand of solving larger-scale problems with the MILP approach. The number of common carriers utilized within the MILP model varies significantly depending on the instance, ranging from 2 in instances with lower demand to as many as 18 in the highest demand instances.

Table 22. Summary of MILP results for each data instance with routing the common carriers (Experiment *E4*).

ID	**Label**	**Nodes**	**Objective**	**Time s**	**Gap (%)**	**Distance (km)**	**Number (of CC)**	**Unserved (%)**
S1	TL	11	300.23	1.41	0.00%	108.09498	2	0
S2	TH	46	3405.09	1800.00	1.87%	1226.62	10	0
S3	FL	10	675.89	1.37	0.00%	243.35	2	0
S4	FH	37	905.4	1800.00	6.93%	325.98	6	0
S5	ML	7	933.49	1.23	0.00%	336.09	2	0
S6	MH	34	3479.06	1800.00	4.08%	1205.84	8	0
AS7	-	60	4479.37	1800.00	4.58%	1612.12	14	0
AS8	-	70	4464.88	1800.00	5.54%	1606.90	14	0
AS9	-	80	5486.02	1800.00	17.53%	1974.41	16	0
AS10	-	100	7099.88	1800.00	19.87%	2555.24	18	0
Average	-	-	3122.93	1260.40	6.04%	1119.46	9	0

The Variable Neighbourhood Search (VNS) results for Experiment *E4*, focusing solely on the use of common carriers, are detailed in Table 23. This table presents the initial solution costs, objective value for each instance, computation time, distance traveled, and the number of vehicles used. For this experiment, the initial solution is executed 100 times for each instance, storing the minimum costs.

The VNS demonstrates its capability to serve all customers effectively, achieving significant improvements from the initial solutions. Across all instances, the average improvement achieved by the VNS is 42%. The computation times for the VNS are consistently lower than those observed for the MILP model, with an average computation time of 8.64 seconds across all tested instances. This efficiency highlights the VNS's ability to quickly refine solutions, making it a viable option for performing real-time optimization.

The distance travelled by all vehicles in the VNS instances is generally lower compared to the MILP model, with an average distance of 1043.64 km. This reduction in travel distance further underscores the VNS's optimization capabilities, potentially leading to lower transportation costs and increased operational efficiency.

The number of common carriers deployed per instance varied, ranging between 2 and 17 vehicles, similar to the range observed in the MILP model. On average, 9 vehicles are utilized across the instances, indicating a consistent approach to meeting demand with the available common carrier vehicles.

Table 23. Summary of VNS results for each instance with common carriers only and routing of vehicles (Experiment *E4*).

ID	**Label**	**Nodes**	**Objective VNS**	**Time (s)**	**Improvement (%)**	**Distance (km)**	**Number of CC**	**Unserved (%)**
S1	TL	11	400.11	7.86	73.24	144.12	2	0
S2	TH	46	3724.43	9.73	41.19	1340.95	10	0
S3	FL	10	675.89	6.57	37.72	243.35	2	0
S4	FH	37	1677.46	7.99	34.89	603.95	4	0
S5	ML	7	947.22	11.37	43.51	341.22	3	0
S6	MH	34	2375.68	8.14	36.62	855.34	7	0
AS7	-	60	4695.46	8.88	40.57	1690.55	17	0
AS8	-	70	4113.1	8.79	41.74	1480.88	15	0
AS9	-	80	4534.26	8.79	39.62	1632.51	16	0
AS10	-	100	5842.36	8.23	40.12	2103.48	17	0
Average	-	-	2898.6	8.64	42.92	1043.64	9	0

Experiment *E4* delves into the potential benefits of routing common carriers on the model's solution, contrasting with the results from Experiment *E3*, where common carriers are not routed. Both the MILP and VNS costs for Experiment *E4* are significantly lower than those observed in Experiment *E3*, underscoring the significant impact of routing external vehicles on improving the transportation problem's objective value. Furthermore, should Meilink face high demand consistently throughout the year, the exclusive use of common carriers—provided they are routed—could still lead to a reduction in the company's transportation costs. Thus, the findings from both solution methods suggest that the VRP can be optimized, and costs can be minimized by employing either method, with the VNS showcasing greater efficiency in terms of computation time and the ability to find near-optimal solutions.

5.4.5 Comparison across the experiments

Throughout the analysis, four distinct experiments (*E1 - E4*) are conducted to evaluate different constraints within the transportation problem faced by Meilink Borculo B.V. These experiments are addressed using two solution methods: Variable Neighbourhood Search (VNS) and Mixed Integer Linear Programming (MILP). The experiments explore various operational strategies, including mixed fleet utilization (*E1*), exclusive use of private fleet vehicles (*E2*), and instances where all vehicles, including common carriers, are routed (*E4*). This comprehensive approach provides insights into the optimization potential under varying constraints and operational strategies.

The mixed fleet utilization experiment (*E1*) assesses the effectiveness of employing both private fleet and common carriers. The results, summarized in Figure 11, indicate that while the MILP is quicker and produces optimal solutions for instances with low demand and lower number of nodes, it cannot cope with the complexity of high-demand instances. The MILP is not able to find good enough solutions for those instances, generating high costs, and high gaps to optimality with an average of 16.28%. VNS is faster in optimizing solutions for this experiment with an average computation time of 18.53 seconds compared to the MILP, which reached 1260 seconds on average. Moreover, the metaheuristic managed to produce on average lower costs over the 10 real-world instances, outperforming the MILP model solutions.

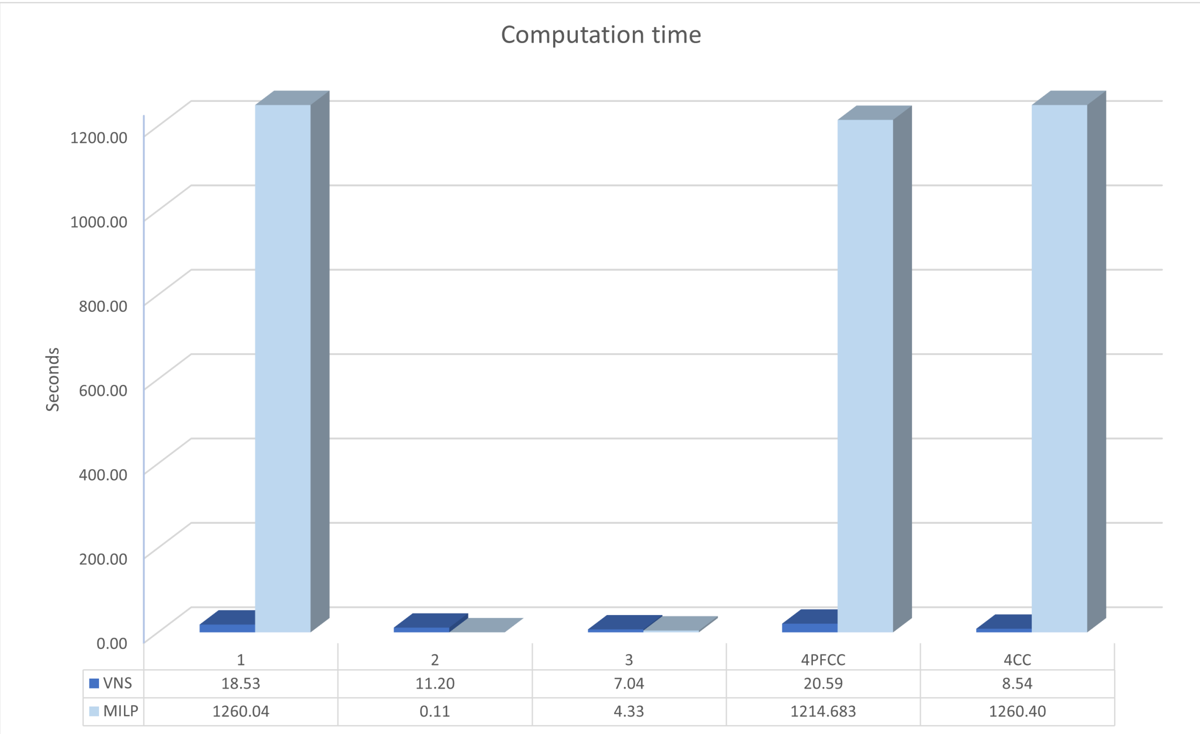

Figure 11. Overview of the average computation time performance of the VNS and MILP solution approaches across Experiments *E1-E4*. Experiment *E4* is divided into 4PFCC, corresponding to the private fleet and common carriers routing situation and 4CC - the instances where only common carriers are routed.

Experiment *E2*, focusing solely on the use of private fleet vehicles, reveals that only low-demand instances could be fully serviced by the existing fleet, according to both VNS and MILP models. This finding suggests that if Meilink Borculo B.V. aims to rely exclusively on its private fleet to accommodate future growth, acquiring additional vehicles would be necessary to meet high-demand instances. Therefore, both the computation times and the overall objective values in Experiment *E2* for both solution methods is very low, compared to the other experiments (Figure 12).

In Experiment *E3*, using only common carriers without routing the vehicles, the results for both models MILP and VNS are consistent. Both solution methods have only one possible solution, due to the straightforward allocation of trucks from the depot to each customer node. Therefore, the VNS could not improve the initial solution within this experiment. Moreover, the experiment showed the highest average costs over all experiments for both solution methods (Figure 12), suggesting that using external carriers without routing them could potentially increase the overall transportation costs of the company.

Experiment *E4* introduces routing for all vehicles, including common carriers, and demonstrates a significant cost reduction across both solution methods when vehicles are routed. Experiment *E4* once again shows the good results in terms of computation time and objective value generated by the VNS algorithm. The VNS method consistently generates lower average costs for both configurations with and without private fleet compared to the results obtained from the MILP (Figure 12). The MILP, on the other hand, is incapable of solving the high-demand instances to optimality, within reasonable computation time. Notably, Experiment *E4* reveals that utilizing only common carriers, especially in high-demand instances, could lead to lower transportation costs, despite their higher per-kilometer charges compared to the private fleet. This outcome suggests that the fixed costs associated with private fleet vehicles contribute substantially to overall transportation expenses.

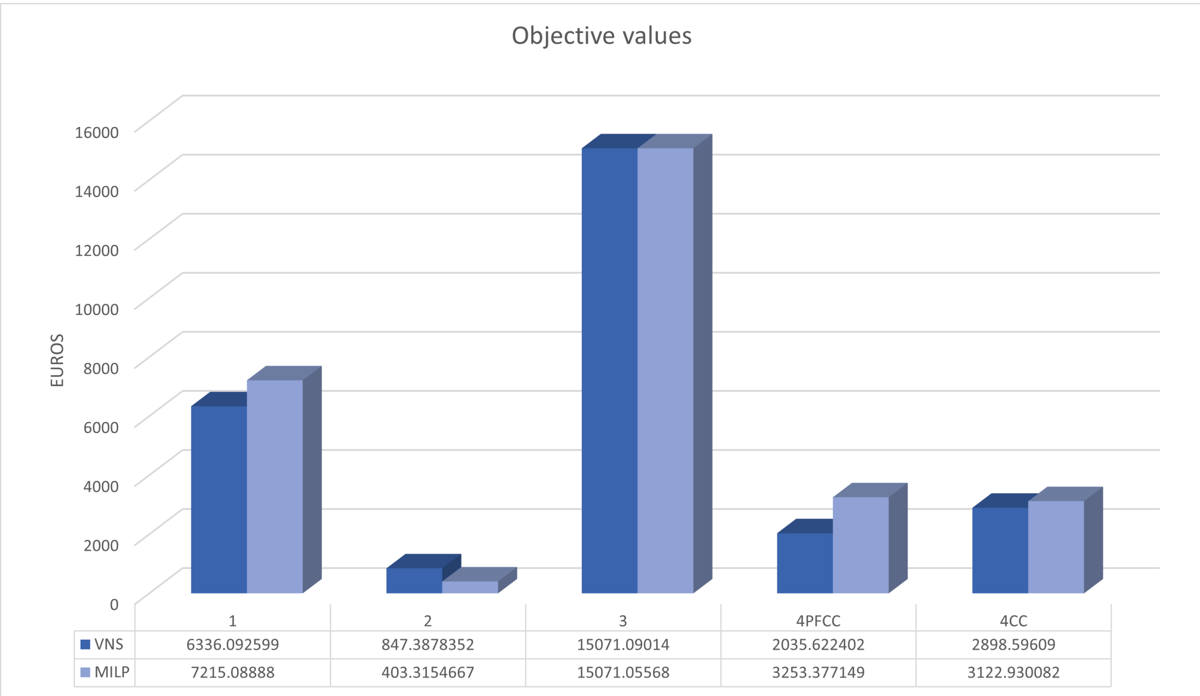

Figure 12. Overview of the average objective value (minimum costs) performance of the VNS and MILP solution approaches across Experiments *E1*-*E4*. Experiment *E4* is divided into 4PFCC, corresponding to the private fleet and common carriers routing situation and 4CC - the scenario where only common carriers are routed.

Comparing the experimental results to Meilink's current transportation costs, both Experiments *E1* and *E4* showed potential for significant yearly cost savings under both the MILP and VNS approaches. However, given the MILP's longer computation times and challenges in achieving near-optimal solutions within a reasonable timeframe, the VNS emerges as a more efficient solution method, generating better objective values within reasonable computation time. The VNS's ability to quickly find optimal or near-optimal solutions, even in complex data instances involving up to 100 customer nodes, underscores its suitability for operational implementation. Therefore, it is recommended that Meilink Borculo B.V. consider adopting the VNS metaheuristic for optimizing its transportation processes, leveraging its efficiency and effectiveness in reducing costs and improving operational performance.

5.5 Conclusion

This chapter delves into the fine-tuning of parameters and the evaluation of various data instances across five distinct experiments. More specifically, the research question *How does the developed solution for optimizing Meilink's transportation process perform compared to the current situation?* is addressed.

The preparation phase involves creating data sets for parameter tuning, including ten artificial instances with a range of demand nodes—some with fewer, some equal to, and some exceeding those in the real-world instances. Additionally, ten real-world data instances are developed to simulate different challenges a company might encounter in their transportation logistics. A detailed analysis of Meilink Borculo B.V.'s transportation costs ensures consistency throughout the evaluation of the experiments.

The initial experiment (*E0*) focuses on fine-tuning the Variable Neighborhood Search (VNS) algorithm's parameters and determining the maximum computation time for the Mixed Integer Linear Programming (MILP) model. The Randomized Vehicle Selection method is found to yield better outcomes, with the optimal number of iterations without improvement set at 100, given the VNS algorithm's speed. Moreover, the sequence of local search operators is established, and the experiment revealed that the five operators should be in the order Swap vehicles ! 2-opt ! Swap customers ! Move ! Reinsertion. The adaptability of the VNS is highlighted, with its adaptive shaking mechanism selected for its ability to enhance results. For the MILP, a balance between computation time and accuracy necessitated setting the model's maximum runtime to 1800 seconds.

Subsequent evaluations of Experiment *E1-E4* with the adjusted parameters for both MILP and VNS on real-world instances shows that VNS consistently outperforms MILP in scenarios with high demand, in terms of both computation time and objective value. However, MILP achieves better cost outcomes in low-demand situations. The VNS solution significantly improves Meilink's transportation process efficiency, offering a more systematic approach to planning and execution, which leads to better resource utilization and reduced operational costs. Compared to the existing transportation costs at Meilink, the VNS solution shows a 64% improvement.

These experiments provide a comprehensive view of the cost efficiency of different vehicle types for the company. It becomes clear that routing all vehicles is crucial for minimizing total transportation costs. The analysis reveals minimal cost differences between using a routed mixed fleet and relying solely on routed common carriers. One scenario, involving external vehicles without routing, results in significantly higher costs. Sole reliance on the private fleet fails to meet high demand instances, indicating that Meilink cannot depend only on its internal fleet to fulfill consumer demand.

As the company grows, using MILP for vehicle routing may become impractical due to long computation times of the model, often exceeding 30 minutes. Therefore, the VNS stands out as the more viable solution, given its good performance in handling large data instances.

6 Conclusions, recommendations, and future research

In the concluding section of this master thesis, insights and recommendations are offered based on the findings from addressing the transportation problem faced by Meilink Borculo B.V. This chapter not only summarizes the outcomes but also advises the company on vehicle selection and strategies for optimizing transportation costs effectively. Additionally, the chapter highlights the theoretical and practical contributions of this research, acknowledges its limitations, and suggests directions for future studies.

6.1 Conclusions

This study has thoroughly investigated Meilink Borculo B.V.'s transportation process, identifying its reliance on manual planning and the absence of a dedicated route optimization system. Despite the complexity of managing a fleet that includes lorries, trailers, and a box truck, alongside occasional use of external carriers, the company's approach is heavily dependent on employee expertise without leveraging advanced routing algorithms. The analysis revealed that transportation costs are a significant expenditure for Meilink, with efforts to minimize these costs being crucial due to their impact on the company's profitability.

This study embarked on addressing the high transportation costs encountered by Meilink Borculo B.V., aiming to identify a solution method that effectively reduces the company's overall transportation expenses. An extensive review of existing literature laid the groundwork for developing a solution. Subsequently, a Mixed Integer Linear Program (MILP) for a Multi-Trip Capacitated Vehicle Routing Problem with Divisible Delivery and Pickup Time Windows and Private Fleet and Common Carriers (MTCVRPDDPTWPFCC) and a Variable Neighbourhood Search (VNS) were developed. These solution methods were tailored to fit the specific characteristics of Meilink's transportation operations, including aspects such as pickup and delivery demands, time windows for customers and the depot, vehicle capacities, types of vehicles, item volumes, and both fixed and variable costs. Data and necessary assumptions were provided by Meilink Borculo B.V. to support this process.

To fine-tune the VNS and MILP for broader applicability and ensure their effectiveness in scenarios similar to those of Meilink, several data instances were created. This tuning process aimed to ensure that both solution methods could efficiently handle demand nodes of varying sizes, reflective of Meilink's actual situation. The optimal settings identified were then applied to the real-world data instances of the company. Extensive experimentation demonstrated the effectiveness of the developed VNS solution in improving Meilink's transportation process efficiency, significantly reducing transportation costs by 64% compared to the current situation. The experiments also highlighted the importance of routing all vehicles to minimize costs and the challenges of relying solely on a private fleet to meet high demand instances. The VNS emerged as a preferable solution for its adaptability and efficiency, especially as the company continues to grow and faces the impracticality of using MILP due to excessive computation times.

6.2 Contribution to theory

The literature review revealed an extensive research on VRP variations, yet existing models do not encompass all aspects of the problem tackled in this study. A study conducted by Zhang et al. (2023) incorporates five out od the six VRP characteristics this thesis dealt with, however, it does not look into differentiating between vehicles as private fleet and common carriers. This showed the need to develop a novel mathematical formulation by merging mathematical formulations from Bolduc et al. (2008) and Wassan et al. (2017) and adding additional constraints, including time-window constraints. This MILP formulation represents a MTCVRPDDPTWPFCC, a model of this specificity not previously found in literature.

The VRP involving a private fleet and common carriers is relatively new in transportation and logistics research, for which not many solution methods have been proposed. In the existing literature, mainly the RB-ACS heuristics has been proposed as a solution method and its performance has been showed for the VRPPFCC. This study undertook a different solution approach by employing a Variable Neighbourhood Search Algorithm and showing its application to routing problems with private fleet and common carriers. The detailed analysis of VNS, including parameter tuning and the strategic combination of local search operators, provides valuable insights into the algorithm's adaptability and efficiency in navigating large and complex solution spaces.

By comparing the performance of a Mixed Integer Linear Programming (MILP) model with the VNS metaheuristic, this thesis contributes to the ongoing discourse on the practical applicability of exact versus heuristic methods in solving VRPs. The findings highlight the limitations of MILP in terms of computation time for large instances and underscore the advantages of metaheuristics in achieving near-optimal solutions within reasonable computation time.

Lastly, the successful application of VNS in this context enriches the metaheuristic theory by providing a case study on its adaptability and effectiveness in a complex, real-world problem. It adds to the evidence supporting the use of VNS in logistics and transportation management, particularly for problems that are not adequately addressed by traditional optimization methods.

6.3 Contribution to practice

Conducted at Meilink Borculo B.V., a company with a rich history in packaging solutions, this research identified opportunities for enhancing the company's transportation strategies. The development and implementation of a tailored Variable Neighborhood Search (VNS) algorithm for Meilink Borculo B.V. presents a practical solution for optimizing transportation processes. This strategy enables companies to systematically plan and execute transportation tasks, leading to improved resource utilization and reduced operational costs. The demonstrated success of the VNS algorithm in reducing transportation costs by 64% offers a compelling case for its adoption in similar logistics operations.

Moreover, it illustrated how operational-level VRP solutions could inform tactical-level decisions, offering guidance on the most cost-efficient and effective vehicle types (private or external) for the company. The comparative analysis of different vehicle routing scenarios, including the use of private fleets versus common carriers and the impact of routing all vehicles, offers critical insights for decision-making in fleet management. Companies can leverage these findings to make informed decisions about fleet composition, vehicle utilization, and the use of external transportation services.

The identification of Variable Neighborhood Search (VNS) as a scalable solution for vehicle routing problems addresses a critical need for logistics companies experiencing growth and increasing complexity in their operations. The adaptability of VNS to different problem sizes and its efficiency in handling extensive problem instances make it a suitable choice for companies at various stages of growth.

Finally, this research offers guidance for logistics and transportation companies on implementing advanced optimization solutions. From problem formulation and algorithm selection to parameter tuning and performance evaluation, the methodologies outlined in this thesis provide a roadmap for companies seeking to enhance their logistics operations through optimization.

6.4 Recommendations

Based on the outcomes of the MILP and VNS analyses, it is advised that Meilink Borculo B.V. should utilize either mixed fleet or solely common carriers, ensuring their routing to efficiently serve clients daily. If the company opts to retain its private fleet, it is still recommended to use these vehicles alongside common carriers. Specifically, common carriers should be deployed to transport to customers located farthest from the depot. The current private fleet alone cannot meet the fluctuating demand. Should the company decide solely to rely on its private fleet, an expansion of the fleet is necessary, which could significantly increase transportation costs due to the high expense of acquiring new trucks. Vehicles with IDs 3, 4, and 5 from the private fleet are recommended for frequent use, as they incur the lowest fixed costs and are consistently selected in both MILP and VNS solutions.

Furthermore, the development of a routing tool is recommended for daily vehicle assignment to client locations. Regardless of the vehicle types used, routing is crucial for reducing both travel distances and transportation costs. The greatest cost reduction is achievable through servicing all demand nodes with external vehicles, together with a routing tool. This tool could integrate either the MILP or VNS models, considering computation time and the number of demand nodes. In data instances with lower number of demand nodes, less than 30, the MILP model can find optimal solution within the computation limit of 30 minutes. For the high-demand scenarios, on the other hand, it is suggested using the VNS algorithm, which offers near-optimal solutions, significantly lowering the company's transportation expenses.

6.5 Limitations and future research

Several assumptions were integral to the mathematical model's formulation, including the volume of transported items (which are often folded for delivery), vehicle speeds, and loading/unloading times at client locations. Additionally, despite the company experiencing variable demand throughout the year, this study focused only on days with the highest and lowest demands, avoiding extreme outliers. For a more accurate yearly cost calculation, incorporating average demand across different weekdays could enhance the solution.

The pickup requests in this study are treated as deterministic, however, this approach overlooks their potential dynamic nature, where requests might emerge after vehicles have departed the depot. The absence of data on such occurrences limits the ability to assess their frequency and impact. Thus, systematic tracking of dynamic pickup requests by the company is recommended.

During parameter tuning, the initial configurations of the MILP and VNS were adjusted. Modifications required by the experiments were not extensively tuned, suggesting that further tuning could optimize results. Additionally, the optimization of only the best cost solutions in the VNS lacked statistical analysis, leaving room for questioning whether the entire solution space was explored. Incorporating statistical analysis could provide deeper insights into the efficacy of the solutions identified.

Future studies could extend this work by comparing multiple metaheuristics or variations of the VNS to the current findings, potentially enhancing solution quality. Such research could also delve into the operational, tactical, and strategic levels of transportation planning at Meilink, evaluating the suitability of truck models and determining the most efficient vehicle types for serving specific clients. Beyond cost considerations, strategic-level research could assess the service level requirements, customer satisfaction and flexibility of using either type of transportation, which could offer a holistic view on the transportation efficiency and effectiveness.

References

Alridha, A., Salman, A. M., and Al-Jilawi, D. A. (2021). The applications of np-hardness optimizations problem. In *Journal Physics: Conf. Ser. 1818 012179*. IOP Publishing.

Archetti, C., Speranza, M., and Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science 40(1)*, pages 64–73. doi:10.1287/trsc.1040.0103.

Bartalero, G., Rosa, A., and Chiara, B. (2020). Analysis of outsourcing conditions for freight transport and logistics among manufacturing companies: insights from a review of data and a field investigation. In *Transportation Research Procedia 45*, pages 45–466. Elsevier B.V.

Bell, J. E. and McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, pages 41–48. doi:10.1016/j.aei.2004.07.001.

Blocho, M. (2020). Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems. In *Smart Delivery Systems*, pages 101–156. Elsevier. doi:10.1016/B978-0-12-815715-2.00009-9.

Bolduc, M.-C., Renaud, J., Boctor, F., and Laporte, G. (2008). A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of Operational Research Society*, pages 776–787. doi:10.1057/palgrave.jors.2602390.

Brandão, J. and Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational Research Society*, pages 799–805. doi:10.1057/palgrave.jors.2600667.

Cattaruzza, D., Absi, N., and Feillet, D. (2016). Vehicle routing problems with multiple trips. *Operations Research*, pages 223–259. doi:10.1007/s10288-016-0306-2.

Chen, J., Chiang, C., Chen, B., and Chen, C. (2007). Solving vehicle routing problem with time window constraints. In *2007 International Conference on Service Systems and Service Management*, Chengdu, China. IEEE. doi:10.1109/ICSSSM.2007.4280252.

Christofides, N., Mingozzi, A., and Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, pages 255–282.

Chu, C.-W. (2005). A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research*, 165:657–667. doi:10.1016/j.ejor.2003.08.067.

Clarke, G. and Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581. url: https://www.jstor.org/stable/167703.

Cuervo, D. P., Goos, P., Sörensen, K., and Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, pages 454–464.

Dang, Y., Allen, T. T., and Singh, M. (2022a). A heterogeneous vehicle routing problem with common carriers and time regulations: Mathematical formulation and a two-color ant colony search. *Computers Industrial Engineering*. doi:10.1016/j.cie.2022.108036.

Dang, Y., Allen, T. T., and Singh, M. (2022b). A heterogeneous vehicle routing problem with common carriers and time regulations: Mathematical formulation and a two-color ant colony search. *Computers Industrial Engineering*. doi:10.1016/j.cie.2022.108036.

Dantzig, G. and Ramser, J. (1959). The truck dispatching problem. *Management Science*, (6):80–91.

David J. Rader, J. (2010). *DETERMINISTIC OPERATIONS RESEARCH Models and Methods in Linear Optimization*. Wiley, 1st edition.

Doan, T. T., Bostel, N., and Há, M. H. (2021). The vehicle routing problem with relaxed priority rules. *EURO Journal on Transportation and Logistics*. doi:10.1016/j.ejtl.2021.100039.

Dror, M. and Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 23(2):141–145. doi:10.1287/trsc.23.2.141.

Duhamel, C., Potvin, J.-Y., and Rousseau, J.-M. (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science*, pages 49–59. doi:10.1287/trsc.31.1.49.

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, pages 109–134.

Fisher, M. L. and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *NETWORKS*, pages 109–124. doi:10.1002/net.3230110205.

Fleischmann, B. (1990). The vehicle routing problem with multiple use of the vehicles. Technical report, Fachbereich Wirtschaftswissenschaften, Universität Hamburg, Hamburg, Germany.

Garside, A. K. and Laili, N. R. (2019). A cluster-first route-second heuristic approach to solve periodic multi-trip vehicle routing problem. *Jurnal Teknik Industri*, pages 172–181. doi:10.22219/JTIUMM.Vol20.No2.172-181.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Comput. & Ops. Res*, pages 533–549.

Haddadene, S. R., Labadie, N., and Prodhon, C. (2016). A grasp × ils for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems With Applications*, pages 274–294.

Hansen, P. and Mladenovic, N. (2005). Variable neighborhood search. In *Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques*, pages 211–238. Springer US. doi:10.1007/0-387-28356-0_8.

Hansen, P., Mladenovic, N., and Pérez, J. A. (2010). Variable neighborhood search: methods and applications. *Annals of Operations Research*, pages 367–407. doi:10.1007/s10479-009-0657-6.

Heerkens, H. and van Winden, A. (2017). *Solving Managerial Problems Systematically*. Noordhoff Uitgevers.

Higino, W., Chaves, A. A., and de Melo, V. V. (2018). Biased random-key genetic algorithm applied to the vehicle routing problem with private fleet and common carrier. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE. doi:10.1109/CEC.2018.8477905.

ITPPackaging (2020). A guide to industrial packaging. https://itppackaging.com/a-guide-to-industrial-packaging. Accessed on: 2023-07-03.

Jaballah, A. and Cherif-Khettaf, W. R. (2021). Multi-trip pickup and delivery problem, with split loads, profits and multiple time windows to model a real case problem in the construction industry. In *10th International Conference on Operations Research and Enterprise Systems(ICORES)*. doi:10.5220/0010253002000207.

Kallehauge, B., Larsen, J., Madsen, O., and Solomon, S. M. (2005). Vehicle routing problem with time windows. doi:10.1007/0-387-25486-2_3.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, pages 671–680.

Laporte, G., Gendreau, M., Potvin, J.-Y., and Semet, F. (1999). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*. doi:10.1111/j.1475-3995.2000.tb00200.x.

Laporte, G. and Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics 31*, pages 147–184. doi:10.1016/S0304-0208(08)73235-3.

Laporte, G., Ropke, S., and Vidal, T. (2014). Heuristics for the vehicle routing problem. In *Vehicle Routing Problems, Methods, and Applications*, pages 87–116. Society for Industrial and Applied Mathematics, PA, United States. doi:10.1137/1.9781611973594.ch4.

Marinakis, Y. (2017). Multiple phase neighborhood search-grasp for the capacitated vehicle. *Expert Systems with Applications*, pages 6807–6815. doi:10.1016/j.eswa.2012.01.015.

McNabb, M. E., Weir, J. D., Hill, R. R., and Hall, S. N. (2015). Testing local search move operators on the vehicle routing problem with split deliveries and time windows. *Computers Operations Research*, 56:93–109. doi:10.1016/j.cor.2014.11.007.

MeilinkB.V. (n.d.). Family meilink. https://meilink.eu/en/about-us/history/. Accessed on: 2023-07-03.

Miranda, D. M. and Conceição, S. V. (2016). The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, pages 104–116.

Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search. *Computers Operations Research*, 24(11):1097–1100.

Mohammadi, M., Mahmoodian, N., and Mohammadi, H. (2022). A simulated annealing approach (sa) to vehicle routing problem with time windows. In *8th International Conference on Control, Instrumentation and Automation (ICCIA)*, pages 1–6, Tehran, Iran, Islamic Republic. IEEE.

Nagy, G., Wassan, N. A., Speranza, M. G., and Archetti, C. (2015). The vehicle routing problem with divisible deliveries and pickups. *Transportation Science*, pages 271–294.

Ni, Q. and Tang, Y. (2023). A bibliometric visualized analysis and classification of vehicle routing problem research. *Sustainability*. doi:10.3390/su15097394.

Nugroho, S. M., Nafisah, L., Khannan, M., Mastrisiswadi, H., and Ramdhani, M. (2020). Vehicle routing problem with heterogeneous fleet, split delivery, multiple product, multiple trip, and time windows: A case study in fuel distribution. In *IOP Conf. Series: Materials Science and Engineering 847*.

Ombuki, B. R. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, pages 17–30.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for vehicle routing problem. *Annals of Operations Research*, 41:421–451.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, pages 21–51.

Polat, O. (2017). A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. *Computers and Operations Research*, pages 71–86.

Rizzoli, A., Montemanni, L. E., and Gambardella, L. (2007). Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, page 135–151. doi:10.1007/s11721-007-0005-x.

Ropke, S. and Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, pages 750–775. doi:10.1016/j.ejor.2004.09.004.

Silva, M. M., Subramanian, A., and Ochi, L. S. (2015). An iterated local search heuristic for the split delivery vehicle routing problem. *Computers and Operations Research*, pages 234–249.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, pages 254–265. doi:10.1287/opre.35.2.254.

Stojanović, (2017). Road freight transport outsourcing trend in europe – what do we really know about it? *Transportation Research Procedia*, 25:772–793. doi:10.1016/j.trpro.2017.05.457.

Suprayogi and Priyandari, Y. (2017). Tabu search for the vehicle routing problem with multiple trips, time windows and simultaneous delivery-pickup. *Jurnal Teknik Industri*, pages 75–82. doi:10.9744/jti.19.2.75-82.

Tezuka, K. (2011). Rationale for utilizing 3pl in supply chain management: A shippers' economic perspective. *IATSS Research*, pages 24–29. doi:10.1016/j.iatssr.2011.07.001.

TheExpressWire (2023). Industrial packaging market shows magnificent growth during forecasting period 2023 to 2030— research by industry experts. https://www.benzinga.com/pressreleases/23/09/34175777/industrial-packaging-market-shows-magnificent-growth-during-forecasting-period-2023-to-2030-researc. Accessed on: 2023-07-03.

Toth, P. and Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, pages 487–512. doi:10.1016/S0166-218X(01)00351-1.

Tütüncü, G. Y., Carreto, C. A., and Baker, B. M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, pages 138–157. doi:10.1016/j.omega.2006.11.001.

Wang, Y., Ma, X., Lao, Y., Wang, Y., and Mao, H. (2013). Vehicle routing problem simultaneous deliveries and pickup with split loads and time windows. *Journal of the Transportation Research Board*, pages 120–128. doi:10.3141/2378-13.

Wassan, N., Wassan, N., Nagy, G., and Salhi, S. (2017). The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers Operations Research*, 78:454–467.

Wassan, N. A. and Nagy, G. (2014). Vehicle routing problem with deliveries and pickups: Modelling issues and meta-heuristics solution approaches. *International Journal of Transportation*, pages 95–110. doi: 10.14257/ijt.2014.2.1.06.

Wei, L., Zhang, Z., Zhang, D., and Leung, S. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, pages 843–859.

Zhang, Z., Che, Y., and Liang, Z. (2023). Split-demand multi-trip vehicle routing problem with simultaneous pickup and delivery in airport baggage transit. *European Journal of Operational Research*, pages 996–1010. doi: 10.1016/j.ejor.2023.07.028.

Çatay, B. (2009). Ant colony optimization and its application to the vehicle routing problem with pickups and deliveries. In *Natural Intelligence for Scheduling, Planning and Packing Problems*. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-04039-9_9.

Appendix

A Algorithms of the Metaheuristics Discussed in Literature Review

In this section, the algorithms for the metaheuristics reviewed in Section 3.8 are presented.

A.1 Simulated Annealing Algorithm

Algorithm 13 Simulated Annealing Algorithm

Construct the initial solution S_0
$S \quad S_0$, $T \quad T_0$, $T^0 \quad T_0$
while time limit is not exceeded **do**
    for $k = 1$ to L_k **do**
        Select a neighborhood structure NS randomly
        Generate a feasible solution S^0 from S with NS
        if cost(S^0) $<$ cost(S) **then**
            $S \quad S^0$
        else
            Set $S = S^0$ with probability p, where $p = \exp\left(\quad \frac{\text{cost}(S') \quad \text{cost}(S)}{T}\right)$
        end if
        if cost(S^0) $<$ cost($S \quad$) **then**
            $S \quad S^0$, $T^0 \quad T$
        end if
        $T \quad \alpha \quad T$
        if $T < 0.01$ **then**
            $T^0 \quad \frac{T'}{2}$, $T \quad \min(T^0, L(T))$
        end if
    end for
end while
return S

A.2 Tabu Search Algorithm

Algorithm 14 Tabu Search Algorithm

$x \quad$ FindInitialSolution()
TabuList $\quad ;$
repeat
    $y \quad$ BestMove($N(x)$)
    if y 2̸ TabuList **then**
        $x \quad y$
    else if y 2 TabuList **and** Aspiration Criteria met **then**
        $x \quad y$
    else
        Do not update x
    end if
    UpdateTabuList(x)
until Maximum Iterations Reached
return x

A.3 Iterated Local Search

Algorithm 15 Iterated Local Search (ILS)

S_{best} FindInitialSolution()
for i 1 **to** numIterations **do**
    S_{pert} Perturb(S_{best})
    S_{next} LocalSearch(S_{pert})
    if Cost(S_{next}) $<$ Cost(S_{best}) **then**
        S_{best} S_{next}
    end if
end for
return S_{best}

A.4 Variable Neighborhood Descent

Algorithm 16 Basic Variable Neighborhood Descent (VND)

x FindInitialSolution()
k 1
while k $k_{\max}$ **do**
    x^0 BestImprovement(x, N_k)
    if Cost(x^0) $<$ Cost(x) **then**
        x x^0
        k 1
    else
        k $k+1$
    end if
end while
return x

A.5 Greedy Randomized Adaptive Search

Algorithm 17 Generic GRASP

procedure GRASP
    InputInstance()
    for GRASP stopping criterion not satisfied **do**
        ConstructGreedyRandomizedSolution(Solution)
        LocalSearch(Solution)
        UpdateSolution(Solution, BestSolutionFound)
    end for
    return BestSolutionFound
end procedure

Algorithm 18 GRASP Construction Phase

procedure ConstructGreedyRandomizedSolution(*Solution*)
    Solution = *fg*
    for construction not done do
        MakeRCL(RCL)
        s SelectElementAtRandom(RCL)
        Solution = Solution *[fsg*
        AdaptGreedyFunction(s)
    end for
end procedure

B Customer locations served predominantly by external vehicles (common carriers)

B.1 Experiment E1

Figure 13. Locations served by common carriers according to the MILP results from Experiment E1.

Figure 13 illustrates the locations served by common carriers as determined by the MILP results from Experiment *E1*. This map aggregates the demand nodes visited by external vehicles across all 10 real-world data instances. Notably, in instances with high demand, the MILP consistently selects specific locations for service by common carriers. These locations include Winterswijk, Spankeren, Papendrecht, Varsseveld, Denekamp, Delden, and Veldhoven, with additional locations depicted in Figure 13.

B.2 Experiment E4

In Figure 14, the locations served by common carriers are presented, as determined by the MILP model across all ten scenarios. This map highlights the demand nodes that external carriers visit, showcasing a strategic allocation of common carriers to the most distant locations from the depot, particularly in the Noord Holland and Zuid Holland regions.

Figure 14. Locations served by common carriers according to the MILP results from Experiment E4.