MSc Industrial Engineering and Management
Graduation Thesis

Optimising Multi-Compartment Routing in Cross-Docking Networks: A Hybrid TS-SA Approach

Gerben Johannes Leonardus de Groot

Supervisor: Patricia Rogetzer

April, 2025

Department of High-Tech Business and Entrepreneurship
Faculty of Behavioural, Management and Social Sciences

University of Twente

UNIVERSITY OF TWENTE.

Preface

Dear reader,

In front of you lies my master's thesis: "*Optimising Multi-Compartment Routing in Cross-Docking Networks: A Hybrid TS-SA Approach*". This research was conducted at CAPE Groep in Enschede and marks the end of my life as an Industrial Engineering and Management student at the University of Twente.

At CAPE Groep, I was welcomed by one of the team leads in Transportation and Logistics and given the opportunity to align my interests with a challenging and relevant assignment for one of their clients. This eventually led to collaboration with the IT Manager at Wolter Koops, and I would like to express my gratitude for his cooperation in developing the assignment and his invaluable support throughout the research.

Special thanks go to my supervisor, Sem Odink, who worked closely with Wolter Koops and helped me to connect with both companies. His insightful feedback and collaborative approach significantly contributed to shaping this thesis. I also want to thank Mart Busger op Vollenbroek, Gilang Charismadiptya, and Maik Wesselink at CAPE Groep for their assistance during the study and the development of the Mendix application.

Moreover, I am grateful to my supervisors at the University of Twente. My first supervisor, Patricia Rogetzer, provided constant guidance throughout this process. Our weekly meetings were particularly helpful, where we discussed every detail of this thesis. I also wish to thank my second supervisor, who introduced me to Patricia and offered valuable feedback that helped position my work within the academic literature.

Finally, I would like to acknowledge my fellow students graduating at CAPE Groep, who were always open to discussions and provided encouragement along the way.

I hope you enjoy reading my master's thesis!

Gerben de Groot

Enschede, 2025

Management Summary

This thesis is conducted as part of the Master's in Industrial Engineering and Management at the University of Twente, in collaboration with CAPE Groep and Wolter Koops. CAPE Groep, a consultancy firm, facilitated the research while the assignment was carried out for Wolter Koops, a leading logistics service provider specialising in temperature-controlled transportation. The study examines a specific case of vehicle routing, focusing on the challenges of routing multi-compartment vehicles in a supply chain that uses a Cross-Dock (CD) in its customer delivery channel.

Wolter Koops operates a logistics network in which goods from multiple suppliers are consolidated at a CD. At this central consolidation point, incoming shipments may be sorted and reassigned to outbound vehicles before being transported to retailers. Vehicle routing and consolidation at the CD require careful planning because commodity incompatibilities prevent shared transport within a single compartment, while each vehicle is equipped with two compartments. These decisions involve complex trade-offs between vehicle availability, time windows, and compartmentalisation constraints. Still, the company currently lacks a standardised algorithm for these decisions, requiring planners to make these decisions manually. This research investigates whether an algorithm can be developed to standardise and improve routing and consolidation, evaluate the efficiency of the current planning method, and identify opportunities for improvement. Thus, the central research question is:

How can an optimisation algorithm be developed to minimise the total cost of servicing all suppliers and retailers in Wolter Koops' fleet routing operations?

A literature review was conducted to explore existing Vehicle Routing Problem (VRP) solutions, particularly those considering cross-docking operations and multi-compartment vehicle constraints. Metaheuristic approaches have shown promise in solving large-scale VRP instances. Still, current models of the Vehicle Routing Problem with Cross-Docking (VRPCD) do not fully capture the intricacies of multi-compartment vehicle constraints, cross-docking synchronisation, strict time windows, and rest periods, all of which are relevant to Wolter Koops' logistics operations. This research develops a tailored approach to address the complexity of integrating cross-docking operations and addressing fleet-specific constraints in temperature-sensitive logistics, as it is currently underexplored in the literature. Related work is explored in Section 3.6.

A second literature review was conducted to identify suitable methods for optimising multi-compartment vehicle routing in a cross-docking network. While exact methods guarantee optimal solutions, their computational infeasibility for large-scale instances renders them impractical for real-world routing problems. Therefore, heuristic methods were explored, and the hybrid Tabu Search–Simulated Annealing (TS-SA) solution approach was proposed to leverage the memory-based nature of TS while incorporating SA-based diversification to escape local optima, improving solution robustness. Furthermore, TS-based implementations have shown promising results in other VRPCD implementations.

This study develops a hybrid TS-SA approach to optimising multi-compartment vehicle routing in a cross-docking network using past data from two days in January 2025. The

solution is generalisable in similar supply chains (for more details on generalisability, see Section 8.2.3). The approach improves an initial solution generated by a two-stage greedy insertion heuristic. The algorithm routes vehicles to efficiently collect, consolidate, and deliver goods for all transportation orders. The cost model considers fixed costs associated with vehicle deployment and variable expenses related to travel time. Each order is pre-assigned to a CD before planning, consistent with Wolter Koops' manual planning process. To validate the model, orders from the EG department (flowers, plants, fruits, and vegetables) routed through the Venlo CD are used to compare with historical planning. Additionally, multiple scenarios and sensitivity analyses, including variations in cost, time window, and travel speed, were conducted to assess the robustness of the proposed TS-SA model.

The algorithm was tested on a two-day dataset of 133 historical orders from Wolter Koops to benchmark performance against manual planning. In this sample, the model has shown:

- The hybrid TS-SA algorithm reduces total transportation costs by 31.8% compared to the historical planning of the two-day sample.
- Load efficiency improves by 8.1%, reducing the number of vehicles required to service all transport orders compared to the historical planning of the two-day sample.
- The algorithm successfully plans 133 orders within Wolter Koops' cross-docking supply chain, effectively handling multi-compartment constraints while minimising costs and ensuring solution feasibility in approximately 3.5 hours.
- The algorithm has shown superior performance to the historical planning across all scenarios and sensitivity analyses.

As a consequence of the insights obtained in this study and the limitations of the model, we would like to provide recommendations for Wolter Koops on how to proceed based on the results of this study. To adopt the model and enhance operational efficiency, Wolter Koops should:

1. Analyse data inconsistencies encountered during the development of the algorithm. More details on these issues can be found in Section 6.1.
2. Integrate the algorithm into existing Transportation Management Systems (TMS) to allow automated routing decision testing.
3. Conduct real-world pilot tests over an extended period to validate algorithmic performance under dynamic conditions.
4. Perform more testing on the optimal parameter setup in different problem settings.

This research contributed to routing optimisation by demonstrating the effectiveness of hybrid metaheuristics in solving a complex variant of the VRP, which, to our knowledge, is not available in the current body of literature. Future research could provide more analysis on (self-)tuning parameters, extend the model to dynamic routing scenarios, incorporate stochastic demand/time variations, or optimise dock scheduling at the CD to improve cross-docking efficiency.

Contents

1 Problem Context and Scope Definition **1**
1.1 Company Backgrounds . . . 1
1.1.1 Wolter Koops . . . 1
1.1.2 CAPE Groep . . . 2
1.2 Problem Identification . . . 2
1.2.1 Problem Context and Action Problems . . . 3
1.2.2 Core Problem and Motivation . . . 4
1.2.3 Research Objective . . . 5
1.3 Problem-Solving Approach and Research Questions . . . 5
1.3.1 Phase 1: Problem Identification . . . 6
1.3.2 Phase 2: Theoretical Foundations . . . 6
1.3.3 Phase 3: Solution Design . . . 7
1.3.4 Phase 4: Modelling and Implementation . . . 7
1.3.5 Phase 5: Evaluation . . . 8
1.4 Research Design . . . 8
1.4.1 Research Scope . . . 8
1.4.2 Limitations . . . 9
1.4.3 Research Methods . . . 9
1.5 Deliverables . . . 10
1.6 Research Framework . . . 11

2 Current Vehicle Routing Practices at Wolter Koops **12**
2.1 Planning Procedure . . . 12
2.1.1 Operational Roles and Decision-Making . . . 12
2.1.2 Operational Data and Supporting Tools . . . 13
2.1.3 Decision Factors in Vehicle Routing . . . 14
2.1.4 Stakeholder Analysis . . . 16
2.2 Business Objectives and Current Performance . . . 17
2.3 Findings and Implications . . . 18

3 Vehicle Routing Problem: Models and Applications **20**
3.1 Introduction to the Vehicle Routing Problem . . . 20
3.2 VRP with Capacity Constraints . . . 21
3.3 VRP with Time Windows . . . 24
3.4 VRP with Multiple Commodities . . . 26
3.5 VRP with Cross-Docking . . . 29
3.6 Related Work . . . 31
3.7 Findings and Implications . . . 33

4 Problem Formulation and Mathematical Framework **35**

4.1 Formal Problem Statement 35
4.2 Graph-Based Representation of the Network 36
4.3 Mathematical Formulation 39
4.3.1 Decision Variables 39
4.3.2 Objective Function and Constraints 39
4.4 Assumptions and Simplifications 41
4.4.1 Assumptions 41
4.4.2 Simplifications 42
4.5 Toy Problem Instance 42
4.6 Findings and Implications 44

5 Adaptive TS-SA Hybrid Solution Approach 45
5.1 Exploration of Suitable Solution Methods 45
5.1.1 Exact Methods 45
5.1.2 (Meta)heuristic Methods 45
5.2 Proposed Tabu Search Implementation 49
5.2.1 Initial Solution Generation 50
5.2.2 Neighbourhood Structures 50
5.2.3 Tabu List and Aspiration Conditions 51
5.2.4 Stopping Criteria 52
5.3 Adaptive Simulated Annealing Integration in Tabu Search 52
5.3.1 Activation and Stagnation Detection 53
5.3.2 Temperature Cooling Schema 53
5.3.3 Probabilistic Move Acceptance and Tabu Constraints 54
5.3.4 Balancing Intensification and Diversification 55
5.4 Findings and Implications 55

6 Model Validation and Experimental Design 56
6.1 Input Data and Filtering 56
6.2 Parameter Tuning 58
6.3 Case Study Validation 62
6.4 Experimental Design 63
6.4.1 Scenarios and Sensitivity Analysis 63
6.4.2 Evaluation Criteria 65
6.5 Findings and Implications 65

7 Experimental Execution and Computational Results 67
7.1 Test Scenario Execution 67
7.1.1 Scenario 1: Algorithmic Variation on the Fixed Cost 67
7.1.2 Scenario 2: Impact of Compartmentalisation 68
7.1.3 Results and Comparison Between Scenarios 68
7.2 Sensitivity Analysis 70
7.2.1 Cost Sensitivity 70
7.2.2 Travel Speed Sensitivity 71
7.2.3 Time Window Tightness Sensitivity 73

8 Conclusions and Recommendations 75
8.1 Conclusions 75
8.2 Contributions to Science and Practice 76
8.2.1 Scientific Contributions 77

8.2.2 Practical Contributions . . . 77
8.2.3 Generalisability and Discussion . . . 78
8.3 Recommendations . . . 79
8.4 Limitations and Future Research . . . 79
8.4.1 Methodological Limitations . . . 79
8.4.2 Practical Limitations . . . 80
8.4.3 Future Research Directions . . . 80

A Mathematical Modelling Techniques **91**
A.1 Two-index vehicle flow formulation . . . 91
A.2 Commodity-flow formulations . . . 91
A.3 Set-partitioning formulation . . . 92

B Toy Problem Coordinates **93**

C Initial Solution Generation **94**
C.1 Stage 1: Retailer/Delivery Routes . . . 94
C.2 Stage 2: Supplier/Pick Up Routes . . . 94

D Data Filtering Process **97**

List of Figures

1.1 Illustration of the flow of commodities A and B from suppliers to retailers via a cross docking facility, with consolidation of the different commodities . 2
1.2 Commodity flow between inbound and outbound docks at a cross-docking facility. Arrows indicate the movement of goods within the facility. In this example, inbound and outbound vehicles are not restricted to specific terminals; their allocation depends on operational requirements [1]. 3
1.3 Problem cluster Wolter Koops 4

2.1 Tools used in the planning process and the role of the routing algorithm. . . 14
2.2 Factors Influencing Routing Decisions 16

3.1 The VRP family hierarchy [2] 21

4.1 Graph representation of the VRPCD 37
4.2 Individual route possibilities (idle state omitted) 38
4.3 Toy problem solution illustration 43

5.1 Example of the neighbourhood operators SWAP (a) and INSERTION (b) for generating a neighbour solution 51
5.2 Temperature cooling schema with $T_0 = 1000$, $\alpha_{\text{int}} = 0.9$, and $\alpha_{\text{ext}} = 0.9$. . . 54

6.1 Objective Value Evolution over the Running Time of Experiment 1 60
6.2 Optimisation History of the Optuna Parameter Tuning Process. 60
6.3 Parameter Importance according to Optuna 61
6.4 Objective Value Evolution 62

7.1 Comparison of vehicle routes and costs between the base scenario (a) and the current scenario (b). The timelines at the top indicate the difference in total fixed costs. 68
7.2 Objective Value Evolution over the Running Time for Scenarios 1 and 2 . . 69
7.3 Objective Value over the Running Time for Different Travel Speeds 72
7.4 Average Acceptance Probability per Iteration over the Running Time 72
7.5 Objective Value Evolution over the Running Time for Different Time Window Tightness Sensitivities 74

C.1 Outline of the two-stage approach employed in the constructive heuristic . . 94

D.1 Data Filtering Process 97

List of Tables

1.1 Research Framework 11

2.1 Planning groups at Wolter Koops 13

3.1 Three-index vehicle flow notations (CVRP). 24
3.2 Overview of VRPCD Literature 33

4.1 Transport Orders 43
4.2 Euclidean distance/travel time/cost matrix 43
4.3 Vehicle schedules with arrival- and departure times 44

6.1 Description of Input Data Attributes 57
6.2 Parameter Notation and Configuration Space 59
6.3 Optimised Parameter Configuration 61
6.4 Comparison of Historical Planning and Hybrid TS-SA Model Performance . 63
6.5 Sensitivity experiments overview. 64

7.1 Historical Planning and TS-SA Model Performance (all scenarios) 69
7.2 Cost Sensitivity Analysis Results 70
7.3 Travel Speed Sensitivity Analysis Results 71
7.4 Time Window Tightness Sensitivity Analysis Results 73

B.1 Toy problem node characteristics 93

List of Abbreviations

ALNS Adaptive Large Neighbourhood Search

ACO Ant Colony Optimisation

B Belgium

CD Cross-Dock

CVRP Capacitated VRP

DA Deterministic Annealing

EDI Electronic Data Interchange

EG Europese Gemeenschap uitgezonderd vlees en zuivel

ETA Estimated Time of Arrival

F France

GA Genetic Algorithm

GER Germany

GRASP Greedy Adaptive Search Procedure

IE Ireland

ILS Iterated Local Search

KPI Key Performance Indicator

LNS Large Neighbourhood Search

MCmpt-VRP Multi-Compartment VRP

MCVRP Multi-Commodity VRP

MPSM Managerial Problem Solving Method

MTVRP Multi-Trip VRP

MTZ Miller-Tucker-Zemlin

NL the Netherlands

PL Poland

PR Path Relinking

SA Simulated Annealing

SS Scatter Search

TS Tabu Search

TSP Traveling Salesman Problem

TMS Transportation Management System

UK United Kingdom

VNS Variable Neighborhood Search

VRP Vehicle Routing Problem

VRPCD VRP with Cross-Docking

VRPCDTW VRPCD with Time Windows

VRPHTW VRP with Hard Time Windows

VRPSTW VRP with Soft Time Windows

VRPTW VRP with Time Windows

1 Problem Context and Scope Definition

This thesis presents the research conducted in collaboration with CAPE Groep and Wolter Koops as part of the requirements for the Master of Science in Industrial Engineering and Management at the University of Twente. The study examines a specific case of vehicle routing, focusing on the challenges of routing multi-compartment vehicles in a supply chain that uses a Cross-Dock (CD) in its customer delivery channel. The CD serves as a central consolidation point where incoming shipments are sorted and reassigned to outbound vehicles without long-term storage. A key challenge arises because many commodities are incompatible and cannot be transported together in a single compartment, necessitating careful compartmentalisation and routing decisions. This introductory chapter starts with Section 1.1, which presents background information on the companies where this research is conducted. Section 1.2 identifies the problem that is the topic of this research, sets the problem into context, and describes the research objectives. In Section 1.3, the problem-solving approach and the research questions are outlined. Section 1.4 shapes the research design by elaborating on the research scope, limitations, and methods used throughout this thesis. Section 1.5 presents the deliverables of this study. The outline of this study is presented in section 1.6, which divides the research phases over the chapters of this study.

1.1 Company Backgrounds

This study focuses on logistical challenges faced by Wolter Koops, an international transportation company specialising in cold chain logistics via road / by vehicles. Section 1.1.1 elaborates on the background information of this company. This research was conducted at CAPE Groep, a software development and system integrator consultancy company which provided support and expertise during the research process. Therefore, Section 1.1.2 includes background information on this company.

1.1.1 Wolter Koops

Wolter Koops, established in 1961, specialises in the temperature-controlled logistics market. With over 2,500 employees, the company has grown into an international service provider, operating a fleet of around 1,000 vehicles and completing, on average, over 5,500 weekly trips, where a trip is defined as any travel between two locations, such as a delivery to a retailer [3, 4]. The company's headquarters is in Zeewolde, the Netherlands, but Wolter Koops has established multiple locations throughout Europe. The company operates from the Netherlands (NL), Germany (GER), and Poland (PL), with specific locations in Zeewolde, Venlo (NL), Osterweddingen (GER), Alzenau (GER), and Komorniki (PL). While most of their customers (retailers) are based in the Netherlands and Germany, the company also serves clients in Belgium (B), France (F), and Ireland (IE), with occasional requests from other European countries [5].

Wolter Koops primarily serves wholesalers and distributors of perishable goods who submit transportation requests to facilitate the delivery of products from third-party suppliers. For instance, the distribution centre of a supermarket chain may request the collection and delivery of dairy products from a designated third-party supplier. In these cases, companies specify the required product quantity, the pick up location, the delivery destination, and a time window for the delivery. Handling hundreds of transportation requests daily, Wolter Koops directs goods to one of its own CD facilities, where the products may be consolidated and loaded onto one or more other vehicles for delivery to respective retailers. This concept is exemplified in Figure 1.1, which depicts the flow of two distinct commodities through the cross docking process.

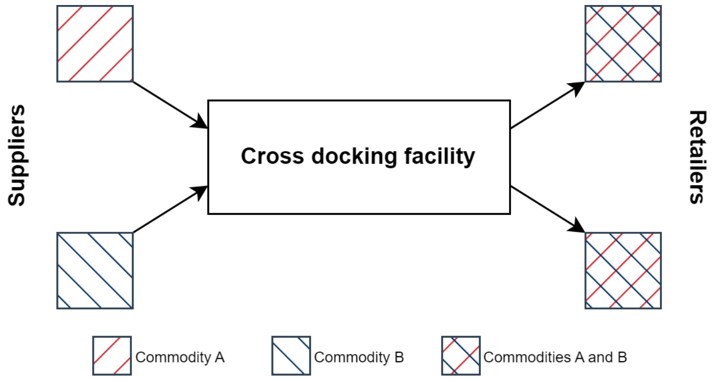


FIGURE 1.1: Illustration of the flow of commodities A and B from suppliers to retailers via a cross docking facility, with consolidation of the different commodities

1.1.2 CAPE Groep

CAPE Groep was founded in Hellendoorn, the Netherlands, in 2000 and is based in Enschede, the Netherlands. Since its founding, the company has expanded its operations domestically to Utrecht and internationally to Sydney, Australia and Zagreb, Croatia. It is a consultancy firm specialising in developing custom software solutions to digitise and optimise business processes through the low-code platform Mendix [6]. CAPE Groep is active in the transportation, logistics, supply chain, construction, and agrifood sectors [7].

Over the past two years, CAPE Groep has developed various custom software solutions for Wolter Koops. These include a customer portal enabling retailers and suppliers to place their orders, track shipment status, and monitor the location of returnable transport items (such as Euro pallets, flower bins, and stackable containers). Furthermore, CAPE Groep has built applications with functionalities for scanning and identifying products in a CD. Recently, the company introduced a digital service that offers real-time insights into the Estimated Time of Arrival (ETA) of shipments and departure times, hereafter referred to as the “ETA service” [8].

1.2 Problem Identification

This research concerns the Wolter Koops’ CD located in Venlo, where approximately 18 of the 26 planners are based. These planners operate based on demand and vehicle availability to compose a routing scheme that serves all retailers (customers) in time. The routing decision entails determining a path or sequence of routes that a fleet of vehicles should follow to service a set of retailers or locations while minimising the relevant costs incurred.

At the CD, goods brought by an incoming vehicle from a supplier are unloaded, sorted, stored for a short period, and loaded onto their respective outgoing vehicles. The commodity flow between the inbound and outbound docks is illustrated in Figure 1.2. Typically, incoming goods are scanned to identify their destination and sorted into the outbound vehicle, often combining multiple commodities into one vehicle. The decision regarding the unloading, sorting, and loading onto the outgoing vehicle is called the consolidation decision. Consolidation and routing decisions are interdependent because the availability of goods, which is determined by the consolidation decisions, serves as input for the routing decision.

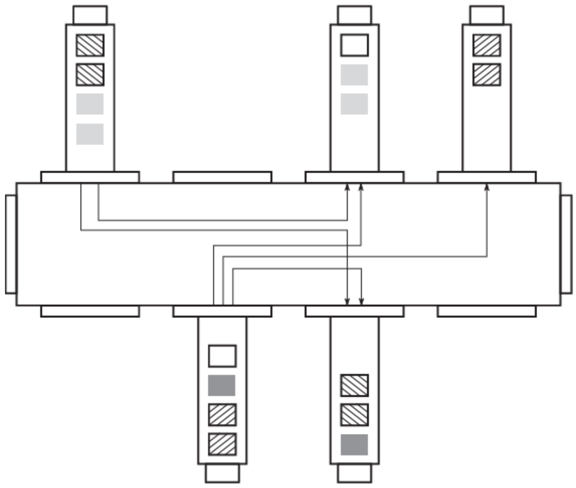


FIGURE 1.2: Commodity flow between inbound and outbound docks at a cross-docking facility. Arrows indicate the movement of goods within the facility. In this example, inbound and outbound vehicles are not restricted to specific terminals; their allocation depends on operational requirements [1].

The remainder of this section details the problem context of this research, identifies the core problem studied using a problem cluster, and outlines the research objectives. Problem identification is the first phase of the research framework adopted in this research, similar to the Managerial Problem Solving Method (MPSM) by Heerkens and van Winden [9].

1.2.1 Problem Context and Action Problems

After an inventory of existing problems was made, the problem context and its connections were mapped into a problem cluster. According to Heerkens and van Winden [9], a problem cluster maps all problems along with their connections. It is a model in which connections are indicated in a cluster of causes and effects, and it serves as a valuable tool to bring order to the problem context and to identify the core problem. Figure 1.3 depicts the problem cluster at Wolter Koops. In the problem cluster, blocks represent problems and the cause and effect relationships with arrows. In this subsection, the action problems are presented, defined as anything or any situation that is not how you want it to be. It is where a discrepancy between the norm and the reality is perceived by the problem owner [9]. In the case of Wolter Koops, the following action problems were expressed:

Wolter Koops is obliged to hire vehicles from third-party providers at significantly higher rates due to an insufficient internal fleet.

Hiring vehicles from third-party providers creates a discrepancy between the norm and the company's reality because hiring these results in a substantial cost increase. Wolter Koops currently hires approximately 100 vehicles externally, whereas the company's operational norm is to meet all demands using its internal fleet alone. As the problem cluster shows, fleet undercapacity is caused by the many trips vehicles need to make due to the suboptimal routing of vehicles and the underutilised vehicle capacity due to the many unnecessary empty kilometres driven.

Wolter Koops cannot identify improvement points in the routing and consolidation planning procedure.

Wolter Koops is unaware of potential improvements in the current planning procedure due to its lack of standardisation, complexity, and limited transparency regarding planning efficiency. A company representative emphasised that gaining insights into the quality of the planning process would be valuable, as there are currently no means for management to assess its effectiveness. Enhancing vehicle routing decisions could lead to more efficient

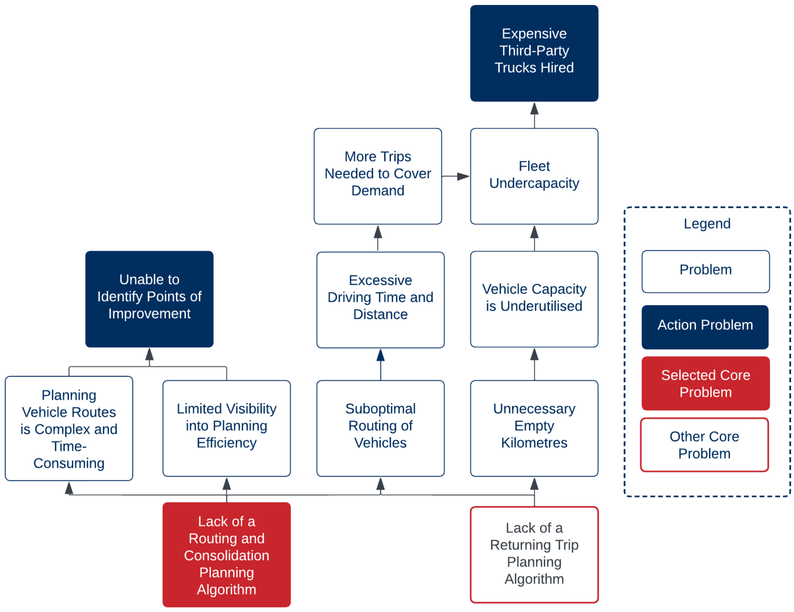


FIGURE 1.3: Problem cluster Wolter Koops

operations by reducing the number of kilometres driven and the time required to complete all trips. This improvement may also reduce the number of vehicles used, which is particularly significant given the current situation at Wolter Koops, where over 100 vehicles are hired externally in addition to the company's fleet.

1.2.2 Core Problem and Motivation

In this subsection, potential core problems are identified by going back in the causal chain of problems depicted by the problem cluster. A core problem is identified when it does not have a direct cause. The possible core problems in the problem context provided are, therefore:

1. *The lack of a routing and consolidation planning algorithm.* Currently, no standardised algorithm efficiently routes vehicles from suppliers to retailers. Thus, planners

are tasked with manually planning trips and must account for spatial and temporal factors to determine the optimal routes and timing for all vehicles' movements. This becomes increasingly complex when the number of possible routes increases[1].

2. *The lack of a returning trip algorithm.* A separate team of planners manages the return trip to the CD once a vehicle has completed deliveries to its retailers. These planners aim to reduce empty kilometres by allowing the vehicle to pick up goods from suppliers on its way back, which can then be consolidated at the CD. However, this practice can extend the vehicle's time away from the CD, delaying its availability for future deliveries. Implementing an algorithm to standardise this decision-making process could improve fleet efficiency by optimising the balance between minimising empty kilometres and ensuring timely vehicle availability at the CD for subsequent trips.

Selecting a core problem involves identifying the issue that, when addressed, will yield the most significant improvements across the related action problems. Following discussions with representatives from Wolter Koops, the following core problem was identified:

Wolter Koops lacks a routing and consolidation planning algorithm.

1.2.3 Research Objective

This research aims to minimise the costs associated with the timely delivery of goods to retailers for Wolter Koops by optimising routing and consolidation decisions through an algorithm. The total cost of serving retailers is assumed to consist of fixed costs associated with each vehicle deployed and variable costs determined by the total time travelled in minutes. Moreover, the use of third-party logistics service providers can be minimised when vehicles are routed efficiently, as optimised routing reduces the need for external capacity by maximising the utilisation of available fleet resources. The proposed algorithm will also provide insights into route planning efficiency, as the current procedure is not standardised.

1.3 Problem-Solving Approach and Research Questions

This section presents the research framework for this applied study and details its phases and the corresponding research knowledge questions. The framework is grounded in the well-established MPSM methodology, which serves as the foundation for these phases. However, minor adjustments have been made to adapt the methodology to the specific context of this study. For a comprehensive explanation of the MPSM, readers are referred to Heerkens and van Winden [9]. To address the research objective outlined in the previous section, the central research question is as follows:

How can an optimisation algorithm be developed to minimise the total cost of servicing all suppliers and retailers in Wolter Koops' fleet routing operations?

This research's problem-solving approach consists of five phases containing research sub-questions. These phases and research questions constitute this paper's outline, formulated below. Before exploring what methods are available in the current body of literature, it is essential to understand the current planning situation comprehensively. Consequently,

[1] Even for a symmetric travelling salesman problem serving N customers, $\frac{(N-1)!}{2}$ possible routes can be composed [10]

phase 1 formulates multiple research sub-questions to explore the planning procedure at Wolter Koops further.

1.3.1 Phase 1: Problem Identification

This phase defines the core challenges by constructing a problem cluster and analysing the problem context to develop a clear understanding of its complexities. Identifying key issues early ensures that the research is well-aligned with the operational realities of the case study. Insights are gathered through expert interviews with Wolter Koops and CAPE Groep (Chapters 1 and 2).

1. *What are the current routing and consolidation decision-making processes at Wolter Koops?*

    1.1 *How is the planning department organised?*

    1.2 *What data and supporting tools are available at Wolter Koops to assist in routing decisions?*

The research questions in this phase aim to analyse the current routing and consolidation processes at Wolter Koops, providing a foundation for identifying potential improvements. Research question 1 examines the organisation of the planning department (1.1) and the availability of data and decision-support tools used in routing (1.2). Understanding these aspects is crucial for assessing the efficiency of existing operations and determining the feasibility of implementing optimisation strategies.

2. *What is the planning horizon and how many trips are routed in this time frame?*

The planning horizon and the number of trips routed within this time frame directly impact the complexity of routing decisions and the effectiveness of optimisation strategies. Research question 2 aims to clearly understand these factors at Wolter Koops, as they influence operational constraints and determine the most suitable solution method. Therefore, understanding these is essential to develop a model that aligns with real-world planning requirements.

1.3.2 Phase 2: Theoretical Foundations

This phase establishes the foundation for developing a solution by reviewing relevant literature on existing approaches. It contributes to answering the central research question by identifying best practices and methodologies for vehicle routing problems in cross-docking supply chains. The insights gained help shape the modelling and optimisation strategies used in later phases (Chapter 3).

3. *What are the key factors influencing routing and consolidation decisions in cross-docking logistics?*

Research question 3 explores the key factors that shape routing and consolidation decisions in cross-docking logistics. Understanding these aspects is crucial for building a model that aligns with real-world challenges and supports effective decision-making.

4. *How can the vehicle routing problem in a cross-docking supply chain be modelled and optimised?*

Research question 4 focuses on formulating and optimising the vehicle routing problem within a cross-docking supply chain. Given the complexity of such a supply chain, a practical model must account for consolidation, timing constraints, and cost efficiency. This question contributes to the theoretical foundation by focusing on the mathematical modelling and optimisation of routing decisions. It forms the basis for developing an effective solution method at Wolter Koops.

1.3.3 Phase 3: Solution Design

This phase focuses on designing a solution to the core problem, directly addressing the central research question. It provides a detailed model description, explores potential solution methods, and justifies selecting the most suitable approach. Additionally, key assumptions and limitations are outlined to clarify the scope and applicability of the solution. This phase ensures a structured and transparent approach to problem-solving (Chapters 4 and 5).

5. *What are the key assumptions made in the solution design?*

This research sub-question aims to document the assumptions made in the solution design to ensure validity, applicability, and reproducibility. Assumptions are critical in simplifying problem-solving, reducing computational time, and preserving solution quality wherever possible.

6. *What solution methods exist for enhancing vehicle routing decisions at Wolter Koops, and which is most suitable for implementation?*

Research question 6 focuses on identifying suitable solution methods to improve vehicle routing decisions at Wolter Koops, and given the complexity of cross-docking logistics, selecting the right approach—whether exact, heuristic, or hybrid—is crucial for balancing solution quality and computational efficiency.

1.3.4 Phase 4: Modelling and Implementation

This phase focuses on developing the solution method outlined in the previous phase to address the central research question. It involves key decisions regarding the model's search process, including parameter tuning, input data validation, and defining termination criteria. By refining these aspects, this phase ensures the model is both effective and applicable in a real-world setting (Chapters 6 and 7).

7. *How can the input data be processed and validated to ensure the model operates reliably?*

Research question 7 examines how input data can be processed and validated to ensure the model functions reliably and produces reproducible results. Accurate and well-structured data is essential for maintaining solution quality, as inconsistencies or errors can compromise validity and performance. This question focuses on preprocessing steps and validation techniques to ensure consistent outcomes.

8. *What key parameters influence the model's performance, and how can they be effectively tuned?*

Research question 8 explores the key parameters that impact the model's performance and how they can be effectively tuned. Parameter tuning is essential to the performance of the model, as it influences solution quality, computational efficiency, and convergence behaviour. This question focuses on identifying the tunable parameters and determining the optimal parameter configuration.

9. *How can the model be evaluated in a case study to assess its applicability to real-world routing scenarios?*

Research question 9 examines how the model can be evaluated through a case study to assess its applicability to real-world routing scenarios. A case study provides practical insights into how well the model performs under realistic conditions, considering operational constraints and business requirements. This question focuses on defining the evaluation criteria and comparing model outcomes to actual routing decisions.

10. *What experimental setup is required to systematically test the model's performance and robustness?*

Research question 10 focuses on designing an experimental setup to systematically test the model's performance and robustness. A well-structured experiment ensures that the model is evaluated under diverse conditions, assessing its reliability, scalability, and sensitivity to parameter changes. This question aims to define the testing framework to evaluate the model under changing conditions.

1.3.5 Phase 5: Evaluation

This phase evaluates the solution model and its performance, assessing whether it effectively addresses the core problem and meets the research objective (refer to Section 1.2.3). It examines Key Performance Indicators (KPIs), compares results to benchmark scenarios, and identifies potential areas for improvement. Additionally, recommendations for implementation are provided to ensure the model's practical applicability. Ultimately, this phase verifies whether the proposed solution achieves the intended outcomes and aligns with the research objectives (Chapter 8).

11. *What are the key findings, implications, and recommendations for practical implementation and further research?*

Research question 11 aims to summarise the study's key findings, assess their implications, and provide recommendations for both practical implementation and future research. Understanding how the proposed solution impacts real-world operations helps determine its feasibility and potential benefits. Additionally, identifying areas for further exploration ensures continuous improvement and refinement of vehicle routing and consolidation strategies.

1.4 Research Design

This section outlines the research design by detailing the scope, limitations, and methodologies to address the research objectives. It establishes the study's boundaries and ensures the research remains focused and applicable to the operational context of Wolter Koops. Additionally, the section discusses the research methods used in this study.

1.4.1 Research Scope

While the solution method is designed to be generalisable, its implementation is tested for applicability at the CD in Venlo. Accordingly, parameter tuning is performed using Venlo data to ensure the model's relevance and practical validity. Furthermore, the model is designed for cross-docking supply chains characterised by a structured flow, where suppliers are positioned upstream, retailers downstream, and the CD functions as a central hub.

This setup reflects a logistics network in which goods move from suppliers to the CD for consolidation and redistribution before reaching retailers.

The scope of this research encompasses key aspects of the routing process such as vehicle routing from suppliers to the CD, consolidation decisions, and the subsequent routing to retailers. Additionally, the study examines the possibility of direct routes from suppliers to retailers, although direct routes may only consist of one supplier and one retailer. This choice is to limit the model for the sake of computational tractability. The solution also accounts for complexities related to multi-compartment vehicles and product incompatibilities.

Furthermore, the number of vehicles employed is variable, with the optimal number determined based on the fixed costs incurred through vehicle use. Routing is conducted under static conditions, meaning that all relevant information is considered known a priori; dynamic routing scenarios are not addressed within this scope.

1.4.2 Limitations

This study is subject to certain limitations in its research design, which are discussed in this section. These limitations define the scope of the research, outlining constraints related to time, data availability, modelling assumptions, and practical applicability:

- As this research is conducted within the scope of a Master's thesis, time constraints limited the ability to explore all possible scenarios and fine-tune parameters individually. Parameter tuning is time-intensive, so not all configurations could be optimised separately. This impact is expected to be minimal as the solution space of all scenarios are similar. Additionally, scalability was not extensively tested across multiple datasets.
- The findings and parameter tuning are based on the case study from Venlo. While the model is designed to be generalisable to other locations, certain limitations may arise due to dataset-specific characteristics and operational differences. Furthermore, the model is intended for structured cross-docking supply chains where suppliers and retailers are geographically dispersed, with a CD serving as an intermediary.
- The model assumes that all routing decisions are made before execution without real-time adjustments. In practice, unexpected disruptions—delays, vehicle breakdowns, or last-minute order changes—can significantly impact routing decisions. This study does not account for dynamic re-optimisation during operations.
- The model operates assuming that all necessary data is available, accurate, and complete at the time of planning. However, real-world logistics often involve missing, uncertain, or evolving information.
- This research primarily addresses short-term operational routing decisions. Higher-level strategic decisions, such as fleet size determination, long-term investment planning, or network design, are beyond the scope of this study.

1.4.3 Research Methods

Different research methods will be used to answer the research questions above. For more elaborate descriptions of the research methods identified, refer to Turnhout et al. [11]. This has been consulted to find the combination of research methods needed to address the core problem defined in Section 1.2.2.

- A literature study: The literature study establishes a foundation of knowledge on vehicle routing problems and related topics. The search for relevant literature was conducted primarily using Google Scholar, Scopus, and FindUT, focusing on high-impact journals and conference proceedings. An exploratory method was adopted instead of following a predefined systematic review approach.

  The process begins by reviewing existing literature reviews and taxonomies on vehicle routing problems, which provide an overview of key research directions and commonly cited works. From there, a snowballing approach was applied:

  - Backward snowballing was used to examine references from key papers to identify foundational studies.
  - Forward snowballing was employed by tracking later works that cited these key papers, ensuring the inclusion of recent advancements in the field.

  This iterative process identified studies on related topics such as cross-docking, multi-compartment vehicle routing, and mathematical formulations. The most relevant studies were then analysed to extract insights applicable to this research.

- A case study: The case study validates the developed model by utilising real-world data from Wolter Koops. The study assesses the efficiency of existing routing decisions by analysing historical transportation data. It evaluates the model's ability to efficiently route a fleet of vehicles in a supply chain with cross-docking. This comparison provides insights into the model's effectiveness in improving routing efficiency and its potential for practical application.

- Expert interviews: Regular meetings were held with the IT manager [5] and the head of planning [4] at Wolter Koops to gain insights into the routing process and validate the research approach. These discussions were conducted in an informal setting, allowing for open dialogue on key operational and strategic considerations. Two business IT consultants from CAPE Groep also participated, bringing valuable input as they were familiar with the research and Wolter Koops' operations. Their involvement helped bridge the gap between theoretical insights and practical constraints. In addition to these meetings, regular discussions were held with the CAPE Groep consultants to refine the research direction further and address emerging questions.

1.5 Deliverables

This study results in the following deliverables:

- Thesis: A comprehensive report documenting the research process, including the methodology, assumptions, limitations, and recommendations for future research or practical implementation.
- Solution Model: The developed model for optimising fleet routing, as described in the thesis and implemented in the software tool.
- Software Tool: A Mendix-based application that implements the solution model, facilitating its practical use. This tool is developed under the supervision of a business IT consultant at CAPE Groep to ensure alignment with industry practices.

1.6 Research Framework

The research framework is the foundation for structuring and guiding this study, providing a clear and logical progression from problem identification to solution evaluation. It outlines the sequence of steps taken throughout the research process and helps to align the methodology with the research objectives. This section presents the research framework, which is structured according to the phases outlined in Section 1.3. Table 1.1 provides an overview of these phases, their corresponding chapters, and the research questions addressed in each. This table defines the layout of the thesis and serves as the basis for its organization.

TABLE 1.1: Research Framework

Phase	**Chapter**	**Research questions**
1. Problem identification	1. Problem Context and Scope Definition	1. What are the current routing and consolidation decision-making processes at Wolter Koops?
	2. Current Vehicle Routing Practices at Wolter Koops	2. What is the planning horizon and how many trips are routed in this time frame?
2. Theoretical foundations	3. Vehicle Routing Problem: Models and Applications	3. What are the key factors influencing routing and consolidation decisions in cross-docking logistics?
		4. How can the vehicle routing problem in a cross-docking supply chain be modelled and optimised?
3. Solution design	4. Problem Formulation and Mathematical Framework	5. What are the key assumptions made in the solution design?
	5. Adaptive TS-SA Hybrid Solution Approach	6. What solution methods exist for enhancing vehicle routing decisions at Wolter Koops, and which is most suitable for implementation?
4. Modelling and implementation	6. Model Validation and Experimental Design	7. How can the input data be processed and validated to ensure the model operates reliably?
	7. Experimental Execution and Computational Results	8. What key parameters influence the model's performance, and how can they be effectively tuned?
		9. How can the model be evaluated in a case study to assess its applicability to real-world routing scenarios?
		10. What experimental setup is required to systematically test the model's performance and robustness?
5. Evaluation	8. Conclusions and Recommendations	11. What are the key findings, implications, and recommendations for practical implementation and further research?

2 Current Vehicle Routing Practices at Wolter Koops

This chapter entails a description of the current situation at Wolter Koops and addresses the following knowledge questions imposed by Phase 1: Problem identification:

1. What are the current routing and consolidation decision-making processes at Wolter Koops?

   1.1 How is the planning department organised?

   1.2 What data and supporting tools are available at Wolter Koops to assist in routing decisions?

2. What is the planning horizon and how many trips are routed in this time frame?

To answer these research questions, Section 2.1 describes the current planning procedure by going into detail on the roles and responsibilities, timing and stages, tools and technology, factors of influence, and the stakeholders involved in the planning process in Sections 2.1.1 - 2.1.4, respectively. Section 2.2 provides more context of the problem by delving into the objectives imposed on the routing of vehicles and the planning performance based on these indicators. Section 2.3 concludes this chapter by summarising the impact of the current situation on problem identification and the solution requirements.

2.1 Planning Procedure

This section describes the current planning procedure for routing vehicles at Wolter Koops by describing how planning groups are organised, which data and tools are used to support decision-making, what factors influence the decision-making process, and which stakeholders can be identified.

2.1.1 Operational Roles and Decision-Making

The planning department aims to devise cost-effective routing plans for all transportation operations. Thus, the planning department at Wolter Koops is organised into specialised planning groups, each responsible for the efficient management of transporting (a set of) specific goods. Planning groups are categorised by geographical location and the type of commodity involved in the planned trips. Because of the incompatibilities of commodities and the geographical locations of goods, eight planning groups exist to plan all operations at Wolter Koops. A planner is responsible for multiple planning groups; trips are specified into one of eight. These eight planning groups are defined in Table 2.1. According to an expert, which commodity type is picked is irrelevant for returning trip planning as incompatibilities rarely occur.

According to insights from an expert, Dutch and German drivers mainly operate within their respective countries. In contrast, Polish drivers are tasked with transporting goods to the United Kingdom (UK), France, Italy, and other destinations on an occasional basis.

Planners responsible for return trips make ad-hoc decisions when vehicles end up empty

TABLE 2.1: Planning groups at Wolter Koops

Planning Group	Description of Routes
ALZ	Round trips from Alzenau (GER)
EG	Fruits, vegetables, plants, and flowers, primarily to GER
ENG	Meat, dairy, and frozen goods across Europe
GP	Retail routes within GER, loaded and delivered domestically
I	Routes from GER to NL or border regions
OW	Various routes from Osterweddingen (GER)
REC	Packaging routes, loaded and delivered at various locations
TRB	Routes for meat, dairy, and frozen goods within NL

at a retailer's location and could potentially pick up goods on the return journey. The decision to pick up an order at a specific location is influenced by factors such as the vehicle's location and availability (considering compulsory breaks), the supplier and retailer locations, and the availability of vehicles at the CD. While these decisions rely on the planners' expertise, no formalised rule or system is in place to guide the process. The primary goal of the return trip planners is to identify opportunities to maximise vehicle utilisation. It is important to note that these return trips are outside the scope of this research, as explained in Section 1.2.2.

Planners are primarily concerned with devising static routing plans. This means that routes are based on known, fixed information about orders and conditions, with no adjustments made during execution. In addition to these planners, dispatchers handle real-time adjustments such as traffic jams, vehicle failure, or other disruptions. Thus, dispatchers dynamically adjust the static routing plans to maximise efficiency.

2.1.2 Operational Data and Supporting Tools

On average, planners handle approximately 1,025 demand requests on Mondays, Tuesdays, and Wednesdays. Demand decreases later in the week, with 850 requests on Thursdays, 775 on Fridays, 650 on Saturdays, and 240 on Sundays. This planning results in a total of around 5,500 trips per week. Although trips are generally planned for the following day, late orders can necessitate adjustments to the existing schedule.

Customer portal
The customer portal functions as an Electronic Data Interchange (EDI) system, facilitating the automated exchange of order information between the customer and Wolter Koops. It allows customers to communicate their order to Wolter Koops and was custom-built by CAPE Groep. These requests include customer specifications such as pick up locations, delivery destinations, time windows, commodity descriptions, order quantities, and temperature requirements. Furthermore, customers can check their order status and review their order history in individualised environments. From this point onwards, we consider suppliers and retailers to be the customers of Wolter Koops.

Transportation Management System (TMS)
The TMS primarily displays the details of freight orders requested by customers via the customer portal. It integrates with the customer portal to automate and streamline the entry of freight orders into the system. Additionally, the TMS streamlines planning by enabling planners to share information regarding the process. For instance, if a vehicle's

trip has been planned but is not fully loaded, it is marked with a corresponding code to facilitate further planning and communication among the team.

Transics on-board computer
The Transics onboard computer offers real-time insight into the operational status of vehicles. It monitors and records driving times, rest periods, and speed and can read and store data from the digital tachograph. It provides real-time tracking to allow planners to monitor the status of vehicles continuously. Drivers receive their routes through the system, while planners receive real-time updates on the progress of the trips. The Transics tool is connected with the TMS through the planned destinations transferred from the TMS. Furthermore, some real-time insights are shared to the TMS.

ETA Service
The ETA service provides insights into the real-time feasibility of ongoing trips, although it is not a standalone tool—hence its name. Vehicle ETAs are calculated at specific trigger points based on the vehicle's status and location. These triggers are set at predetermined intervals or events, such as exact time intervals and departure times. The service is integrated into the TMS, where the calculated ETAs are displayed. CAPE Groep developed this custom-built service.

In summary, planners utilise the TMS and Transics tools to communicate with one another and to access insights into freight orders and real-time vehicle information. Although ETAs calculated by the ETA service are displayed within the TMS, the Transics tool provides a more comprehensive overview of vehicle status. Therefore, planners rely on both tools to obtain a complete view of the operational situation. These tools are further supported by the customer portal and the ETA service, which provides demand requests and up-to-date information on ETAs. The interactions between these tools and the ETA service are depicted in Figure 2.1. Furthermore, it shows how a routing algorithm is placed relative to the existing tools. The routing algorithm takes order information from the TMS and returns a set of routes. It is important to note that the dotted lines represent tools and integrations developed by CAPE to enhance the core functionalities of the TMS and Transics.

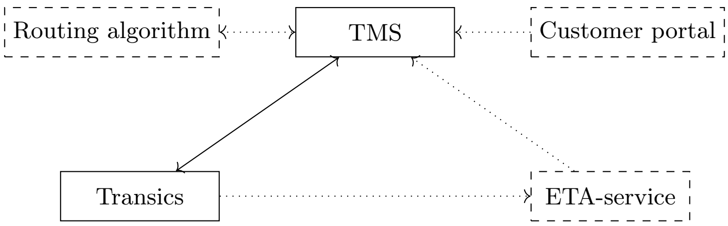


FIGURE 2.1: Tools used in the planning process and the role of the routing algorithm.

2.1.3 Decision Factors in Vehicle Routing

The vehicle routing process at Wolter Koops is shaped by multiple operational, logistical, and regulatory factors, each of which plays a critical role in ensuring efficiency and compliance in transportation planning. These factors were identified through consultations with the head of the planning department at Wolter Koops [4] and are categorised into three primary groups: timing and scheduling constraints, resource constraints, and operational and legal constraints.

Timing Constraints

Suppliers and retailers impose time windows for the pick up and delivery of goods. Adherence to these constraints is essential for maintaining service reliability and operational efficiency. Four key factors influence scheduling decisions: time windows, travel times, loading and unloading times, and rest periods.

Time windows define the permissible periods during which pick ups and deliveries must occur. Travel times are influenced by real-time traffic conditions and estimated using dedicated software tools such as AWS and Google Maps. Loading and unloading times are calculated based on historical data, commodity type, and handling procedures, but actual durations may vary due to operational conditions. In temperature-controlled transport, additional time may be required for pre-cooling before loading or stabilisation upon unloading. Furthermore, rest periods must be incorporated into routing decisions to comply with legally mandated driver breaks and working hour regulations.

Resource Constraints

Resource constraints affect the feasibility of vehicle routing and include vehicle availability, commodity availability, commodity compatibility, and temperature compatibility.

Vehicle availability determines the number of vehicles that may be allocated to a given set of routes at any time. Commodity availability ensures that all required goods are present at the CD when needed to prevent disruptions in distribution. Additionally, some products are subject to compatibility regulations, prohibiting certain commodities from being transported together due to food safety concerns. For instance, dairy products and raw poultry must be kept separate to prevent cross-contamination.

Temperature compatibility further restricts route feasibility, as each vehicle has two configurable compartments, allowing for, at most, two distinct temperature zones per trip. If a route requires goods with conflicting temperature requirements, the trip may not be feasible. Since compartment configurations must be determined before departure, goods requiring three or more distinct temperature conditions cannot be transported simultaneously.

Operational and Legal Constraints

Operational and regulatory factors influence routing feasibility by imposing driver work regulations and temperature-controlled transport laws.

Driver work regulations govern the maximum allowable working hours, required rest periods, and shift allocations, ensuring compliance with labor laws. Additionally, temperature-controlled transport must comply with temperature-controlled transport regulations, such as HACCP, FDA, or EU Food Hygiene Regulations, which specify requirements for storage, handling, and temperature monitoring perishable goods.

Beyond these constraints, routing decisions may also incorporate consolidation strategies at the CD. Depending on factors such as supplier location, CD position, retailer proximity, demand volume, and supplier shipment schedules [1], goods may either be consolidated at the CD or directly shipped from suppliers to retailers.

Balancing all these constraints makes routing decisions a complex challenge, requiring careful trade-offs between efficiency, feasibility, and compliance. To make informed decisions and determine the best possible routes, planners must carefully evaluate these factors and find an approach that optimises logistics.

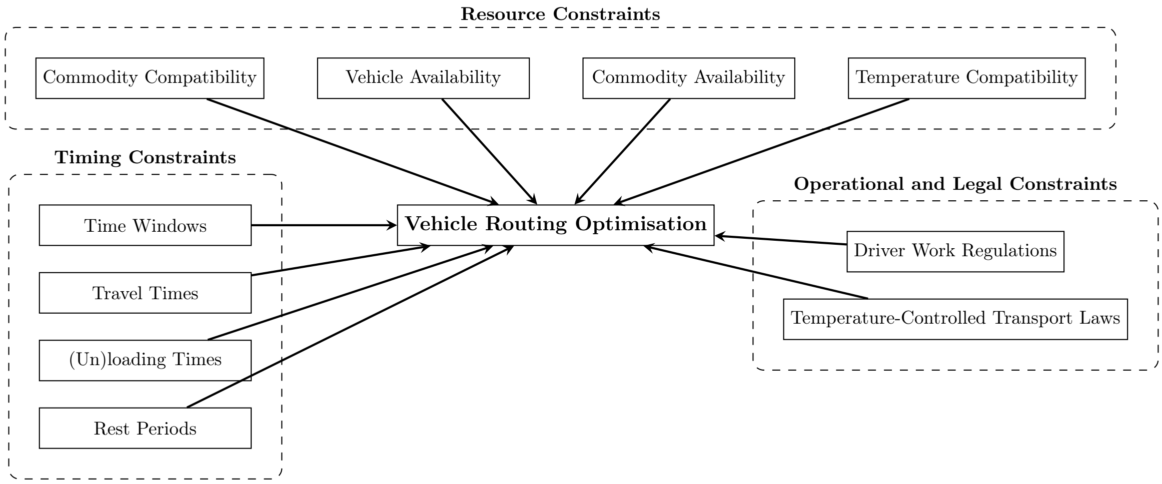


FIGURE 2.2: Factors Influencing Routing Decisions

2.1.4 Stakeholder Analysis

This section examines the stakeholders who are directly involved and affected by the planning decisions made by the planners. The stakeholders are identified through expert interviews, and the role of each stakeholder is explained briefly.

Planners

Planners are responsible for planning vehicle movements, routes, and schedules to ensure efficient operations. They form a small portion of Wolter Koops' personnel, with 26 planners currently active at the company. Currently, routing planning is done manually by planners. As discussed in Section 2.1.1, planners work across multiple planning groups and must continuously communicate with one another, which complicates the planning process.

Dispatchers

Dispatchers play a pivotal role in real-time adjustments to the planning process, as they are responsible for issuing instructions to drivers through the Transics system. Unlike planners, who typically operate in an offline manner by creating routes and schedules in advance, dispatchers continuously monitor ongoing operations and communicate any changes in routing to drivers in real-time. These adjustments are often necessitated by unforeseen events such as traffic congestion, road closures, or vehicle breakdowns. Dispatchers act as problem solvers in this capacity, adapting the routing plan to accommodate unexpected disruptions. Additionally, they are responsible for notifying retailers of any delays that may affect their shipments.

Wolter Koops Management

The management of Wolter Koops plays a crucial role in overseeing logistics operations, ensuring strategic alignment with company goals, and maintaining efficiency in supply chain processes. Their focus lies on optimising transportation, warehousing, and distribution.

Drivers

Drivers are vital to Wolter Koops's daily operations, as they drive planned trips. Dutch and German drivers typically drive set trips and return home after their shifts, whereas Polish drivers often drive greater distances with shifts lasting multiple days or weeks. To this end, it is common for them to drive in pairs so that breaks do not affect whether a vehicle is driven or not.

Retailers

Retailers request one or multiple commodities through Wolter Koops' customer portal. In each order, various trips with different characteristics, such as time windows, temperature, and quantities, may be requested. Retailers are primarily wholesalers and distributors of perishable goods, including seafood, meats, dairy products, and flowers [3]. The retailer is responsible for immediately handling the goods upon delivery, allowing the vehicle to continue its trips as soon as possible. Additionally, they are expected to make reasonable requests to ensure it is feasible to meet their time windows. Wolter Koops strives to meet their demands on time and does not refuse customers.

Suppliers

Suppliers provide the goods that Wolter Koops transports to retailers, primarily perishable products like produce, meat, and dairy. They are responsible for ensuring that products are available for pick up within agreed-upon time windows, as delays can disrupt planning and affect vehicle utilisation. Suppliers must also ensure that goods are packaged and prepared for transport on time, especially for temperature-sensitive items, to maintain product quality during delivery.

Digital Freight Exchange Platforms

Digital freight exchange platforms are online marketplaces that connect third-party transportation service providers with shippers seeking freight transport services. These platforms enable transporters to access and bid on available freight orders posted by various shippers, thereby optimising logistics operations. At Wolter Koops, returning trip planners utilise these platforms to identify opportunities outside of their customers to fill return trips with freight, thereby increasing vehicle utilisation.

Third-party Logistics Service Providers

Third-party logistics service providers enhance Wolter Koops' operational capacity, ensuring flexibility during periods of high demand. Effective collaboration between Wolter Koops and these providers is crucial for maintaining service quality and operational efficiency when internal capacity is constrained.

2.2 Business Objectives and Current Performance

Currently, Wolter Koops evaluates its performance based on KPIs such as the on-time delivery rate, load efficiency, and the idle times for customers. These include:

- *Costs* encompass both fixed and variable transportation expenses, including vehicle usage, fuel consumption, and other operational costs such as driver wages and vehicle maintenance.
- *On-time service rates* refer to the ratio of shipments that are picked up or delivered within the specified time interval.
- *Load efficiencies* refers to the extent to which the load capacity of vehicles is utilised.
- *Carbon emissions* refer to the amount of carbon dioxide emitted in total or per travelled kilometre. Reporting on this KPI has become mandatory following the introduction of the ISO-14083 standard

The overall performance of the current manual planning procedure is not fully known, but a sample of historical routes provides a reference for evaluating the model's performance. KPIs include costs, loading efficiency, and on-time service rates. The sample indicates an average loading efficiency of 70.7%, with estimated fixed and variable costs totalling €32,199. The on-time service rate cannot be precisely determined, as actual arrival times were not recorded. However, using the same travel times as the model, 236 out of 284 supplier/retailer location visits would have been considered on time (83.0%). Additionally, carbon emissions are implicitly accounted for through travelled minutes and loading efficiency but are not explicitly analysed further in this study.

2.3 Findings and Implications

1. *What are the current routing and consolidation decision-making processes at Wolter Koops?*

This chapter describes the existing planning procedures at Wolter Koops from a high perspective, identifying key operational characteristics that influence routing decisions. Key insights from this analysis are:

- The routing process is managed by eight specialised planning groups, each overseeing specific goods and regions. Planners create static routes based on historical demand patterns, with real-time adjustments handled separately by dispatchers.
- While digital systems are available, routing decisions remain manual and experience-driven rather than automated. No structured rule-based system is currently in place to route vehicles.

These findings indicate that routing decisions lack automation and rely heavily on human expertise, which may limit scalability and consistency. The absence of a formalised decision-support system suggests that optimisation techniques could improve planning efficiency.

2. *What is the planning horizon and how many trips are routed in this time frame?*

This chapter provides insights into the time horizon and volume of routing decisions at Wolter Koops:

- Routing is performed daily for the following day, with around 5,500 trips scheduled per week.
- The number of trips varies significantly throughout the week, with higher demand early in the week and a decline toward the weekend.

These findings show that multiple planning groups exist to divide the planning operations. Given that Wolter Koops schedules 5,500 trips weekly, this research is scoped to focus on one planning department. As a result, we concentrate on data from the Europese Gemeenschap uitgezonderd vlees en zuivel (EG) department. Additionally, the factors influencing routing decisions highlight the complexity of vehicle routing in a cross-docking environment, and emphasizes the need for standardising the routing process.

3 Vehicle Routing Problem: Models and Applications

This chapter explores relevant research and existing knowledge on routing decisions, focusing on algorithmic methods that could address the problem outlined in the previous chapters. It begins with a general overview of routing decisions based on the literature, followed by a discussion of problem variations that share characteristics similar to those of the case presented. Finally, several recent studies are reviewed to provide insights into comparable applications and a deeper understanding of the problem's complexities. Therefore, the goal of this chapter is to answer the following research questions of the second phase of the research framework (see Section 1.3):

3. *What are the key factors influencing routing and consolidation decisions in cross-docking logistics?*
4. *How can the vehicle routing problem in a cross-docking supply chain be modelled and optimised?*

Section 3.1 introduces the problem and explores its variations as defined by existing literature. Sections 3.2 - 3.5 present definitions and models from the literature that explore the characteristics and variations of the problem, developed over decades of extensive research on vehicle routing optimisation. Section 3.6 compiles relevant implementations of the problem that closely align with the presented case and discusses how the study addresses the research gaps identified in this chapter. Section 3.7 concludes this chapter by revisiting the research questions.

3.1 Introduction to the Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was first introduced by Dantzig and Ramser [12] as the "vehicle Dispatching Problem", a generalisation of the Traveling Salesman Problem (TSP) presented by Flood [13] aiming to find the "optimum routing of a fleet of gasoline delivery vehicles between a bulk terminal and a large number of service stations supplied by the terminal." Due to their practical relevance and computational complexity, this instigated decades of research on the VRP with numerous collaborations between businesses and academia [14, 15]. This is illustrated by the numerous taxonomies and surveys that have appeared in the last decades devoted to the VRP [16–25].

The VRP aims to find a set of delivery routes where (1) each customer is known in advance and visited exactly once, (2) all vehicle routes start and end at the depot, and (3) some side constraints are satisfied [26]. It represents an essential class of combinatorial optimisation problems, where customers are served by several vehicles while satisfying some constraints [2, 27]. In most cases, the objective remains the same and minimises total distribution costs, while distribution services remain at a high level [28].

A generic definition of the VRP is given in Toth and Vigo [29]:

Given: A set of transportation requests and a fleet of vehicles.

The problem is then to find a plan for the following:

> **Task:** Determine a set of vehicle routes to perform all (or some) transportation requests with the given vehicle fleet at minimum cost; in particular, decide which vehicle handles which requests in which sequence so that all vehicle routes can be feasibly executed.

As an extension of the travelling salesman problem, the VRP is considered both combinatorial and $\mathcal{NP}$-hard, indicating its significant computational complexity [30–32]. This complexity arises from (1) the exponential growth in the number of possible solutions as the size of the instances increases, and (2) the exponential increase in the number of conceivable problem variants due to the variety of problem attributes (i.e. constraints, decision sets, and objectives) arising from real applications [33]. Exact optimisation methods can only solve small instances of the problem VRP in reasonable computation times, but to describe specific logistical settings, the number of variants of the problem has exploded. Fortunately, advances in computer technologies have increased academic attention and the possibilities for solving and implementing new variants and solution methods. This is illustrated in Figure 3.1, which depicts the VRP family defined by its various introduced and adapted variants.

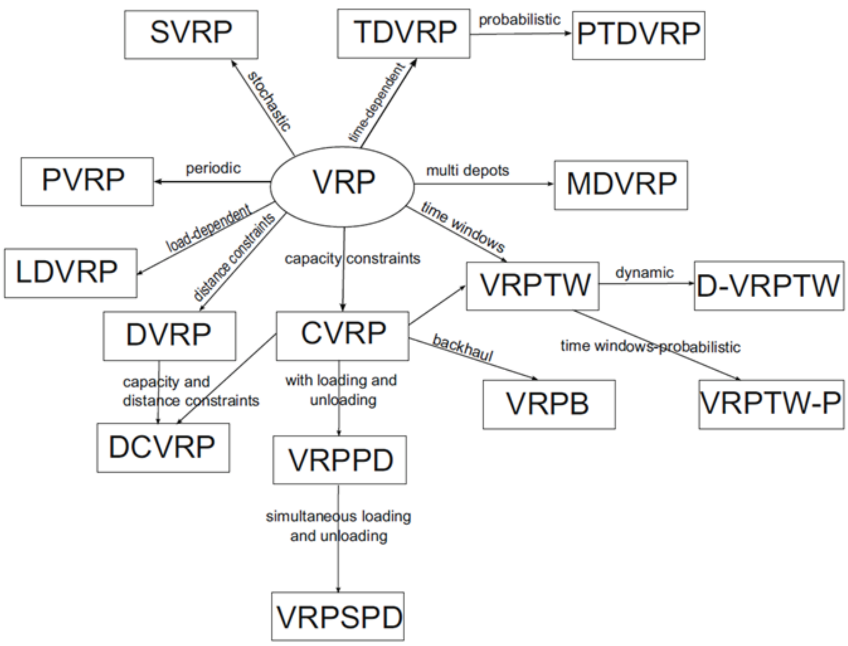


FIGURE 3.1: The VRP family hierarchy [2]

Figure 3.1 provides a structured overview of key VRP extensions, serving as a valuable reference for categorising problem adaptations. However, it represents only a subset of the vast landscape of its variants, as many additional modifications exist beyond this hierarchy. This is also evident by the numerous taxonomies proposed to classify VRP instances according to their characteristics [21, 22, 34, 35].

3.2 VRP with Capacity Constraints

The most elementary form of the VRP considered in the literature is the Capacitated VRP (CVRP). The CVRP consists of finding routes for vehicles of known capacity. These vehicles originate from a single depot and are tasked with servicing a predefined set of customers at known positions, each with specified demands. As the name suggests, in this class of the VRP, the total demand on each route must not exceed the vehicle capacity [36]. The CVRP can be formulated as either a symmetrical (undirected) variant, where travel costs between locations are the same in both directions, or an asymmetrical (directed) variant, where travel costs differ depending on direction. This chapter bases its formulations on the asymmetrical variant, but it can be easily adapted to consider the symmetrical case.

The capacity constraint generally exists as a fundamental constraint in VRPs, meaning that various solution methods are available for solving the CVRP. According to Zhang et al. [26], precise and heuristic algorithms are available for solving the CVRP. Efforts to address real-world applications of the CVRP often rely on heuristic methods, as highlighted by Dorigo et al. [37]. These methods, sometimes combined, have effectively addressed the CVRP. Further details on these solution methods are discussed in Section 5.1.

In the CVRP, the transportation of goods involves a single depot, denoted by 0. A set of other points, typically referred to as customers, is denoted by $\mathcal{N}_c = \{1, 2, \ldots, n\}$. Let the set of all nodes (or nodes), including the depot, be $\mathcal{N} = \{0\} \cup \mathcal{N}_c$, and the set of arcs be $\mathcal{A}$. Set $\delta^+(i)$ represents the set of arcs leaving node i in a graph, while $\delta^-(i)$ represents the set of arcs entering node i. That is, the set of successors and predecessors, respectively.

If at least one pair of nodes $i, j \in \mathcal{N}$ has asymmetric costs $c_{ij} \neq c_{ji}$, then the underlying graph is a complete digraph $G = (\mathcal{N}, \mathcal{A})$ with set of arcs $\mathcal{A} = \{(i, j) \in \mathcal{N} \times \mathcal{N} : i \neq j\}$ and arc costs c_{ij} for $(i, j) \in \mathcal{A}$ [29]. Thus, the digraph $G(\mathcal{N}, \mathcal{A})$ depicts all nodes (locations) and all connecting arcs (routes) of the directed VRP.

Moreover, each node has a demand q_i, such that $q_i > 0$ for each $i \in \mathcal{N}_c$ and $q_0 = 0$ (since the node 0 represents the depot location). Additionally, Q represents the capacity of the vehicles (assuming a homogeneous fleet [3]). For an arbitrary customer subset $\mathcal{S} \subseteq \mathcal{N}$, it is convenient to let $r(\mathcal{S})$ depict the minimum number of vehicle routes needed to serve $\mathcal{S}$. In the CVRP, the number $r(\mathcal{S})$ can be computed by solving a bin packing problem with items $\mathcal{N}$ of weight q_i, $i \in \mathcal{N}$, and bins of size Q (see Martello and Toth [39]). A lower bound, often used instead of $r(\mathcal{S})$, is given by $\lceil q(\mathcal{S})/Q \rceil$. Note that the short notation $q(\mathcal{S})$ is used for $\sum_{i \in \mathcal{S}} q_i$.

Using graph $G(\mathcal{N}, \mathcal{A})$ with $\mathcal{N}$ the set of nodes (i.e. the depot and the customers) and $\mathcal{A}$ the set of arcs that connect the nodes, a variety of integer programming models were proposed to depict the CVRP and all of its variants. In particular, the models differ in the chosen set of decision variables [31, 40]. Given the definitions of Rieck and Zimmermann [14], the four integer-programming modelling techniques that can be distinguished are the following:

- *Two-index vehicle flow formulations* containing binary variables indicating whether an arc in the underlying graph G is selected or not.
- *Three-index vehicle flow formulations* explicitly indicate the vehicle that traverses an arc. Therefore, they consider a binary variable for every arc-vehicle combination.
- *Commodity-flow formulations* require a new set of (continuous) variables representing

[3] A uniform fleet composition with identical characteristics, such as capacity, speed, and operational cost [38], with $|\mathcal{V}|$ vehicles where $\mathcal{V}$ represents the set of all vehicles in the fleet. The opposite is referred to as a heterogeneous fleet.

the amount of demand that flows along the associated arcs. This is in addition to the variables used by a two- or three-index vehicle flow formulation.

- *Set-partitioning formulations* containing a binary variable for every potential vehicle route.

All formulations have advantages and disadvantages regarding practical relevance, flexibility, and computation times. Still, the three-index model is adopted in the following since many modifications and extensions can be described conveniently in that case. For more information on the distinct formulations, the reader is referred to Appendix A and the book of Toth and Vigo [29].

Three-index vehicle flow formulation

Next, a three-index formulation is given, which is based on a directed graph $G = (\mathcal{N}, \mathcal{A})$ in which the depot is depicted by two nodes o and d representing the origin and destination of a route. By explicitly modelling o as the starting point and d as the ending point of routes, the formulation supports the inclusion of additional constraints or features that may require separate definitions for starting and ending locations.

This is because, unlike the two-index formulation, the three-index formulation explicitly models routes, making it convenient to add the definition of distinct start and end points [29]. The new definition of the node and arc sets is:

$$\mathcal{N} = \mathcal{N}_c \cup \{o, d\} \;\text{ and }\; \mathcal{A} = (\mathcal{N} \setminus \{d\} \times \mathcal{N} \setminus \{o\})$$

Let x_{ij}^v be defined as a binary variable that assumes value 1 if and only if there is a route that is traversed by some vehicle $v \in \mathcal{V}$, going from customer i to j directly, for $(i, j) \in \mathcal{A}$. In addition, binary variables y_i^v indicate whether or not the vehicle v visits the node $i \in \mathcal{N}$. Finally, u_i is an integer variable associated with customer i and the corresponding inequalities serve to eliminate tours that do not begin and end at a depot [41]. The variables u_i $(\forall i \in \mathcal{N}_c)$ represent the total demand of nodes on the route until node i (including node i) [41, 42]. The three-index vehicle flow formulation is then defined by the integer linear programming formulation:

$$\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^v && && \text{(1a)}\\
\text{subject to} \quad & \sum_{v \in \mathcal{V}} y_i^v = 1 && \forall\, i \in \mathcal{N}_c && \text{(1b)}\\
& x_v\left(\delta^+(i)\right) - x_v\left(\delta^-(i)\right) = \begin{cases} 1, & i = o \\ 0, & i \in \mathcal{N}_c \end{cases} && \forall\, i \in \mathcal{N} \setminus \{d\}, \quad v \in \mathcal{V} && \text{(1c)}\\
& y_i^v = x_v\left(\delta^+(i)\right) && \forall\, i \in \mathcal{N} \setminus \{d\}, \quad v \in \mathcal{V} && \text{(1d)}\\
& y_d^v = x_v\left(\delta^-(d)\right) && \forall\, v \in \mathcal{V} && \text{(1e)}\\
& u_i^v - u_j^v + Q x_{ij}^v \leq Q - q_j && \forall\, (i,j) \in \mathcal{A}, \quad v \in \mathcal{V} && \text{(1f)}\\
& q_i \leq u_i^v \leq Q && \forall\, i \in \mathcal{N}, \quad v \in \mathcal{V} && \text{(1g)}\\
& x = (x_v) \in \{0,1\}^{\mathcal{V} \times \mathcal{A}} && && \text{(1h)}\\
& y = (y_v) \in \{0,1\}^{\mathcal{V} \times \mathcal{N}} && && \text{(1i)}
\end{aligned}$$

In the three-index vehicle flow formulation, the x_{ij}^v variables are differentiated (i.e. they are not aggregated into a single variable x_{ij}) [43]. Furthermore, this section uses a slightly modified notation, as presented in Toth and Vigo [29], compared to the original formulation proposed by Golden et al. [44]. The formulation given above again starts with equation (1a), which represents the adapted objective function that aims to minimise the total travel costs of the routes, considering the decisions made per vehicle $v \in \mathcal{V}$. Equations (1b) ensure that each customer is served exactly once. Equations (1c) ensure that the starting point o is only left and not entered, the customers are both entered and left, and the endpoint d is only entered and not left. The latter holds because these equations imply that all vehicles return to the end point (i.e. $x_v(\delta^+(d)) - x_v(\delta^-(d)) = -1$), which ensures that each route forms a valid $o - d$ path. Equations (1d) and (1e) ensure that the binary variable y_i^v assumes 1 if and only if a customer i is served by a vehicle v, by coupling it with the binary variable x_{ij}^v. Moreover, equations (1f) and (1g) respectively represent an extended vehicle-specific Miller-Tucker-Zemlin (MTZ) and capacity constraints [45, 46]. Equations (1h) and (1i) are the integrality conditions on the decision variables.

Since the three-index vehicle flow formulation of the CVRP forms a basis for the notation in other variants presented in this chapter, its basic notation is summarised in Table 3.1.

TABLE 3.1: Three-index vehicle flow notations (CVRP).

Category	*Symbol*	*Description*
	$\mathcal{N}$	Set of nodes, where o (origin) and d (destination) represent the depot, and $\mathcal{N}_c$ is the set of customers.
	$\mathcal{A}$	Set of arcs, where $\mathcal{A} = (\mathcal{N} \setminus \{d\} \times \mathcal{N} \setminus \{o\})$.
Sets	$\mathcal{V}$	Set of vehicles $\{1, \ldots, V\}$.
	$\delta^+(i)$	Set of successors of i in graph $G = (\mathcal{N}, \mathcal{A})$.
	$\delta^-(i)$	Set of predecessors of i in graph $G = (\mathcal{N}, \mathcal{A})$.
	c_{ij}	Cost of traversing arc $(i, j) \in \mathcal{A}$.
Parameters	q_i	Demand at node $i \in \mathcal{N}$, with $q_0 = 0$.
	Q	Capacity of vehicles.
	x_{ij}^v	Binary variable: 1 if vehicle $v \in \mathcal{V}$ traverses arc $(i, j) \in \mathcal{A}$, 0 otherwise.
Variables	y_i^v	Binary variable: 1 if vehicle $v \in \mathcal{V}$ visits customer $i \in \mathcal{N}$, 0 otherwise.

3.3 VRP with Time Windows

The VRP with Time Windows (VRPTW) extends the CVRP by introducing time windows associated with each customer, specifying that service must be completed within these designated time intervals. As an extension of the CVRP, the VRPTW is $\mathcal{NP}$-hard. In addition to addressing the issues discussed in Section 3.2, there is added complexity in adhering to allowable delivery times. Routes must be designed to ensure that each customer is visited by exactly one vehicle within a given time interval, that all routes start and end at the depot, and that the total demands of all points on a particular route must not exceed the capacity of the vehicle [47]. Hence, both the spatial and temporal aspects of the routing problem must be carefully considered [48]. With a realistic number of customers, the interaction between these aspects often leads to optimal routes that deviate significantly from traditional patterns, making manual planning exceptionally challenging. In this case, computerised methods have demonstrated superiority over manual planning [29].

The time windows in the VRPTW can be classified as hard or soft. Hard time windows in vehicle routing specify strict arrival times where a vehicle can only service a customer within the associated time window; for instance, arriving early requires waiting until the window starts. In contrast, soft time windows allow flexibility, permitting arrivals outside the window with a penalty cost. This chapter presents the mathematical formulation of the VRPTW, detailing the mixed integer linear programming formulation for the version with hard time windows and the necessary adaptations for implementing soft time windows. From this point onward, the difference between the VRP with Hard Time Windows (VRPHTW) and the VRP with Soft Time Windows (VRPSTW) will be clarified whenever necessary.

Refer to Section 3.2 for the definitions, and let the graph $G(\mathcal{N}, \mathcal{A})$ be defined accordingly. The time windows associated with a customer i are defined by interval $[e_i, l_i]$, where the service time is equal to s_i. Note that $[a_o, b_o] = [a_d, b_d]$, where a_o and d_o represent the earliest possible departure time from the depot and the latest possible arrival time at the depot, respectively. The travel time along arc (i, j) is equal to t_{ij}. Moreover, if vehicles are allowed to remain at the depot (i.e. when minimising the number of vehicles), it is necessary to add the arc (o, d) with $c_{od} = t_{od} = 0$ to the arc set $\mathcal{A}$. The fleet size V is typically a decision variable in this case [49].

For the VRPTW, it is necessary to define two types of variables: For each arc $(i, j) \in \mathcal{A}$ and each vehicle $v \in \mathcal{V}$, there is a binary arc-flow variable x_{ij}^v that is equal to 1 if arc (i, j) is used by vehicle v, and 0 otherwise. Furthermore, for each node $i \in \mathcal{N}$ and vehicle $v \in \mathcal{V}$, there is a time variable T_i^v that specifies the start of service time at node i when serviced by vehicle v. The VRPHTW can be described as in the following mixed integer linear programming formulation [29]:

$$
\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}^v && && \text{(2a)}\\
\text{subject to} \quad & \sum_{v\in\mathcal{V}} \sum_{j\in\delta^+(i)} x_{ij}^v = 1 && \forall\, i \in \mathcal{N}_c && \text{(2b)}\\
& \sum_{j\in\delta^+(o)} x_{oj}^v = 1 && \forall\, v \in \mathcal{V} && \text{(2c)}\\
& \sum_{i\in\delta^-(j)} x_{ij}^v - \sum_{i\in\delta^+(j)} x_{ji}^v = 0 && \forall\, v \in \mathcal{V}, \quad j \in \mathcal{N}_c && \text{(2d)}\\
& \sum_{i\in\delta^-(d)} x_{id}^v = 1 && \forall\, v \in \mathcal{V} && \text{(2e)}\\
& x_{ij}^v (T_i^v + s_i + t_{ij} - T_j^v) \le 0 && \forall\, v \in \mathcal{V}, \quad (i,j) \in \mathcal{A} && \text{(2f)}\\
& e_i \le T_i^v \le l_i && \forall\, v \in \mathcal{V}, \quad i \in \mathcal{N} && \text{(2g)}\\
& \sum_{i\in\mathcal{N}_c} q_i \sum_{j\in\delta^+(i)} x_{ij}^v \le Q && \forall\, v \in \mathcal{V} && \text{(2h)}\\
& x_{ij}^v \in \{0, 1\} && \forall\, v \in \mathcal{V}, \quad (i,j) \in \mathcal{A} && \text{(2i)}
\end{aligned}
$$

The objective function (2a) aims to minimise the total cost of serving all customers. Equations (2b) ensure that every customer is served exactly once. The restrictions (2c) - (2e) ensure that each vehicle v is assigned to an $o - d$ path. Constraints (2f) ensure that if a vehicle v travels from customer i to customer j, the start of service (defined by the decision variable T_j^v) at customer j may only occur after vehicle v has completed its service at customer i and travelled from customer i to customer j. Equations (2g) and (2h) guarantee schedule feasibility concerning time windows and vehicle capacity, respectively. Finally, equations (2i) are the integrality conditions on the decision variable x_{ij}^v.

The model defined by (2a)-(2i) is nonlinear due to constraints (2f) that can, however, be linearised as shown by (2f') below [29, 48, 50]:

$$T_i^v + s_i + t_{ij} - T_j^v \leq \left(1 - x_{ij}^v\right) M_{ij} \tag{2f'}$$

Here, M_{ij} with $(i,j) \in \mathcal{A}$ represent large constants that effectively deactivate the constraint when the corresponding arc $(i,j) \in \mathcal{A}$ is not used by the vehicle (thus not affecting the feasibility of the model). As suggested by Toth and Vigo [29], these can be set to $\max\{l_i + s_i + t_{ij} - e_j, 0\}$.

As a generalisation of the VRPHTW, the VRPSTW can easily be derived from the mixed integer linear programming formulation described earlier in this section. To this end, the following four steps are needed [49, 51]:

1. First, introduce the decision variables and quantify how much a vehicle arrives outside the time window interval $[e_i, l_i]$. In this case, E_i represents units of time for arriving early, and L_i denotes units of time for arriving late.
2. Second, define two parameters representing the unit penalty costs of vehicles arriving earlier or later than the earliest or latest specified time. In this case, γ and θ represent these penalty costs, respectively.
3. Third, alter the objective function to include the penalty costs for arriving early or late:
$$\sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^v + \sum_{i \in \mathcal{N}_c} (\gamma E_i + \theta L_i) \tag{3a}$$
4. Last, modify the time window constraints (2g) to account for penalties when arriving outside of the time window, making the constraint non-binding:
$$e_i \leq T_i^v + E_i \leq l_i + L_i \tag{3b}$$

Note that the unit penalty costs for vehicles arriving earlier or later than the earliest or latest specified time may be specific to customers. This can be modelled by simply adding the index i to these costs, such that γ_i and θ_i represent these customer-specific costs and substituting them into the objective function (3a) after.

3.4 VRP with Multiple Commodities

In practical applications, various scenarios require explicit modeling of multiple commodities due to their distinct characteristics and delivery requirements. These routing problems belong to the family of the Multi-Commodity VRP (MCVRP). In the case where different commodities can be aggregated or separated and normalized into vehicle capacity units, explicit modeling of the distinct commodities is unnecessary; they are then implicitly modeled by solving the classical CVRP (multiple times). Aggregation and separation is possible in the following situations:

1. *Aggregation* of the different customer demands is possible when all commodities must be delivered to each customer at once with a single vehicle. In this case, the demands for various commodities can be combined into a single demand for each customer. When this aggregation occurs, the problem simplifies, and the model becomes the classical CVRP [52].

2. *Separation* is possible by if a dedicated fleet must deliver commodities and each commodity is associated with a set of vehicles. The model can then be decomposed by vehicle, i.e. by commodity. For each vehicle (commodity), the corresponding problem is the classical CVRP [52].

From this point onward, it assumed that the different commodities must be modelled explicitly because otherwise, the solution would prove infeasible or suboptimal. In this case, commodities are compatible or incompatible. Commodities are considered incompatible if they cannot be transported simultaneously in the same vehicle or the same compartment when a vehicle has multiple compartments. Otherwise, they are said to be compatible [52]. When the goods are incompatible, the same vehicle can transport different goods only if multiple compartments or other trips separate them. These situations correspond to the class of Multi-Compartment VRP (MCmpt-VRP) [53] and Multi-Trip VRP (MTVRP) [54], respectively. Compatible goods may need to be explicitly modelled when they have different origins or destinations or because of other characteristics [52].

Below are the mathematical definitions and notations used in the subsequent integer linear programming formulation on the CVRP with multiple commodities. This formulation represents the case where different vehicles can deliver commodities to a customer, provided a single commodity is delivered at once by a single vehicle. Furthermore, the commodities are compatible and can thus be transported in the same vehicle. To present the multi-commodity VRP and its variants conveniently and efficiently, the changes required to reformulate to the MCmpt-VRP and the MTVRP are specified after. These formulations are based on the ones in the review on the MCVRP by Gu et al. [52].

Let the complete directed graph $G(\mathcal{N}, \mathcal{A})$ define the set of nodes $\mathcal{N} = \{0, 1, \ldots, N\}$, representing all locations in the network, with 0 being the depot and $\mathcal{N}_c = \mathcal{N} \setminus \{0\}$ the set of customers, and the set of arcs that connect these locations $\mathcal{A} = \{(i,j), i, j \in \mathcal{N}, i \neq j\}$. As new definitions to accommodate the multiple commodities in the formulation, let $m \in \mathcal{M}$ denote a specific commodity where $\mathcal{M}$ represents the set of all commodities. Consequently, the demand for each customer must be expanded to include the commodity index m, so that q_i^m specifies the demand for commodity m by customer i. Finally, the decision variable y_i^{vm} is generalised to indicate whether vehicle v visits node $i \in \mathcal{N}$ while transporting the commodity m.

In the first case presented in this section, commodities are assumed to be compatible and can thus be transported together in the same vehicle. Accordingly, the integer linear programming formulation of the multi-commodity variant with exclusively compatible goods becomes [52]:

$$\min \quad \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^v \tag{4a}$$

$$\text{subject to} \quad \sum_{j \in \delta^-(i)} x_{ij}^v = \sum_{j \in \delta^+(i)} x_{ij}^v \qquad \forall\, i \in \mathcal{N}_c,\ v \in \mathcal{V} \tag{4b}$$

$$\sum_{j \in \delta^+(0)} x_{0j}^v = 1 \qquad \forall\, v \in \mathcal{V} \tag{4c}$$

$$\sum_{i,j \in S} x_{ij}^v \leq |S| - 1 \qquad \forall\, v \in \mathcal{V},\ S \subset \mathcal{N}_c,\ |S| \geq 2 \tag{4d}$$

$$\sum_{v \in \mathcal{V}} y_i^{vm} = 1 \qquad \forall\, i \in \mathcal{N}_c,\ m \in \mathcal{M} \tag{4e}$$

$$\sum_{m \in \mathcal{M}} y_i^{vm} \leq |\mathcal{M}| \sum_{j \in \delta^+(i)} x_{ij}^v \qquad \forall\, i \in \mathcal{N}_c,\ v \in \mathcal{V} \tag{4f}$$

$$\sum_{i \in \mathcal{N}_c} \sum_{m \in \mathcal{M}} d_i^m y_i^{vm} \leq Q \qquad \forall\, v \in \mathcal{V} \tag{4g}$$

$$x_{ij}^v \in \{0,1\} \qquad \forall\, i,j \in \mathcal{N},\ v \in \mathcal{V} \tag{4h}$$

$$y_i^{vm} \in \{0,1\} \qquad \forall\, i \in \mathcal{N}_c,\ v \in \mathcal{V},\ m \in \mathcal{M} \tag{4i}$$

In this formulation, the objective function in (4a) aims to minimise the total travel costs while serving the demands of all customers $\mathcal{N}_c$. Constraints (4b) guarantee that all customers are served exactly once by ensuring that all vehicles that enter a location must also leave that location. Constraints (4c) ensure that all vehicles are used exactly once by stating that the number of outgoing vehicles is equal to the number of vehicles V. In the case where more vehicles are available than needed (i.e. $K > r(\mathcal{N})$), the equalities can be replaced with inequalities of type "$\leq$" (although it is important to note that fleet size minimisation and routing costs are conflicting objective [29]). Constraints (4d) are subtour elimination constraints [55]. Constraints (4e) guarantee that each customer's required commodity is delivered exactly once by a single vehicle. Constraints (4f) ensure that if vehicle v visits customer i, the total number of commodities m that vehicle v delivers to customer i is at most M. This ensures that a vehicle can potentially deliver M different commodities to a single customer, but not more. The restrictions (4g) are imposed to satisfy the capacity of the vehicle, and (4h) and (4i) define the domain of the decision variables [52].

Multi-Compartment VRP (MCmpt-VRP)

The MCmpt-VRP describes the case where multiple incompatible commodities can be transported by separating them by compartments in a vehicle. Many contributions were made to the case where vehicles have multiple compartments were made in recent years. For brevity, however, only the case where commodities have dedicated compartments [56] and the case where commodities may be loaded in several compartments are reviewed in this section [53].

In case vehicles have dedicated compartments with a given capacity, the formulation can be altered to associate specific compartments with commodities, as proposed in Fallahi et al. [56]. In other words, for each vehicle $v \in \mathcal{V}$, the compartment for commodity $m \in \mathcal{M}$ has the commodity-specific capacity of Q^m associated with it. The following constraints are imposed to consider the case in which commodities have dedicated compartments:

$$\sum_{i \in \mathcal{N}_c} d_i^m y_i^{vm} \leq Q^m \qquad \forall\, m \in \mathcal{M},\ v \in \mathcal{V} \tag{5}$$

Thus, when dedicated compartments are available, equations (4g) need to be replaced with equations (5). This imposes that the capacity restrictions are satisfied per compartment associated with commodity m.

Alternatively, the more generic case of the MCmpt-VRP is where the incompatibilities of the commodities are modelled. This case, proposed by Derigs et al. [53], entails vehicles having a set of $\mathcal{L}$ compartments and each compartment $l \in \mathcal{L}$ is associated with a capacity Q^l. Contrary to Fallahi et al. [56], commodities do not have dedicated compartments but can be loaded into several compartments. The model incorporates two types of incompatibilities:

1. between commodities m and compartments l, indicated by the set:

   $$\mathcal{I}_{comp} = \{(m, l) \mid m \in \mathcal{M}, l \in \mathcal{L}, \text{commodity } m \text{ is incompatible with compartment } l\}$$

   and;

2. among different commodities m and m', indicated by the set:

   $$\mathcal{I}_{comm} = \{\left(m, m'\right) \mid m, m' \in \mathcal{M}, m \neq m', \text{ commodities } m \text{ and } m' \text{ are incompatible}\}$$

Here, binary variables z_i^{vml} are introduced that take a value of one if the commodity m of customer i is loaded in compartment l of vehicle v, zero otherwise. These replace the variables y_i^{vm} as they are equal when taking the sum over the compartments $l \in \mathcal{L}$ of the z_i^{vml} variables. Compared to the formulation of the MCVRP presented at the beginning of this section, and in addition to replacing variables y_i^{vm} with $\sum_{l \in \mathcal{L}} z_i^{vml}$, the following constraints are imposed which are related to compartment capacities and incompatibilities [52]:

$$\sum_{i \in \mathcal{N}_c} \sum_{m \in \mathcal{M}} d_i^m z_i^{vml} \leq Q^l \qquad \forall\, v \in \mathcal{V},\ l \in \mathcal{L} \tag{6a}$$

$$z_i^{vml} + z_i^{vm'l} \leq 1 \qquad \forall\, i, j \in \mathcal{N}_c,\ v \in \mathcal{V},\ l \in \mathcal{L},\ (m, m') \in \mathcal{I}_{comm} \tag{6b}$$

$$z_i^{vml} = 0 \qquad \forall\, i \in \mathcal{N}_c,\ v \in \mathcal{V},\ (m, l) \in \mathcal{I}_{comp} \tag{6c}$$

In the context of the formulation (4a) - (4i), equations (6a) replace equations (4g), and equations (6b) and (6c) are added to include the multiple compartments without dedicated compartments. Equations (6a) are similar to equations (5), ensuring that the capacity per compartment $l \in \mathcal{L}$ is satisfied. Equation (6b) ensures that incompatible commodities m and m' are not loaded into the same compartment l of vehicle v. Equations (6c) ensure that commodities m are not loaded into incompatible compartments l within vehicles V.

3.5 VRP with Cross-Docking

Cross-docking is a logistics strategy that reduces storage time by transferring goods directly from inbound shipments to outbound transportation at an intermediate facility. Unlike traditional distribution centres, cross-docking terminals prioritize immediate consolidation and dispatch of goods, lowering storage costs and improving supply chain efficiency [1].

As cross-docking has seen widespread adoption in logistics over the past decades, vehicle routing in such supply chains has received increasing attention [1]. This is particularly relevant for Wolter Koops, which employs cross-docking to consolidate shipments and optimise routing decisions. This section examines the VRP with Cross-Docking (VRPCD), presenting a mathematical formulation that accounts for its operational constraints. A review of relevant literature and problem variations follows in Section 3.6.

In the first definition of the VRPCD, presented by Lee et al. [57], simultaneous arrival and departure times of vehicles at the CD were required to reduce the waiting time of each vehicle. Furthermore, for the consolidation process, products that arrive at the CD are classified and loaded according to destination. The objective is to determine a set of vehicle routes starting and terminating at the depot that minimises total transportation cost [58].

One may think that the complexity of the VRPCD is similar to solving to separate instances of the VRP (i.e. one for the delivery routes and one for the pick up routes), but the introduction of cross-docking increases computational complexity by adding synchronization constraints, precedence relationships, and staged routing decisions. As a generalisation of well-known $\mathcal{NP}$-hard combinatorial problems such as the CVRP, the VRPTW, and, in this case, the MCVRP, exact methods become computationally infeasible for large-scale instances due to the exponential growth of the solution space [58–64]. Consequently, heuristic and metaheuristic approaches are commonly employed to derive near-optimal solutions within reasonable computational times.

This section presents this classical version of the VRPCD to introduce the problem that is the topic of this research. The mathematical formulation below tackles the VRPCD without time windows, and is modified from the classical version of Lee et al. [57]. Let the graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ define nodes $\mathcal{N} = \mathcal{S} \cup \mathcal{O} \cup \mathcal{R}$, where $\mathcal{S}$ represents the supplier nodes, $\mathcal{O}$ the set of CD facilities (in this case a singleton set), and $\mathcal{R}$ the retailer nodes. Furthermore, let the arc set $\mathcal{A}$ define the arc set $\{(i,j) : i,j \in \mathcal{S} \cup \mathcal{O}, i \neq j\} \cup \{(i,j) : i,j \in \mathcal{R} \cup \mathcal{O}, i \neq j\}$. Note that δ^+ and δ^- denote the set of successors and predecessors, respectively, as previously defined.

Furthermore, let the loading quantity at pick up node $i \in \mathcal{S}$ be denoted by p_i, and the unloading quantity at delivery node $i \in \mathcal{R}$ be denoted by d_i. In addition, y_{ij} and z_{ij} represent the quantity of products transported from node i to node j in the pick up and delivery process, respectively. The travel time along arc (i,j) is represented by t_{ij}, and the length of a visit of a vehicle in node i is equal to s_i. AT^v is the arrival time of vehicle v at the CD, and DT_i^v is the departure time of vehicle v from node i. Lastly, T represents the planning horizon in which all customers must be served.

For each arc $(i,j) \in \mathcal{A}$ and each vehicle $v \in \mathcal{V}$, there is a binary arc-flow variable x_{ij}^v that is equal to 1 if arc (i,j) is used by vehicle v, and 0 otherwise.

$$\min \quad \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^v \tag{7a}$$

$$\text{subject to} \quad \sum_{v \in \mathcal{V}} \sum_{i \in \delta^-(j)} x_{ij}^v = 1 \qquad \forall\, j \in \mathcal{N} \tag{7b}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \delta^+(i)} x_{ij}^v = 1 \qquad \forall\, i \in \mathcal{N} \tag{7c}$$

$$\sum_{i \in \delta^-(j)} x_{ij}^v - \sum_{i \in \delta^+(j)} x_{ji}^v = 0 \qquad \forall\, v \in \mathcal{V},\ j \in \mathcal{N} \tag{7d}$$

$$\sum_{j \in \delta^+(0)} x_{0j}^v \leq 1 \qquad \forall\, v \in \mathcal{V} \tag{7e}$$

$$\sum_{i \in \delta^-(0)} x_{i0}^v \leq 1 \qquad \forall\, v \in \mathcal{V} \tag{7f}$$

$$\sum_{v\in\mathcal{V}}\sum_{j\in\delta^+(0)} x_{0j}^v \le K \qquad \forall\, v\in\mathcal{V} \tag{7g}$$

$$y_{ij} + z_{ij} \le Q x_{ij}^v \qquad \forall\, v\in\mathcal{V},\ (i,j)\in\mathcal{A} \tag{7h}$$

$$\sum_{i\in\mathcal{N}} p_i = \sum_{i\in\mathcal{N}} d_i \tag{7i}$$

$$y_{jl} - y_{ij} = \begin{cases} p_j, & \text{if } j\in\mathcal{S} \\ 0, & \text{if } j\in\mathcal{R} \\ -\sum_{i\in\mathcal{N}} p_i, & \text{if } j\in 0 \end{cases} \qquad \forall\, i,l\in\mathcal{N} \tag{7j}$$

$$z_{ij} - z_{jl} = \begin{cases} 0, & \text{if } j\in\mathcal{S} \\ d_j, & \text{if } j\in\mathcal{S} \\ \sum_{i\in\mathcal{N}} d_i, & \text{if } j\in 0 \end{cases} \qquad \forall\, i,l\in\mathcal{N} \tag{7k}$$

$$\sum_{i,j\in\mathcal{N}} s_i^v x_{ij}^v + \sum_{i,j\in\mathcal{N}} t_{ij}^v x_{ij}^v \le T \qquad \forall v\in\mathcal{V} \tag{7l}$$

$$DT_j^v = (t_{ij} + DT_i^v + s_j)\, x_{ij}^v \qquad \forall\, v\in\mathcal{V} \tag{7m}$$

$$AT^v = (DT_i^v + t_{i0})\, x_{i0}^v \qquad \forall\, v\in\mathcal{V}, i\in\mathcal{N} \tag{7n}$$

$$AT^v = AT^{v'} \qquad \forall\, k\neq k' \tag{7o}$$

The objective function (7a) aims to minimise the total costs of serving all customers within the planning horizon T, by determining the number of vehicles and the best route, schedule, and arrival time of each vehicle at the CD. Constraints (7b) - (7f) are constraints ensuring that all vehicles arrive and leave all nodes, and whether or not a vehicle arrives and leaves the CD. Constraints (7g) guarantee that the number of vehicles used to satisfy all demand is less than the number of available vehicles V. Constraints (7h) is a capacity constraint imposed on all vehicles, and constraint (7i) ensures flow conservation between the pick up and delivery. The quantity of transported goods is depicted by constraints (7j) and (7k). Constraints (7l) impose a time horizon on the complete operation, equal to T, such that the sum of the total length of the visit to each node and total transportation time must be less than this planning horizon. Constraints (7m) ensure that the departure time of a vehicle is determined by the sum of the arrival time at a node, the length of a visit, and time to move. In addition, constraints (7n) express that the arrival time at the CD is equal to the sum of the departure time of the previous node and the travel time between these. Constraints (7o) sets the simultaneous arrival of vehicles at the CD.

It should be noted that Lee et al. [57] consider an objective function that incorporates a fixed costs for every vehicle used. These fixed costs are left out for simplicity, but play a cruexpresscial role when optimising fleet utilisation. For further details on the objective function of this study, refer to Section 4.3. Furthermore, the model presented in this section does not include time windows for each pick up and delivery and contains the strong assumptions that the consolidated goods must arrive simultaneously. Later models included different approaches where the dependency among the vehicles is determined by consolidation decisions, such as in Wen et al. [63].

3.6 Related Work

This subsection provides an overview of some similar applications in the extensive literature on the VRP. In particular, it examines the literature that concerns the VRPCD as presented in Subsection 3.5 to gain a deeper understanding of the intricacies associated with the problem.

Lee et al. [57] first introduced the VRPCD, where the objective of the problem is to determine the number of vehicles and a set of vehicle schedules with a minimum sum of operational cost and transportation cost. A Tabu Search (TS) algorithm was proposed which obtained solutions with an average gap of under 5% compared to the optimal results within a reasonable computing time. Building on this work, Liao et al. [65] proposed a new TS algorithm where the solution quality showed a 10-36% improvement, depending on the problem instance, in considerably less computing time than the original TS algorithm. Both approaches, however, limit the transportation time for the pick up and delivery process while disregarding time windows imposed by customers.

In a different approach, Yu et al. [66] presented an algorithm based on Simulated Annealing (SA) to solve another variant that considers the open variant of the VRPCD, where vehicles are allowed to end at a customer, not considering their returning trip. The SA approach further improved the solutions of many benchmark instances compared to existing optimisers. Furthermore, Santos et al. [67] studied a variant where costs are associated with transferring load between vehicles using a branch-and-price algorithm. Unlike our problem formulation, the study does not incorporate time windows, break constraints, or multi-product transportation, limiting its direct application. Furthermore, the branch-and-price approach focuses on exact optimisation, which is computationally impractical for large-scale instances where heuristic approaches are preferred.

Wen et al. [63] extended the traditional VRPCD to include time windows. Whereas the typical VRPCD aims to minimise the sum of the total travel and operational costs, the objective became the minimisation of the total travel time while adhering to the time windows. The resulting VRPCD with Time Windows (VRPCDTW) has received considerable attention. Multiple solving algorithms were proposed such as TS [63, 68, 69], Variable Neighborhood Search (VNS) [69], Genetic Algorithm (GA) [70], matheuristic [71], and Iterated Local Search (ILS) [72]. Grangier et al. [73] studied the case where the number of docks that can be used simultaneously is limited by altering the matheuristic presented earlier. The key characteristics of these implementations are summarised in Table 3.2, highlighting their incompatibility with our problem formulation.

In addition to routing decisions, Ting and Chen [62] studied the scheduling of vehicle arrivals at the CD. This extension, which determines the sequence of arrivals, is crucial when the number of docking doors at the facility is insufficient or fewer than the number of vehicles used [59]. Given that both routing and scheduling problems are $\mathcal{NP}$-hard, two Ant Colony Optimisation (ACO) algorithms were employed: one to determine vehicle routing and the other to schedule vehicle arrivals. In this approach, routing decisions provide input for the scheduling problem. This approach is, however, not directly applicable, as it assumes a fixed fleet size, whereas our model allows for variable vehicle availability. Additionally, time windows and break times are not considered, which are key constraints in our formulation.

Table 3.2 summarises the key features of related VRPCD studies, such as time windows, break times, multiple products, variable vehicles, and direct shipments. These features were selected based on expert interviews [4, 5], ensuring relevance to the operations. Note that this table is not exhaustive but focuses on models that might be relevant to this research.

Table 3.2: Overview of VRPCD Literature

Literature	*Time windows*	*Break times*	*Multi product*	*Variable vehicles*	*R-S*	*Approach*
Lee et al. [57]	–	–	–	✓	–	TS
Liao et al. [65]	–	–	–	✓	–	TS
Gunawan et al. [59]	–	–	–	–	–	Matheuristic (ALNS[a] + SA)
Dondo and Cerdá [74]	–	–	–	–	–	Sweep based approach
Wen et al. [63]	✓	–	–	–	–	TS
Tarantilis [68]	✓	–	–	–	–	Adaptive multi-restart TS
Esfahani and Fakhrzad [69]	✓	–	✓	–	–	TS + VNS
Touihri et al. [70]	✓	–	–	–	–	GA
Grangier et al. [71]	✓	–	–	–	–	Matheuristic based LNS[b]
Urtasun and Montero [75]	✓	–	–	–	–	GRASP[c]
Morais et al. [72]	✓	–	–	–	–	ILS
Grangier et al. [73]	✓	–	✓	–	–	Matheuristic based LNS
Dondo and Cerdá [76]	✓	–	–	–	–	Sweep based approach
Baniamerian et al. [64]	✓	–	–	✓	–	Matheuristic (ALNS + SA)
This research	✓	✓	✓	✓	✓	Hybrid TS-SA

[a] Adaptive Large Neighbourhood Search (ALNS), [b] Large Neighbourhood Search (LNS), [c] Greedy Adaptive Search Procedure (GRASP)

The review of related work indicates that most research has concentrated on developing heuristics and metaheuristics, which perform effectively even on large VRPCD instances. This research aims to develop an implementation that performs well even on larger instances, motivating a similar approach.

Despite extensive research on the VRPCD, existing models often fail to incorporate multiple real-world constraints simultaneously. While prior studies have considered variations such as capacity constraints, time windows, multi-product transportation, and cross-docking, few approaches unify these aspects into a comprehensive implementation. Many models assume a fixed fleet size or time windows, whereas this study adopts a variable fleet approach with time windows at supplier and retailer locations.

This study addresses these limitations by contributing to the literature in the following ways:

- Developing an adaptive TS-SA hybrid algorithm that dynamically balances intensification and diversification.
- Integrating cross-docking dynamics with time windows, break times and multi-product constraints to reflect real-world logistics challenges.

3.7 Findings and Implications

3. *What are the key factors influencing routing and consolidation decisions in cross-docking logistics?*

Chapter 3 identifies key factors influencing routing decisions as described in the existing literature. These include:

- Time windows: Constraints ensuring deliveries and pick ups occur within predefined slots.
- Fleet constraints: Limitations on vehicle capacities and scheduling to maintain operational efficiency.

- Multi-compartment Vehicles: The need to transport different products separately within the same vehicle adds complexity.
- Cross-docking Operations: Efficient unloading, sorting, and reloading of goods at a central facility to optimise deliveries.
- Cost considerations: The goal is to minimise both fixed and variable transportation costs while balancing service-level agreements.

4. *How can the vehicle routing problem in a cross-docking supply chain be modelled and optimised?*

Chapter 3 provides a broad overview of various modelling approaches for VRPs, particularly those that incorporate constraints such as time windows, multi-compartment vehicles, and cross-docking operations. It highlights that existing models typically focus on determining the number of vehicles required or optimising the transportation of multiple products through a CD supply chain, but rarely both simultaneously. The literature review indicates that metaheuristic approaches (e.g., TS, ALNS) have shown promise in solving large-scale VRP instances. However, the specific constraints and operational challenges faced by Wolter Koops require a tailored approach, as standard VRP models do not fully capture the complexities of temperature-controlled logistics with multi-compartment vehicle constraints, cross-docking synchronization, strict time windows, and rest periods.

These findings form a theoretical foundation for understanding routing constraints, and complement the case-specific factors influencing routing decisions identified in Section 2.2. Whereas that section examined operational and contextual aspects of Wolter Koops' routing decisions, Chapter 3 presents generalised insights identified in VRP literature. Chapter 4 extensively addresses these factors by integrating them into a structured problem definition and a mathematical model.

4 Problem Formulation and Mathematical Framework

With the theoretical foundations established in the previous chapter, this chapter formulates a precise problem definition and develops a mathematical framework to guide the selection and implementation of a suitable solution method. By formulating the problem precisely, this chapter provides a basis for addressing the complexities of vehicle routing in a cross-docking supply chain. Therefore, it contributes to Phase 3: Solution Design of the research framework.

The following research questions are addressed in this chapter:

5. *What are the key assumptions made in the solution design?*

Section 4.1 presents a detailed formalisation of the routing problem to ensure the solution method's alignment with the operational context of Wolter Koops. Section 4.2 provides a network-based representation of the routing problem. Section 4.3 introduces the mathematical formulation, detailing the decision variables, constraints, and objective function. Section 4.4 discusses the necessary assumptions and simplifications required for computational feasibility. Section 4.5 presents a simplified problem instance to demonstrate the model's complexity and guide future solution development.

4.1 Formal Problem Statement

The transportation and logistics sector plays a vital role in global supply chains, with companies like Wolter Koops specialising in temperature-controlled goods and offering complex services to deliver goods from suppliers to retailers. Wolter Koops operates a fleet of approximately 1,000 vehicles and 1,200 trailers. However, the current routing process heavily relies on manual planning despite the complex nature of routing multi-compartment vehicles in a cross-docking supply chain. With over 5,500 trips, which are defined as any travel between two locations, such as a delivery to a retailer, planned weekly company-wide, this practice likely leads to inefficiencies in route optimisation. It introduces challenges related to time window compliance and fleet size optimisation.

While numerous VRP studies exist, the complexity of integrating cross-docking operations and addressing fleet-specific constraints, such as time windows and loading/unloading operations, has not been sufficiently explored within the context of temperature-sensitive logistics (see Section 3.6). Therefore, this research addresses the problem to improve operational costs and service quality.

The problem involves a cross-docking distribution network with multiple suppliers, denoted by the set $\mathcal{S}$, and various retailers, denoted by the set $\mathcal{R}$, serviced by a single CD. Each arc $(i, j) \in \mathcal{A}$ has an associated travel time t_{ij}, representing the time required for a vehicle to traverse from node i to node j. In addition, retailers and suppliers have limited time availability, represented by time windows $[e_i, l_i]$ $\forall\, i \in \mathcal{N} \setminus \{CD\}$. Loading and unloading operations take A minutes and may only begin within these designated time windows. The CD is assumed to be available at all times.

The CD is located at a fixed, pre-determined site and consists of multiple terminals for inbound vehicles (receiving pallets from suppliers) and outbound vehicles (delivering goods to retailers). Commodity classes are denoted by the set $\mathcal{M}$, and each vehicle can transport up to two distinct classes of products due to its dual-compartment design, ensuring the separation of incompatible goods. The inter-commodity incompatibilities are modelled similarly to the formulation in Derigs et al. [53], where:

$$\mathcal{I}_{comm} = \{(m, m') \mid m, m' \in \mathcal{M}, m \neq m', \text{ commodities } m \text{ and } m' \text{ are incompatible}\}$$

The compartments can be configured to any capacity as long as the sum of both compartments equals the vehicle's total capacity.

Each order $o \in \mathcal{O}$ represents a specific request for the transportation of goods, characterised by the following parameters:

- $s_o \in \mathcal{S}$: The supplier from which order o originates.
- $r_o \in \mathcal{R}$: The retailer to which order o must be delivered.
- $m_o \in \mathcal{M}$: The commodity class associated with order o.
- d_o: The demand quantity of order o in units.
- $[e_o, l_o]$: The allowable time window for order o at its destination.

Each retailer $r \in \mathcal{R}$ receives goods of commodity type m from suppliers in $\mathcal{S}$, represented by a set of individual orders $\mathcal{O}_r$. Similarly, each supplier $s \in \mathcal{S}$ ships goods to retailers in $\mathcal{R}$, represented by a set of orders $\mathcal{O}_s$. Each order $o \in \mathcal{O}$ is associated with exactly one commodity $m_o \in \mathcal{M}$.

Goods must be transported through the CD using a fleet of homogeneous vehicles $v \in \mathcal{V}$, where all vehicles have a fixed capacity Q.

Each vehicle incurs a fixed cost c_f when deployed in any route type (pick up, delivery, transfer, or direct service). Additionally, a variable cost c_{var} is associated with each minute of travel time, where the total travel cost depends on the duration t_{ij} required to traverse each arc $(i, j) \in \mathcal{A}$. The variable cost c_{var} is applied to all route types once, including transferring routes, as each vehicle is deployed in the planning horizon, and its presence in any route incurs operational costs regardless of specific movements.

Upon arrival at the CD, goods are sorted and consolidated onto multiple outbound vehicles for delivery. Goods can be transferred directly to an outgoing vehicle or temporarily stored for a short duration before being assigned to a route. However, this study does not address the detailed scheduling of vehicles to specific terminals within the cross-docking facility.

4.2 Graph-Based Representation of the Network

To formally describe the VRP under study, we represent the logistics network of Wolter Koops as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$, defined by:

- $\mathcal{N}$ is the set of nodes, where each element represents a distinct physical location within the logistics network. The set consists of:
  - $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$, the set of suppliers, where goods originate. Each supplier represents a location where goods are picked up. At least one vehicle must

visit these, ensuring all required goods enter the system. A supplier location can be visited multiple times when a single supplier location requests multiple transport orders.

– $\{CD\}$, the CD, where goods are consolidated before final delivery. The CD is the central consolidation point where shipments from different suppliers can be combined and redirected for delivery.

– $\mathcal{R} = \{r_1, r_2, \ldots, r_{|\mathcal{R}|}\}$, the set of retailers, which receive goods. Each retailer represents a customer delivery location and must be visited to ensure the fulfilment of orders. A retailer location can be visited multiple times when a single retailer location requests multiple transport orders.

The set of nodes is therefore defined as:

$$\mathcal{N} = \mathcal{S} \cup \{CD\} \cup \mathcal{R}$$

- $\mathcal{A}$ is the set of arcs, which consists of all feasible travel connections between suppliers, retailers, and the CD. The arc set is defined as:

$$\mathcal{A} = \{(i, j) : i, j \in \mathcal{S} \cup \{CD\} \cup \mathcal{R}, i \neq j\}$$

The network structure is depicted in Figure 4.1, where nodes and arcs illustrate the movement of goods across the supply chain. In the exemplary graph structure, the node set $\mathcal{N}$ contains seven elements: two suppliers, four retailers, and one CD node. Note that the arc set $\mathcal{A}$ contains all possible combinations of nodes, as defined previously.

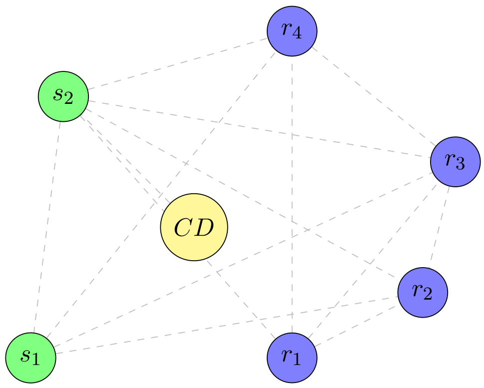


FIGURE 4.1: Graph representation of the VRPCD

The graph allows direct transportation between two locations in the network, including supplier-to-supplier, retailer-to-retailer, and CD-mediated routes. Each arc $(i, j) \in \mathcal{A}$ has an associated travel time. While the graph captures all possible movements within the network, the actual vehicle routes are determined by operational constraints. These constraints include vehicle capacity limitations, restricting the amount of goods a vehicle can carry per trip; time window constraints, where certain suppliers or retailers have fixed time slots for pick ups or deliveries; and routing rules, which may require vehicles to pass through the CD before reaching retailers, thereby restricting direct supplier-to-retailer connections.

Vehicles can follow one of four active routing strategies, each defining a different approach to moving goods within the supply chain. These are graphically represented in Figure 4.2, and formally represented by the arc set $\mathcal{A}$ as follows:

- pick ups only: The vehicle collects goods from one or more suppliers and returns them to the cross-docking terminal. This corresponds to the arc set:

$$\mathcal{A} = \{(i,j) : i,j \in \mathcal{S} \cup \{CD\}, i \neq j\}$$

- Deliveries only: The vehicle departs from the CD and transports goods to one or more retailers before returning. This corresponds to the arc set:

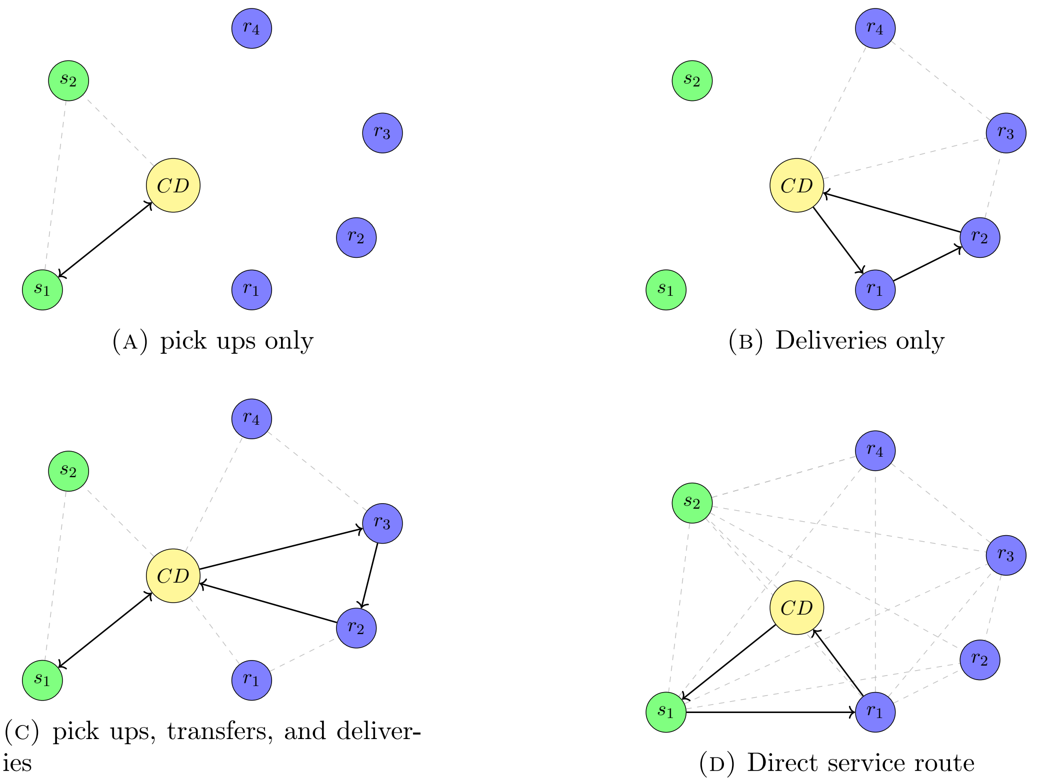
$$\mathcal{A} = \{(i,j) : i,j \in \{CD\} \cup \mathcal{R}, i \neq j\}$$

- pick ups, transfers, and deliveries (i.e. transferring route): The vehicle first picks up goods, returns to the CD, and subsequently delivers them to retailers before returning. This corresponds to the arc set:

$$\mathcal{A} = \{(i,j) : i,j \in \mathcal{S} \cup \{CD\}, i \neq j\} \cup \{(i,j) : i,j \in \{CD\} \cup \mathcal{R}, i \neq j\}$$

- Direct supplier-to-retailer service: In cases where a vehicle directly services an order without returning to the CD, a one-to-one supplier-retailer ($\mathcal{S}$: $\mathcal{R}$) pairing is established. This corresponds to the arc set:

$$\mathcal{A} = \{(i,j) : i,j \in \mathcal{S} \cup \{CD\} \cup \mathcal{R}, i \neq j\}$$



(A) pick ups only

(B) Deliveries only

(C) pick ups, transfers, and deliveries

(D) Direct service route

FIGURE 4.2: Individual route possibilities (idle state omitted)

Additionally, vehicles may remain idle throughout the planning horizon:

- Idle state: The vehicle remains unused at the CD throughout the planning horizon. Although this scenario is considered in the routing model, it is not explicitly visualised in Figure 4.2.

4.3 Mathematical Formulation

Using the network representation and route notation, we now define the mixed integer linear programming formulation for the VRPCD specific to the logistical setting described.

4.3.1 Decision Variables

The following decision variables define the key routing and scheduling choices in the VRPCD regarding vehicle movements, order assignment, and timing and sequences.

- Routing Variables:
  - $x_{i,j}^{v}$: Binary variable indicating whether vehicle v travels arc (i, j).
  - y^{v} : Binary variable indicating whether vehicle v is used in any route.
  - $\delta_{o,m}^{v}$: Binary variable indicating whether vehicle v directly transports order o of commodity m from its supplier to its retailer, bypassing the CD.
- Order Assignment Variables:
  - $\pi_{o,m}^{v}$: Binary variable indicating whether vehicle v picks up order o of commodity m.
  - $\lambda_{o,m}^{v}$: Binary variable indicating whether vehicle v delivers order o of commodity m.
  - $\tau_{o,m}^{v,v'}$: Binary variable indicating whether order o of commodity m is transferred at the CD from vehicle v to vehicle v'.
  - $z^{v,v'}$: Binary variable that equals one if at least one order is transferred from vehicle v to v'
- Time and Sequence Variables:
  - T_{i}^{v} : Continuous variable representing the start of service time at node i when serviced by vehicle v.
  - W_{i}^{v} : Continuous variable representing the waiting time at node i when serviced by vehicle v.
  - u_{i}^{v} : Continuous variable representing the visit order of node i in vehicle v's route.

4.3.2 Objective Function and Constraints

The primary objective is to minimise the total cost associated with the selected vehicle routes:

$$\min c_f \sum_{v \in \mathcal{V}} y^{v} + c_{var} \sum_{(i,j) \in \mathcal{A}, v \in \mathcal{V}} t_{ij} x_{ij}^{v} \tag{8a}$$

Where:

- c_{var} represents a variable cost component incurred per minute travelled by any vehicle.
- c_f represents a fixed cost component incurred per vehicle in the planning horizon.

The full mixed-integer linear program is given by the objective function as defined by Equation (8a) and the constraints as follows:

$$
\begin{aligned}
\min \quad & c_f \sum_{v \in \mathcal{V}} y^v + c_{var} \sum_{(i,j) \in \mathcal{A}, v \in \mathcal{V}} t_{ij} x_{ij}^v && && \text{(8a)} \\
\text{subject to} \quad & \sum_{j \in \mathcal{N}} x_{i,j}^v = \sum_{j \in \mathcal{N}} x_{j,i}^v && \forall\ (i,v) && \text{(8b)} \\
& \sum_{j \in \mathcal{N}} x_{CD,j}^v \geq y^v && \forall\, v && \text{(8c)} \\
& \sum_{i \in \mathcal{N}} x_{i,s_o}^v \leq \pi_{o,m}^v && \forall\ (o,m,v) && \text{(8d)} \\
& \sum_{i \in \mathcal{N}} x_{i,r_o}^v \leq \lambda_{o,m}^v && \forall\ (o,m,v) && \text{(8e)} \\
& \pi_{o,m}^v + \lambda_{o,m}^v \leq 2y^v && \forall\ (o,m,v) && \text{(8f)} \\
& \sum_{v \in \mathcal{V}} \pi_{o,m}^v = 1 && \forall\ (o,m) && \text{(8g)} \\
& \sum_{v \in \mathcal{V}} \lambda_{o,m}^v = 1 && \forall\ (o,m) && \text{(8h)} \\
& \sum_{o \in \mathcal{O}} \sum_{m \in \mathcal{M}} d_o \cdot \pi_{o,m}^v \leq Q && \forall\, v && \text{(8i)} \\
& \sum_{o \in \mathcal{O}} \sum_{m \in \mathcal{M}} d_o \cdot \lambda_{o,m}^v \leq Q && \forall\, v && \text{(8j)} \\
& \sum_{(m,m') \in \mathcal{I}_{comm}} \left(\pi_{o,m}^v + \pi_{o,m'}^v \right) \leq k && \forall\ (o,v) && \text{(8k)} \\
& \sum_{(m,m') \in \mathcal{I}_{comm}} \left(\lambda_{o,m}^{v'} + \lambda_{o,m'}^{v'} \right) \leq k && \forall\ \left(o,v'\right) && \text{(8l)} \\
& \sum_{\substack{v \in \mathcal{V} \\ v \neq v'}} \tau_{o,m}^{v,v'} \leq \lambda_{o,m}^{v'} && \forall\ \left(o,m,v'\right) && \text{(8m)} \\
& \sum_{\substack{v' \in \mathcal{V} \\ v' \neq v}} \tau_{o,m}^{v,v'} + \lambda_{o,m}^v \leq \pi_{o,m}^v && \forall\ (o,m,v) && \text{(8n)} \\
& z^{v,v'} \geq \tau_{o,m}^{v,v'} && \forall\ \left(o,m,v,v'\right), v \neq v' && \text{(8o)} \\
& e_i \leq T_i^v + W_i^v \leq l_i && \forall\ (i \setminus CD, v) && \text{(8p)} \\
& W_i^v = max\{0, e_i - T_i^v\} && \forall\ (i \setminus CD, v) && \text{(8q)} \\
& W_{CD}^v \geq T_j^v - t_{CD,j} x_{CD,j}^v - 2A && \forall\ (j,v) && \text{(8r)} \\
& T_j^v = T_i^v + W_i^v + A + t_{ij} x_{i,j}^v && \forall\ (i \setminus CD, j, v) && \text{(8s)} \\
& T_j^v = T_{CD}^v + W_{CD}^v + 2A + t_{CD,j} x_{CD,j}^v && \forall\ (j,v) && \text{(8t)} \\
& T_j^v - t_{CD,j} x_{CD,j}^v = max\{T_{CD}^v, T_{CD}^{v'}\} && && \\
& \qquad + W_{CD}^v + 2A - M\left(1 - z^{v,v'}\right) && \forall\ \left(j,v,v'\right), v \neq v' && \text{(8u)} \\
& T_{CD}^{v'} + W_{CD}^{v'} \geq T_{CD}^v + 2A z^{v,v'} && \forall\ \left(v,v'\right), v \neq v' && \text{(8v)} \\
& u_i^v - u_j^v + |\mathcal{N}| \cdot x_{i,j}^v \leq |\mathcal{N}| - 1 && \forall\ (i,j,v), i \neq j && \text{(8w)}
\end{aligned}
$$

$$x_{i,j}^v, y^v, \delta_{o,m}^v \text{ binary} \qquad \forall\ (i, j, m, o, v) \tag{8x}$$
$$z^{v,v'}, \pi_{o,m}^v, \lambda_{o,m}^v, \tau_{o,m}^{v,v'} \text{ binary} \qquad \forall\ (o, m, v, v'), v \neq v' \tag{8y}$$
$$T_i^v, W_i^v, u_i^v \geq 0 \qquad \forall\ (i, v) \tag{8z}$$

The objective function in Equation (8a) minimises total costs, combining fixed vehicle costs and variable travel costs. Equation (8b) ensures flow conservation, such that entries and exits at each node are balanced. Constraints (8c) regulate CD accessibility, requiring active vehicles ($y^v = 1$) to have valid inbound and outbound connections, while (8d)–(8e) ensure order feasibility by restricting pick ups and deliveries to designated locations. Equations 8f-(8h) enforce that vehicles may pick up and/or deliver all orders.

Capacity limits are imposed in (8i) and (8j), ensuring vehicle loads at pick up and delivery do not exceed Q. Constraint (8k) and (8l) prevent vehicles from picking up or delivering more incompatible commodity classes than the number of compartments, respectively (in our problem setting, $k = 2$). Furthermore, Equations (8m) and (8n) ensure that a receiving vehicle at the CD must also deliver the orders, and vehicles that pick up orders from suppliers must either transfer or deliver these.

Constraints (8o) link decisions variables $z^{v,v'}$ and $\tau_{o,m}^{v,v'}$. Equations (8p)-(8v) are time window constraints. Note that these are based on the mathematical formulation provided in Section 3.3. However, the addition of the CD require the addition of Constraints (8t), (8u), and (8v) to ensure that transferring vehicles require extra service time and vehicles depart after receiving their goods (if applicable). Note that M represents a sufficiently large number to render Constraints 8v non-binding when necessary. We recommend setting M to the duration of the planning horizon.

Constraints (8w) are subtour elimination constraints, and Equations (8x)-(8z) are sign restrictions on the decision variables.

4.4 Assumptions and Simplifications

This section presents the assumptions and simplifications made to make the complex intricacies inherent to the transportation industry manageable. These serve to make the problem computationally tractable while preserving its key characteristics. Their potential impact on the validity and generalisability of the results is considered in Chapter 8.

4.4.1 Assumptions

Each vehicle is assigned to at most one route throughout the planning horizon. The routing framework assumes that all vehicles must depart from and return to the CD at the beginning and end of their respective routes. Furthermore, each vehicle incurs a fixed cost c_f irrespective of whether it performs pick ups or deliveries. This assumption is justified by the operational requirement that charter vehicles must be pre-arranged, leading to an all-or-nothing cost structure where a vehicle is either leased or not.

The fleet is considered homogeneous, as Wolter Koops operates approximately 1,000 vehicles and 1,200 trailers, all identical in range, capacity, and loading capabilities. While the company also owns 26 dolly units, their inclusion introduces additional complexity concerning loading and unloading operations. To maintain tractability, these units are not explicitly modelled.

The CD is assumed to have an infinite storage capacity, meaning there is no upper limit on the number of pallets that can be temporarily stored. Additionally, the number of docks is considered infinite, ensuring that all vehicles arriving at the CD can be serviced immediately without experiencing delays.

Unloading and loading operations require a fixed duration of A minutes per action. Consequently, a complete transfer of goods involving both actions is approximated to require $2A$ minutes. While these operations are realistically load-dependent, this assumption prevents incentives that could otherwise lead to inefficient vehicle utilisation. Specifically, if service times were modelled as load-dependent, prioritising shorter service times for smaller loads might unintentionally discourage full vehicle utilisation.

Holding costs are not considered, and it is assumed that all goods need to be delivered by the end of the planning horizon. Additionally, transportation orders may be rejected if they cannot be fulfilled within the required time constraints—that is if the earliest feasible delivery time exceeds the retailer's available time window.

While the CD remains continuously available, suppliers and retailers operate within predefined time windows, requiring that service at these locations begins within their respective availability periods. Finally, all orders are assumed to be single-commodity, such that $d_o^m = d_o$ if $m = m_o$ and 0 otherwise. Thus, an order cannot consist of multiple commodity types, thereby excluding mixed shipments from the scope of this study.

4.4.2 Simplifications

Breaks are scheduled regularly to comply with legal driving regulations. Specifically, a 45-minute break is required after every 4.5 hours of driving, and after every 9 hours of driving (i.e. following two 4.5-hour intervals), an extended break of 11 hours is mandated. Although some additional rules and exceptions allow for more flexibility in real-world scenarios [77], these have been excluded from the current implementation. This simplification is intentional, as the exact regulations can vary between countries, and incorporating such exceptions would significantly complicate the model at the cost of its generalisability.

4.5 Toy Problem Instance

This section presents an example of a toy problem corresponding to this chapter's problem definition. The purpose is to illustrate the complexities common to more advanced instances of the problem rather than to explain the solution method used in this study. Additionally, a solution is presented here.

In this toy problem instance, the same network is considered as in Figure 4.2, consisting of two suppliers ($s1, s2 \in \mathcal{S}$, four retailers $r_1, r_2, r_3, r_4 \in \mathcal{R}$, and one CD. Table 4.1 represents transport orders of suppliers and retailers, including time windows for pick up and delivery, the number of EUR pallets (EPP), and the commodity class per order.

In addition to understanding the characteristics of the nodes, it is crucial to analyse the travel times between all nodes. To this end, we calculate the Euclidean distances between all possible nodes in the network and construct a distance matrix, as shown in Table 4.2 (we do not include coordinates here for the sake of conciseness. These can be found under Appendix B). For the sake of simplicity, we assume a uniform travel speed of one time unit per distance unit, allowing us to equate the distances directly with the corresponding

TABLE 4.1: Transport Orders

Order ID	Supplier ID	Retailer ID	EPP	TW Supplier	TW Retailer	Commodity Class
1	s_1	r_1	5	$[0, 5]$	$[10, 14]$	A
2	s_2	r_3	5	$[0, 3.61]$	$[20, 24]$	B
3	s_1	r_4	5	$[5, 10]$	$[25, 27]$	B
4	s_2	r_2	5	$[3, 7]$	$[35, 40]$	B

travel times. Furthermore, the cost of travelling from node i to node j leads to a variable cost c_{ij} of one per distance/time unit.

TABLE 4.2: Euclidean distance/travel time/cost matrix

	CD	s_1	s_2	r_1	r_2	r_3	r_4
CD	0.00	4.47	3.61	2.24	4.12	5.10	3.61
s_1	4.47	0.00	4.12	5.00	8.06	9.49	7.81
s_2	3.61	4.12	0.00	5.66	7.61	8.06	5.10
r_1	2.24	5.00	5.66	0.00	3.16	5.00	5.10
r_2	4.12	8.06	7.61	3.16	0.00	2.24	4.47
r_3	5.10	9.49	8.06	5.00	2.24	0.00	3.61
r_4	3.61	7.81	5.10	5.10	4.47	3.61	0.00

We consider three vehicles $v_1, v_2, v_3 \in \mathcal{V}$ to be homogeneous with identical vehicle capacities $Q = 10$. Employing one vehicle results in a fixed cost of $c_f = 10$, and each minute travelled results in a variable cost of $c_{var} = 1$. Additionally, (un)loading time at a (retailer) supplier location is equal to a fixed time component A. Furthermore, transfers at the CD require a service time of $2 * A$. We use $A = 1$ in this toy example.

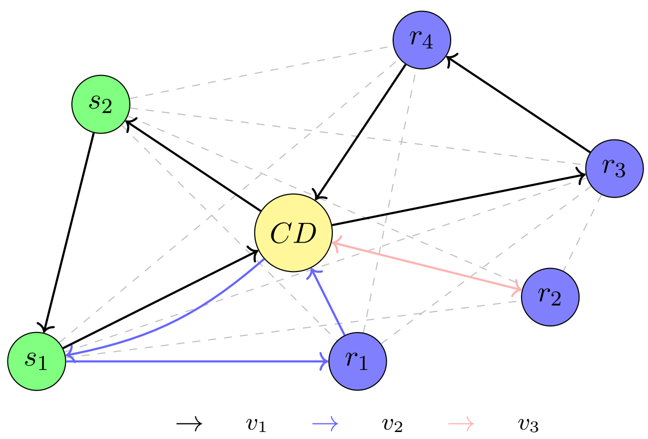


FIGURE 4.3: Toy problem solution illustration

Figure 4.3 presents an illustration of the solution of the toy problem instance. The solution can be represented by a vector for each vehicle $v \in \mathcal{V}$. In this case, the solution is identical to the vectors: v_1 : [CD $\rightarrow s_2 \rightarrow s_1 \rightarrow$ CD $\rightarrow r_3 \rightarrow r_4 \rightarrow$ CD], v_2 : [CD $\rightarrow s_1 \rightarrow r_1 \rightarrow$ CD], and v_3 : [CD $\rightarrow r_2 \rightarrow$ CD]. In this solution, vehicle v_3 traverses a direct service route from supplier to retailer because consolidation is impossible due to the tight time windows imposed in the corresponding transport order. Furthermore, vehicle v_1 picks up orders 1, 2, and 3 even though it cannot deliver these. Therefore, order 3 is loaded onto vehicle v_3 at the CD which delivers it separately. The objective function of the solution given in this example can easily be calculated by summing the total cost of travel of all trips and the fixed costs incurred per vehicle. This results in an objective value of 74.47.

While this representation suffices for a VRPCD without time window constraints, we incorporate time windows imposed by suppliers and retailers. Consequently, The solution's spatial and temporal dimensions are combined in a schedule, as shown in Table 4.3.

TABLE 4.3: Vehicle schedules with arrival- and departure times

Vehicle	Node	Arrival time	Departure time
v_1	CD	–	0.00
	s_2	3.61	4.61
	s_1	8.73	9.73
	CD	14.2	16.2
	r_3	19.81	20.81
	r_4	24.42	25.42
	CD	30.52	–
v_2	CD	–	0.00
	s_1	4.47	5.47
	r_1	10.47	11.47
	CD	13.71	–
v_2	CD	–	30.88
	r_2	35.00	36.00
	CD	40.12	–

4.6 Findings and Implications

This chapter detailed the problem's intricacies and provided a basis for selecting a suitable solution method. It, therefore, contributed to Phase 3: Solution Design of the research framework. The research question addressed in this chapter is 5.

5. *What are the key assumptions made in the solution design?*

The assumptions outlined in Section 4.4 establish the scope and computational tractability of the solution but also introduce constraints on its validity and applicability in real-world settings. Because of these assumptions, the model can facilitate efficient solution generation, making it suitable for large-scale routing problems. However, it is essential to note that several implications may arise as a consequence:

- The assumptions limit adaptability to real-time changes, meaning that the model performs best in predictable environments where demand and constraints remain stable. In dynamic logistics operations, additional re-optimisation mechanisms would be needed to handle disruptions.
- The model is tailored to a structured cross-docking environment with predefined planning horizons and fixed vehicle assignments. Its applicability to other logistics structures, such as multi-trip scheduling or decentralised fleet operations, may require modifications.

5 Adaptive TS-SA Hybrid Solution Approach

In this chapter, Phase 3: Solution Design is completed by exploring potential solution methods, selecting the most suitable, and designing the method to address the VRP in a cross-docking supply chain: the VRPCD. The research question tackled in this chapter is:

6. *What solution methods exist for enhancing vehicle routing decisions at Wolter Koops, and which is most suitable for implementation?*

This chapter develops a detailed plan to reach the research objective: to create an algorithm that minimises the costs of serving a set of geographically dispersed customers. Section 5.1 provides a concise literature review to develop a fitting solution approach for our problem. Section 5.2 presents the selected approach adopted in this study, after which Section 5.3 extends the SA-based diversification approach that is proposed. Finally, Section 5.4 concludes this chapter with the findings and implications of this chapter.

5.1 Exploration of Suitable Solution Methods

This section reviews the solution methods proposed in the literature over the past decades, generally categorised into exact and heuristic-based methods. It presents the types of solution methods to gain an understanding of them rather than to provide a comprehensive overview of all solution methods. To this end, an interested reader is referred to [22, 26, 28, 29]. Sections 5.1.1 and 5.1.2 explore exact and (meta)heuristic solution methods, respectively.

5.1.1 Exact Methods

Despite many efforts, solving instances of the CVRP to optimality in polynomial time is feasible only for relatively small instances, involving up to approximately 100 customers. Moreover, there is a significant variance in the computing times for these instances [78]. Variants incorporating additional complexities, such as time windows or multiple depots, present even more significant computational challenges. Therefore, exact methods cannot integrate the intricacies of larger, more complex instances [28]. As discussed in Section 3.5, heuristics and metaheuristics are commonly employed due to their effectiveness in addressing complex optimisation problems. This justifies applying such methods in the intricate VRPCD considered in this research.

5.1.2 (Meta)heuristic Methods

Real-life instances are often large and must be solved within a reasonable computing time. Therefore, heuristic and metaheuristic approaches (as well as their combinations) are widely used and have gained prominence in recent literature [22, 25].

Heuristic algorithms apply specially designed functions to intelligently explore the solution space, aiming to construct feasible solutions efficiently. These methods typically focus on problem-specific rules to generate good solutions quickly, though they do not guarantee optimality [79]. Heuristics can be categorised into constructive heuristics, which iteratively build a solution from scratch, and improvement heuristics, which refine an existing solution through local modifications [29, 31].

Metaheuristic algorithms, on the other hand, provide a higher-level framework that guides subordinate heuristics in exploring and exploiting the search space. These approaches use iterative generation processes and adaptive learning strategies to find high-quality, near-optimal solutions [79]. By balancing diversification and intensification, metaheuristics enable the efficient solving of complex problems within an acceptable time [80]. Section 5.1.2 defines metaheuristics and provides examples to clarify the distinction between heuristics and metaheuristics.

Thus, while heuristics are problem-specific techniques that construct or improve solutions based on predefined rules, metaheuristics act as generalised frameworks that enhance search effectiveness by dynamically guiding heuristics. The field of VRP (meta)heuristics is so extensive that this subsection can focus on only a small selection of methods, guided by the surveys conducted by Zhang et al. [26], Konstantakopoulos et al. [28], Toth and Vigo [29], Cordeau et al. [80].

Constructive Heuristics

Constructive heuristics were among the earliest heuristic methods [29]. They generate an initial route, which serves as input for subsequent improvement heuristics. Constructive heuristics typically start with an empty solution and iteratively build upon it by inserting customer(s) until all customers are routed. The insertion of customers can be performed either sequentially or in parallel [29].

Although constructive heuristics have historically received significant attention, sophisticated metaheuristics have become sufficiently robust so that the initial solution can be any random solution without affecting the quality of the solution [29]. Because of this, most constructive heuristics have now fallen into disuse. For further details, interested readers are directed to the work of Toth and Vigo [31].

Improvement Heuristics

Improvement heuristics are used to improve initial solutions often generated by constructive heuristics. Route improvement heuristics search for local improvements in the solution by applying perturbations, or modifications, to the solution. These result in either intra-route neighbourhoods if they operate on a single route at a time or inter-route neighbourhoods otherwise. Inter-route improvements are essential for achieving good results and include classical operators such as swapping consecutive customers between different routes (SWAP), removing customers from their route and inserting them elsewhere (RELOCATE), or removing k edges from a route and reconnecting them differently (k-OPT) [29].

The number of standard neighbourhoods of a routing problem is proportional to n^2 and k^2, which leads to a rapid increase in the number of possible operations as the number of customers (n) and the number of edges considered in the k-OPT (k) grow. Therefore, reducing the number of possible operations is necessary to allow a complete exploration of all neighbourhoods for larger instances. An example of such a pruning technique is the granular search used in Toth and Vigo [31], which reduces the number of potential operations by restricting moves between geographically distant customers. Inter-route improvement schemes can also involve alternating between moves that destroy part of the solution and moves that reconstruct the solution, such as in the ALNS and LNS metaheuristic (for more information on the ALNS procedure, the reader is referred to Pisinger and Ropke [81]).

Metaheuristic Methods

Unlike classical improvement heuristics, metaheuristics usually incorporate mechanisms to continue the exploration of the search space after a local minimum is encountered [80]. Metaheuristics can be described as higher-level frameworks that guide the application of other heuristics designed to find global optima across the entire solution space. They often combine improvement heuristics with stochastic elements to balance diversification and intensification [33]. This results in a more thorough solution space search, which means that metaheuristics are less likely to remain in a local optimum. Current metaheuristics can be classified into **local search** and **population-based** methods. This distinction forms the remaining outline of this section. Given the extensive attention that metaheuristics have received, covering all of them is beyond our scope. This section defines widely adopted approaches used in problem settings similar to the one in this research [29, 33, 80].

Local Search Methods explore the solution space starting with an initial solution x_1, moving iteratively from a solution x_t at iteration t to a solution x_{t+1} in the neighbourhood $\mathcal{N}(x_t)$ until a certain stopping criterion is satisfied [80]. Note that if $f(x)$ denotes the costs incurred in solution x, $f(x_{t+1})$ is not necessarily less than $f(x_t)$. Thus, local search algorithms should implement mechanisms to avoid cycling (i.e. revisiting the same solutions repeatedly).

SA [82] prevents cycling by balancing diversification (i.e. exploration) and intensification (i.e. exploitation) by selecting a random solution $f(x)$ in $\mathcal{N}(x_t)$. If $f(x) \leq f(x_t)$, then $x_{t+1} = x$. Otherwise:

$$x_{t+1} = \begin{cases} x & \text{with probability } p_t \\ x_t, & \text{with probability } 1 - p_t \end{cases} \tag{9}$$

where p_t is a decreasing function of t and $f(x) - f(x_t)$, often defined as:

$$p_t = \exp\left(-\frac{f(x) - f(x_t)}{T_t}\right) \tag{10}$$

where T_t denotes the temperature at iteration t which is a decreasing function of t. Since the acceptance probability of a new solution depends only on the current state and not on past transitions, SA inherently follows a Markov Chain process at a fixed temperature. This Markovian property allows SA to accept inferior solutions to explore the search space probabilistically. Deterministic Annealing (DA) [83] is similar to SA, but the rule for selecting x is deterministic. If the best known solution is x^*, the solution in iteration $t+1$ is selected according to the rule $x_{t+1} = x$ if $f(x_{t+1}) \leq \sigma f(x^*)$, where σ is a user-controlled parameter that is usually slightly greater than 1 (e.g. 1.05). If $f(x_{t+1}) > \sigma f(x^*)$, then $x_{t+1} = x_t$.

TS was first proposed by Glover [84], which prevents cycling by declaring solutions tabu, or forbidden, if they share certain attributes with the current solution x_t. Specifically, a tabu list records attributes associated with recently evaluated solutions to avoid selecting the same or similar solutions in subsequent iterations. It operates on the premise that effective problem-solving must integrate adaptive memory and responsive exploration. The adaptive memory in TS enables efficient and targeted exploration of the solution space, as decisions are informed by knowledge accumulated throughout the search process. This stands in contrast to memoryless approaches, which rely on semi-random sampling methods to explore potential solutions. The focus on responsive exploration stems from the notion that a strategically poor choice can often provide more valuable insights than a randomly good one [85].

ILS, as used in Baxter [86], is a metaheuristic that iteratively applies a local search algorithm until a predefined stopping criterion is met. Once this criterion is reached, a perturbation is introduced to the best solution obtained, altering the initial solution for the next iteration. This process allows for exploring different neighbourhoods, reducing the likelihood of cycling.

VNS, proposed by Mladenović and Hansen [87], prevents cycling by systematically alternating between different neighbourhood structures $\mathcal{N}_1, \ldots, \mathcal{N}_p$. These neighbourhoods are often organised using increasingly complex perturbation rules, ensuring that the algorithm employs more distant neighbourhoods to explore the solution space more comprehensively as the search progresses. The process begins with an initial solution, and the algorithm iteratively explores these neighbourhoods through local searches. When an improved solution is found, the current solution is updated to this new best solution, and the Neighbourhood structure is reset to the simplest form. The algorithm stops after a preset number of iterations or when no further improvements are possible.

Population-Based Methods are inspired by natural concepts such as species' evolution and insects' behaviour. Population-based algorithms manage a diverse set of solutions and use population-wide interactions to explore and exploit the solution space more broadly, in contrast with the local moves employed in local search methods. Most population-based algorithms implement some local search heuristic, making them inherently hybrid because they borrow concepts from other heuristics.

ACO, proposed by Dorigo et al. [37], is inspired by the real foraging behaviour of ants, which communicate by laying pheromone trails on the paths they traverse to guide other ants in finding shorter paths, as they tend to follow routes with higher pheromone concentrations. In ACO, a similar mechanism is implemented, which deploys virtual pheromones to represent the solution quality. New solutions are generated using a savings-based procedure and local search. In addition to the classic savings definition as described in subsection 5.1.2, where the savings for combining two routes are calculated using $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, an attractiveness value $\chi_{ij} = \tau_{ij}^{\alpha} - s_{ij}$ is used, where τ_{ij} represent the pheromone value to describe how good combining vertices i and j was in previous iterations. Furthermore, α and β are user-controlled parameters . The combination of vertices i and j take place with a probability $p_{ij} = \chi_{ij} / \left(\sum_{(h,l) \in \Omega_k} \chi_{hl} \right)$, with Ω_k the set of feasible (i,j) combinations that yield the best k savings [29].

GAs, developed by Holland [88], are evolutionary algorithms based on the process of natural selection. They operate on a population of potential solutions called chromosomes, which evolve over successive generations. The algorithm applies three key operators: selection, which chooses the fittest individuals from the current population to serve as parents; crossover (recombination), which combines the genetic material of two parents to produce offspring; and mutation, which introduces random changes to individual chromosomes to maintain diversity within the population.

Scatter Search (SS) and Path Relinking (PR) are metaheuristic techniques that combine solution recombination and local improvement as complementary methods. SS was originally proposed by Glover [89] and is an evolutionary metaheuristic that operates on a small set of high-quality solutions and explores solution spaces while using only limited randomisation in the diversification process. This is because of the notion that a structured and more purposeful approach could provide significant benefits. PR, originally proposed by Glover [90], is an intensification strategy considered an extension of the diversification method used in SS. It begins with one or more starting solutions and systematically explores the path to an end solution, generating paths between and beyond these solutions in the neighbourhood space [91]. During this exploration, intermediate solutions are evaluated and incorporated into the solution set if they perform better than the current solutions. PR is frequently integrated with other metaheuristics to enhance the intensification process.

5.2 Proposed Tabu Search Implementation

As explained earlier in Section 3.5, the VRPCD is an extension of an already $\mathcal{NP}$-hard problem, justifying the use of heuristic methods to find near-optimal solutions in feasible CPU times.

TS has become widely recognised in academic research for its effectiveness and flexibility [85] in solving various VRPs, including the VRPCD, as detailed in the literature search in Section 3.6 and further emphasized by Wang et al. [92]. This study adopts a TS-based approach to tackle a specific version of the VRPCD that incorporates customer time windows, multiple product types, and a variable number of vehicles, while also allowing direct supplier-retailer trips (under the condition that only one supplier-retailer pair is visited in the route). The complexities inherent in this problem necessitate a flexible optimisation method, which is a key advantage of TS. As noted by Glover et al. [85], the strategic use of memory within TS can significantly enhance problem-solving capabilities in virtually any optimisation problem. This allows it to effectively model the specific intricacies of the problem studied here.

TS is a metaheuristic that guides a local search heuristic that iteratively moves from a current solution S to the best admissible solution S' in a subset $\Omega_N(s)$ of a neighbourhood N, using some local search heuristic. Cycling back to previously visited, or tabu, solutions is prevented (exceptions are provided by aspiration criteria) by the use of a memory, which stores attributes of these solutions in short-term memory (i.e. the tabu list) for a predefined number of iterations θ [68]. TS has shown promising results in VRPCD resolution and is known for its flexibility in adjusting to any optimisation problem [85].

The procedure improves an existing solution, thus requiring an initial solution to be generated. In this study, a greedy insertion heuristic provides this initial solution, which is outlined in Section 5.2.1.

The algorithm presented in Algorithm 1 provides a general outline of the TS method, highlighting the roles of its essential components. This section details these basic algorithmic components of the TS method: initial solution, neighbourhood structure, tabu list and aspiration criteria, and stopping criteria [69]. Sections 5.2.1 - 5.2.4 discuss these components of the proposed implementation, respectively.

Algorithm 1: Tabu search algorithm

```
Data: Initial solution and parameters                          ▷ See Section 5.2.1 and 6.2
Result: Current best solution found
Initialise: TabuList ← ∅, Solution ← ConstructInitialSolution, CurrentBest ← Solution,
            AddElement(Solution, TabuList)
while NOT stopping_criteria do                                 ▷ See Section 5.2.4
    Neighbourhood ← GenerateNeighbourhood(Solution)            ▷ See Section 5.2.2
    BestNeighbour ← ChooseNeighbour(Neighbourhood, TabuList)   ▷ See Section 5.2.3
    Solution ← BestNeighbour
    if Solution < CurrentBest then
        CurrentBest ← Solution
    if Length(TabuList) ≥ TabuListLength then
        RemoveOldestElement(TabuList)
    AddElement(Solution, TabuList)
return CurrentBest
```

5.2.1 Initial Solution Generation

The constructive heuristic enables initial solution generation with minimal computational effort and utilises a two-stage greedy insertion method adapted from the VRPTW context in Solomon [93]. For the interested reader, Appendix C defines the solution generation process in detail.

5.2.2 Neighbourhood Structures

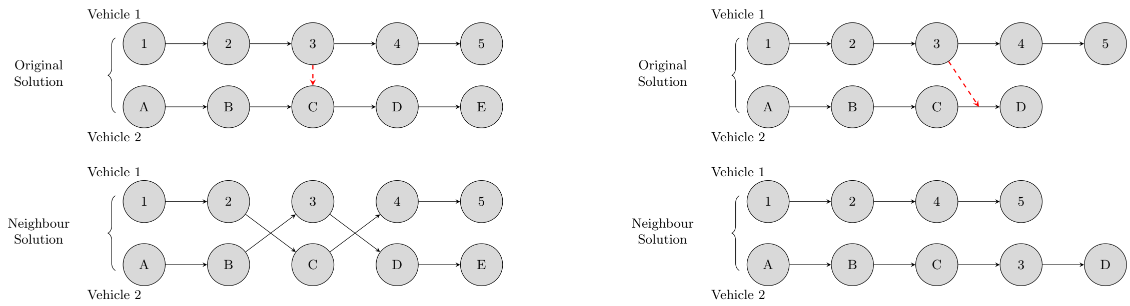
As indicated in Section 5.2, neighbouring solutions are iteratively generated to transition from the current solution S to the best admissible solution S' within a subset $\Omega_N(S)$ of the neighbourhood $N(S)$. Neighbourhood generation is, therefore, a critical component of the solution approach. This section details the neighbourhood generation process and the structures employed in this study.

In the literature, pairwise exchange — referred to as the SWAP strategy in this study — is one of the most commonly applied methods for transitioning between solutions [69, 94]. This local search technique generates a new solution by swapping the nodes at positions i and j. For a J-city problem in a TSP, the total number of possible swaps, or neighbours, is $[N(s)] = J(J-1)/2$ [1]. Another widely used approach is extraction and reinsertion, called the INSERTION strategy here. This method generates a neighbourhood by removing the node at position i and inserting it immediately before or after the node at position j, resulting in a neighbourhood size of $[N(s)] = J(J-1)^2$ [2], which is significantly larger and computationally more demanding than the SWAP strategy. For an illustration of the local search strategies, see Figure 5.1.

Despite the larger neighbourhood size, the INSERTION strategy should, however, not be overlooked. As noted by Adenso-Díaz [95], in the context of the TSP, neither SWAP nor INSERTION consistently outperforms the other in terms of solution quality relative to runtime. However, in the context of the problem studied in this research, INSERTION has the distinct advantage of being able to reduce the number of vehicles required, as opposed

[1] The total number of possible swaps in a J-city problem is given by $\binom{J}{2} = \frac{J(J-1)}{2}$.

[2] The total number of possible insertions in a J-city problem is given by $J(J-1)^2$ because each node can be extracted and inserted in $J-1$ possible positions for every other node.



(A) SWAP operator

(B) INSERTION operator

FIGURE 5.1: Example of the neighbourhood operators SWAP (a) and INSERTION (b) for generating a neighbour solution

to the SWAP strategy. Therefore, to improve solution quality in our case, both the SWAP and INSERTION strategies are employed interchangeably. Additionally, it is important to note that the local search strategies are applied both inter-route and intra-route depending on the most gain achieved by performing the operation.

In the implementation proposed in this research, each iteration of the neighbourhood search begins by classifying vehicles based on their role in the current solution, distinguishing between those that only pick up, only deliver, perform both operations, or remain idle. This classification ensures that appropriate operations can be applied based on vehicle function. Instead of generating the complete neighbourhood, a subset of vehicle routes is sampled, and only for these selected routes is a neighbourhood constructed. This approach balances exploration and computational efficiency. Valid operations (either swap or insertion) are precomputed to ensure that modifications maintain feasibility based on the vehicle's route type and whether it interacts with suppliers, retailers, or a CD. For instance, a supplier cannot be inserted into a delivery route.

After neighbourhood generation, the best admissible solution is selected as the next current solution, and the process repeats itself. Note that the best admissible solution is chosen from the generated neighbourhood of the sampled routes rather than from the complete neighbourhood of the solution $N(s)$, as generating all possible neighbours would be computationally infeasible. Additionally, some solutions may be marked *tabu*, further restricting the search space. This selective neighbourhood exploration aligns with strategies such as those used by Esfahani and Fakhrzad [69], where instead of examining all neighbourhoods, a candidate list is formed based on vehicles with the highest number of nodes.

5.2.3 Tabu List and Aspiration Conditions

In a VRP, storing entire solutions in the tabu list would be computationally intensive. Instead, specific attributes or characteristics of the solutions are stored to mark certain moves or changes as tabu. These attributes reflect key modifications to the current solution that the search should avoid for a defined number of iterations, θ, thereby helping to prevent cycling or revisiting similar solutions. Specifically, when a node is removed from a vehicle, reinserting it into that vehicle is prohibited for θ iterations, similar to the approach adopted by other TS implementations [63, 96]. Such moves are referred to as *reverse moves* because a customer moved from route r to route r' at iteration t may be prohibited from being reinserted in route r (until iteration $t + \theta$). This forms the recency-based, or short-term, memory of the implementation proposed in this study.

In our implementation, we adopt two separate tabu lists to facilitate a recency-based memory for reverse moves of SWAP and INSERT operations. To this end, every iteration's best candidate move is saved to its corresponding tabu list for θ iterations, which is independent of SWAP/INSERT selection probabilities.

Some algorithms use a fixed value of θ [97], whereas [98] suggests randomly selecting θ in an interval $\left[\underline{\theta}, \overline{\theta}\right]$, according to a discrete uniform distribution. The latter is the strategy adopted in this implementation.

An aspiration criterion exists to revoke a move's tabu status if this causes no risk of cycling. In our case, this yields a better overall incumbent feasible solution.

5.2.4 Stopping Criteria

In theory, the search could go on forever unless the optimal value of the problem is known beforehand. In practice, however, the search has to be stopped at some point, as defined by some stopping criteria. Common stopping criteria in tabu search that are also used in our implementation include [85, 99]:

- After a fixed amount of CPU time
- After some preset consecutive iterations without improving the objective function value (the criterion used in most implementations).

In this study, we employ two stopping criteria: limiting the CPU time and terminating the algorithm after a predefined number of consecutive iterations without improvement in the objective function. The CPU time limit criterion ensures that the algorithm is predictable in its running time. However, it should be noted that strictly limiting this criterion increases the chance of ending with a solution far from optimal. The stopping criterion related to the number of consecutive iterations without an improvement is adopted to prevent excessive, unnecessary CPU time when convergence has already been achieved. This stopping criterion is referred to as the patience of the algorithm and should be chosen tactfully to prevent unintended termination before reaching convergence.

While the proposed TS implementation explores the solution space using memory-based strategies, it remains susceptible to premature convergence when the best-found objective value stabilises. In other words, the risk of getting stuck in a local optimum remains significant. An adaptive SA mechanism is embedded to address this, activating when the search stagnates. This hybrid approach combines SA's probabilistic acceptance as a diversification mechanism while maintaining the advantages of TS' memory-based exploration of the solution space. The hybridisation of these methods is detailed in Section 5.3.

5.3 Adaptive Simulated Annealing Integration in Tabu Search

The main problem of TS, despite the beneficial impact of tabus, is that they tend to spend most or all time in a restricted portion of the search space [100]. Therefore, many adaptations of TS may fail to explore all interesting parts of the search space and end up with a solution far from the global optimum [100]. In addition to the aspects of the solution method outlined in Section 5.2, this section introduces an embedded diversification mechanism designed to alleviate this issue by guiding the search toward previously unexplored areas of the solution space. This integration dynamically employs SA, activating it only when the reduction in the best-found objective function stagnates beyond a predefined threshold.

5.3.1 Activation and Stagnation Detection

Diversification is a mechanism designed to explore restricted portions of the search and must be carefully balanced with intensification to ensure an effective trade-off between exploration and exploitation. In the adaptive TS-SA approach, rather than being applied continuously, diversification is selectively triggered when progress slows down for a sustained period. The activation condition is designed to detect stagnation in the search to ensure that alternative regions of the search space are only explored when necessary. Specifically, diversification is only activated if the stagnation condition is met for at least M consecutive iterations. With f the objective value, the stagnation detection criterion is defined as:

$$\frac{|f_{\text{best},t} - f_{\text{best},t-1}|}{|f_{\text{best},t-1}|} < \tau \tag{11}$$

for M consecutive iterations, where:

- $f_{\text{best},t}$ is the best objective value at iteration t,
- $f_{\text{best},t-1}$ is the best-found objective value at iteration $t-1$,
- τ is a predefined stagnation threshold,
- M is the minimum number of consecutive iterations for which the stagnation condition must hold before SA is triggered.

This approach prevents premature activation of an SA phase due to small fluctuations in objective value improvements. Additionally, by requiring the stagnation condition to hold for at least M consecutive iterations, it introduces a control mechanism to ensure that the SA phase is only invoked when the search is truly stagnating.

5.3.2 Temperature Cooling Schema

While SA inherently follows a Markov Chain process due to its probabilistic acceptance mechanism, this property is violated when SA is embedded within TS since the memory structure in TS retains historical information through tabu restrictions (i.e. move acceptance is influenced not only by the current state but also by past decisions). As a result, the hybrid approach no longer satisfies the Markov property since future transitions are constrained by the memory structure rather than being conditionally independent of past states. Consequently, the length of the SA phase represents an adaptive diversification period rather than an independent Markov process.

Unlike traditional SA, where temperature follows a uniformly decreasing schedule, the embedded diversification phases employ a two-fold cooling mechanism:

1. Internal cooling factor (α_{int}): Guides the temperature decay *within* each activation of SA.
2. External cooling factor (α_{ext}): Gradually reduces the initial temperature T_0 *across* multiple activations.

The initial temperature at the start of an SA phase is determined by an external cooling schedule to let successive SA activations begin with progressively lower initial temperatures:

$$T_0^{(k)} = T_0 \cdot (\alpha_{\text{ext}})^k \tag{12}$$

where:

- T_0 is the original starting temperature,
- α_{ext} is the external cooling factor,
- k represents the number of times SA has been activated.

Within an active SA phase, the actual temperature at iteration t is derived from the externally determined initial temperature and follows an internal cooling schedule:

$$T^{(t)} = T_0^{(k)} \cdot (\alpha_{\text{int}})^t \tag{13}$$

where:

- α_{int} is the internal cooling factor,
- t represents the iteration count within the current SA activation.

In conclusion, each SA activation starts with an externally controlled temperature $T_0^{(k)}$, which then gradually decreases within the SA phase according to the internal cooling schedule. Figure 5.2 depicts an example of the evolution of the temperature over the iterations of a trial of the algorithm.

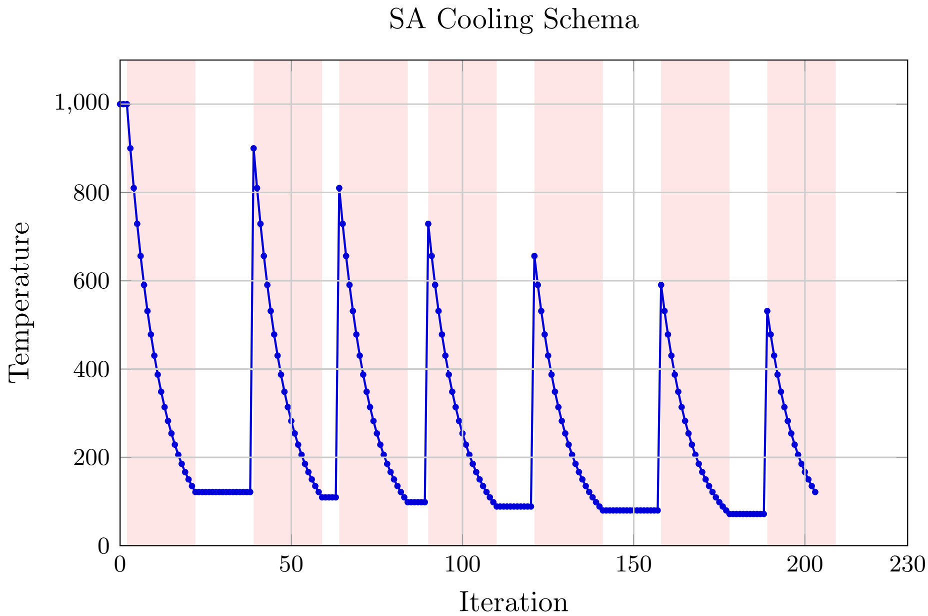


FIGURE 5.2: Temperature cooling schema with $T_0 = 1000$, $\alpha_{\text{int}} = 0.9$, and $\alpha_{\text{ext}} = 0.9$.

5.3.3 Probabilistic Move Acceptance and Tabu Constraints

During an active SA phase, candidate solutions are accepted probabilistically based on Equations 9 and 10. However, since SA is nested within TS, reverse moves from the SA phase are declared tabu, ensuring that the search does not immediately return to previously visited solutions.

5.3.4 Balancing Intensification and Diversification

The integration of TS and SA balances intensification and diversification, leveraging the strengths of both methods:

- TS ensures structured exploration by guiding the search through adaptive memory.
- SA introduces controlled diversification to prevent stagnation and escape local optima.

By activating SA only when necessary, the hybrid approach maintains the efficiency of memory-based approaches while allowing for flexibility in the exploration of probabilistic acceptance methods. The combination of TS and SA improves the search process by balancing the exploitation of high-quality solutions and exploring new areas when progress stagnates.

5.4 Findings and Implications

6. *What solution methods exist for enhancing vehicle routing decisions at Wolter Koops, and which is most suitable for implementation?*

This chapter explores various solution methods for the VRP in the context of Wolter Koops's cross-docking operations. Solution approaches are explored using exact, heuristic, and metaheuristic methods. While exact methods offer optimal solutions, their computational infeasibility for large-scale instances renders them impractical for real-world routing problems. Consequently, heuristic and metaheuristic methods are more widely adopted because they generate near-optimal solutions within reasonable computational time frames.

The hybrid TS-SA solution approach integrates TS and SA to balance intensification and diversification in the search process. This hybrid approach leverages the memory-based nature of TS while incorporating SA-based diversification to escape local optima, improving solution robustness. The method effectively addresses the complexities of the VRPCD and incorporates time windows, multi-compartment vehicle constraints, and consolidation at the CD.

The findings suggest that metaheuristic approaches are best suited for implementation at Wolter Koops due to their computational efficiency and ability to handle large-scale problem instances. Moreover, TS is highly generalisable, as it can be applied to virtually any optimisation problem. The selected approach significantly improves cost efficiency and routing feasibility compared to manual planning. However, fine-tuning key parameters, such as the tabu tenure, cooling schedule, and diversification mechanisms, remains crucial for optimal performance.

Thus, while other metaheuristic approaches could also be viable alternatives, the TS-SA hybrid is identified as the most suitable approach for implementation, given its adaptability. Furthermore, the TS metaheuristic has shown promising results in similar applications (see Section 3.6). Future work could explore hybridisation with other heuristics and real-time dynamic re-optimisation techniques to enhance scalability and responsiveness in practical applications.

6 Model Validation and Experimental Design

This chapter aims to contribute to Phase 4: Modelling and Implementation by detailing the input data and filtering process, the model optimisation process, implementation of the case study, and the experimental setup. The research questions of this phase are:

7. *How can the input data be processed and validated to ensure the model operates reliably?*
8. *What key parameters influence the model's performance, and how can they be effectively tuned?*
9. *How can the model be evaluated in a case study to assess its applicability to real-world routing scenarios?*
10. *What experimental setup is required to systematically test the model's performance and robustness?*

These research questions provide the basis for executing the experiments to test the model's performance and robustness. Chapter 7 then presents the experimental design results.

Section 6.1 presents the input data and filtering process used to validate the model in a case study. Section 6.2 presents the parameter tuning process for optimising the model. Section 6.3 details the case study validation, where the algorithm is tested by comparing its best-found solution with historical routes. Section 6.4 defines the experimental framework used to test the algorithm's performance and robustness. These results are presented in Chapter 7.

All runs were conducted on a high-performance computing server with an AMD EPYC 9534 64-Core Processor (128 threads) and 1024GB RAM.

6.1 Input Data and Filtering

The input dataset comprises historical shipment records from Wolter Koops, explicitly focusing on shipments within the EG group. These shipments involve the transportation of plants and flowers from the Westland region in the Netherlands to various destinations in Germany, utilising CD facilities in Venlo and Osterweddingen. However, to simplify the analysis and improve tractability, shipments passing through Osterweddingen were excluded. This decision was based on the clear geographical distinction between shipments routed via Venlo and those passing through Osterweddingen.

Additionally, shipments passing through alternative locations such as Alzenau and Zeewolde were excluded to maintain consistency in the dataset. The data spans two consecutive days, January 20, 2025 and January 21, 2025, and includes the attributes outlined in Table 6.1. The attributes were chosen carefully to ensure that historical routes can be reproduced to compare the algorithm's performance with the historical planning. Historical trips are defined by the *Type action*, *Shipment number*, *Trip number* and *Sequence number* attributes. Algorithm performance is assessed by comparing the cost and load efficiency of the algorithm's planning with the historical route planning KPIs, with the gap between the two serving as a performance measure.

TABLE 6.1: Description of Input Data Attributes

Attribute	**Description**
Type action	Specifies the type of action (i.e. load or unload).
Shipment number	Unique identifier for each shipment.
PTA start	Planned Time of Arrival start/start time window.
PTA end	Planned Time of Arrival end / end time window.
Latitude	Latitude coordinates of the location.
Longitude	Longitude coordinates of the location.
Total pallet places	Total number of pallet spaces required for the shipment.
Total weight	Total weight of the shipment in kilograms.
Temperature	Temperature requirements for the shipment (if applicable).
Trip number	Unique identifier for the trip associated with a shipment.
Location number	Identifies the specific location within the route.
Sequence number	Determines the order of visits within a trip.
Cross dock	Indicates if the location specified is a CD.

The initial dataset consisted of 1,575 records, although inconsistencies and irregularities required a filtering process to ensure data compatibility for analysis. The raw data contained discrepancies in pallet counts, where the number of unloaded pallets at retail locations exceeded that of loaded pallets at supplier sites, and shipments with more than four distinct actions.

The data filtering process involved several sequential steps to refine the dataset and make it consistent with operational constraints. The filtering process is outlined below in preparation for running the model described in Chapter 5. All specific numbers can be found in Appendix D for reproducibility.

1. Removal of records with missing values: Entries lacking essential attributes, including *Type action*, *Shipment number*, *PTA start*, *PTA end*, *Latitude*, *Longitude*, *Total pallet places*, *Total weight*, *Temperature*, *Trip Number*, *Location number*, *Sequence number*, or *Cross dock*, were eliminated.

2. Exclusion of trips passing through specific locations: Trips travelling through pre-specified locations were removed based on operational considerations. Due to confidentiality, exact location names are not documented, but the filtering was applied to:

   - A location with no known operational relevance.
   - A site identified as a third-party logistics facility near Bremen.
   - A location along a route no longer used in the study, ensuring shipments were routed exclusively through Venlo.
   - An alternative location near Venlo, which was either converted into a CD site or removed to maintain pallet balance ($\sum$ loaded $= \sum$ unloaded).

3. Retention of only relevant trips: The dataset was filtered only to include historical trips containing an action at the CD, either at the start of the trip or at the end. This step is required to ensure all trips in the input data pass through the CD.

4. Exclusion of trips with specific vehicle IDs: Vehicles that arrived too late in the validation process were removed from the dataset. These vehicle IDs included:

- 22525414
- 22524722
- 22524777
- 22500900

5. Exclusion of records with identical PTA start and PTA end: Shipments with identical PTA start and end were removed, as they often represented actual arrival times rather than valid scheduling data, leading to potential inaccuracies.

Following these filtering steps, the dataset is reduced to 307 records, which forms the final input dataset for calculating benchmark values and solutions in subsequent analyses. The significant reduction in the dataset is due to removing 835 records, as all orders were required to visit the CD in Venlo, the Netherlands. Additionally, historical trips that arrived outside the designated time windows were excluded to ensure a fair comparison of the model. Furthermore, 433 records were deleted because of data issues.

6.2 Parameter Tuning

This section presents the parameter tuning process for the heuristic algorithm introduced in Chapter 5. Parameter tuning aims to determine the optimal configuration of algorithmic parameters to enhance performance on a given problem set. The effectiveness of heuristic algorithms is highly sensitive to parameter selection, making tuning essential for achieving improved solution quality and computational efficiency [101].

Hoos [102] defines parameter tuning as the process of identifying an optimal configuration c^* within a parameter space C, given:

- An algorithm A with parameters $p_1, \ldots, p_k$ that influence its behaviour,
- A configuration space C specifying possible values for these parameters,
- A problem instance set I,
- A performance metric m measuring the algorithm's effectiveness on I under a given configuration c.

The goal is to determine c^* that maximises algorithmic performance on I according to m. Furthermore, we define the tunable parameters $p_1, \ldots, p_k$ that influence the algorithm's behaviour. The network setting parameters, including the number of vehicles, their capacity, loading and unloading times, and costs, are non-tunable as operational constraints at Wolter Koops determine them and cannot be altered without deviating from real-world conditions. Furthermore, we differentiate between TS and SA parameters. A summary of all non-tunable and tunable parameters are summarised in Table 6.2. The values for the fixed network settings are based on estimations resulting from meetings with Wolter Koops and CAPE Groep. It is important to note that, unless explicitly stated otherwise, these values remain the same throughout all optimisation trials and scenarios defined by the experimental design (Section 6.4).

TS parameter settings guide the meta-heuristic on a higher level, and include the stopping criteria (maximum execution time of the algorithm t_{max} and patience P; see Section 5.2.4 for more information), and minimum and maximum tabu tenure settings (see Section 5.2.3).

SA parameter settings denote the diversification settings, which include the SA activation criterion M, SA iterations per activation N, and the reduction threshold τ. For more details on stagnation and SA activation, refer to Section 5.3.1. Furthermore, initial temperature T_0 and the internal and external cooling factors α_{int} and α_{ext} are required (see Section 5.3.2 for the temperature cooling schema parameters).

TABLE 6.2: Parameter Notation and Configuration Space

Parameter Notation and Configuration Space				
Network Settings (Fixed)		*Tabu Search (TS) Settings (Tunable)*		*Range*
Number of Vehicles	$V = \infty$	Max Execution Time (s)	$t_{\max}$	-
Vehicle Capacity	$Q = 33$	Patience	P	-
(Un)loading Time	$S = 5$	Min Tabu Tenure (Swap)	$\underline{\theta}^{swap}$	$[5, 40]$, step 1
Variable Cost per min	$c_v = 1$	Max Tabu Tenure (Swap)	$\overline{\theta}^{swap}$	$[\underline{\theta}^{swap} + 1, 50]$, step 1
Fixed Vehicle Cost	$c_f = 1000$	Min Tabu Tenure (Insertion)	$\underline{\theta}^{ins}$	$[5, 40]$, step 1
		Max Tabu Tenure (Insertion)	$\overline{\theta}^{ins}$	$[\underline{\theta}^{ins} + 1, 50]$, step 1
		Simulated Annealing (SA) Settings (Tunable)		*Range*
		SA Activation Iterations	N	$[1, 20]$, step 1
		Reduction Threshold	τ	$[0, 0.07]$, step 0.005
		SA Phase Duration	M	$[1, 30]$, step 1
		Initial Temperature	T_0	$[500, 3500]$, step 100
		Cooling Factor (External)	α_{ext}	$[0.7, 0.95]$, step 0.01
		Cooling Factor (Internal)	α_{int}	$[0.7, 0.95]$, step 0.01

While traditionally, parameter values have been set manually using expertise and experimentation, recently, several automated tuning methods have been proposed [101]. In this implementation, we adopt the Optuna parameter-tuning software [103]. Optuna is an open-source parameter optimisation framework that automates the tuning process using efficient search algorithms like Bayesian optimisation, Tree-structured Parzen Estimator, and Hyperband. Optuna is embedded in the algorithm A with a configuration space $\mathcal{C}$, where each configuration $c \in \mathcal{C}$ is defined by the values of the parameters. The possible values of each parameter are based on experience developing the algorithm and are shown in Table 6.2.

The Optuna parameter-tuning software can easily integrate the adaptive TS-SA hybrid solution approach, although the running time per trial needs to be determined. Because there is no strict maximum running time, we estimate the time until convergence is expected to be achieved. For this, we perform a trial of sufficient length to estimate the needed time (we ignore the patience parameter to not risk early termination). Because a trial $t \in \mathcal{T}$ is a run of algorithm A with a given configuration, we estimate reasonable values of the parameters within their respective ranges. This eight-hour trial is displayed in Figure 6.1, and demonstrates that convergence is achieved in approximately 3.5 hours. Therefore, the maximum running time per trial is set accordingly in the Optuna parameter-tuning process.

The parameter tuning process consists of 101 trials, where the best-found objective value is 21,964, found in trial 68. An initial objective value of 23,745 was achieved in the case with estimated parameter values. Thus, parameter tuning decreased these costs by another 7.5%. The parameter tuning process was carried out until convergence of the best-found objective value was observed, which occurred after 68 trials. The parameter tuning history

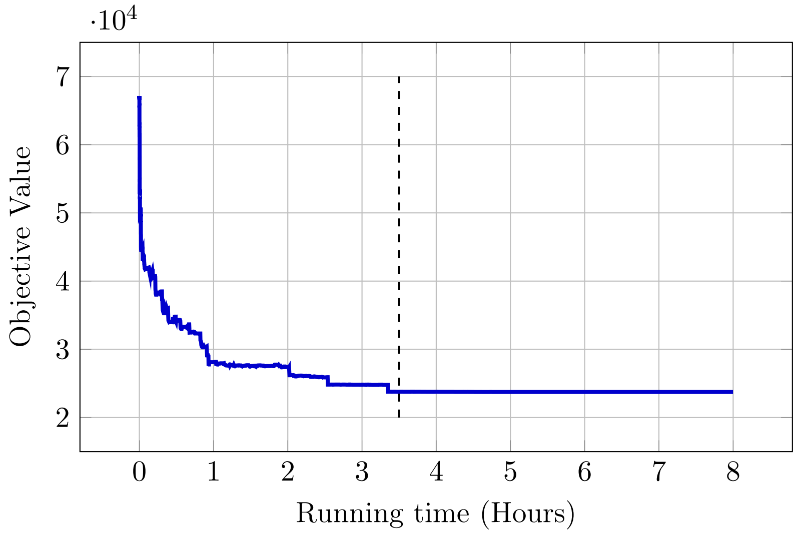


FIGURE 6.1: Objective Value Evolution over the Running Time of Experiment 1

is summarised in Figure 6.2, which highlights the effect of parameter tuning on the objective value. In the worst-case scenario, overly restrictive diversification activation rules cause the algorithm to become trapped in a local optimum early in the search, preventing the TS algorithm from effectively diversifying (see trials 16 and 41 in Figure 6.2).

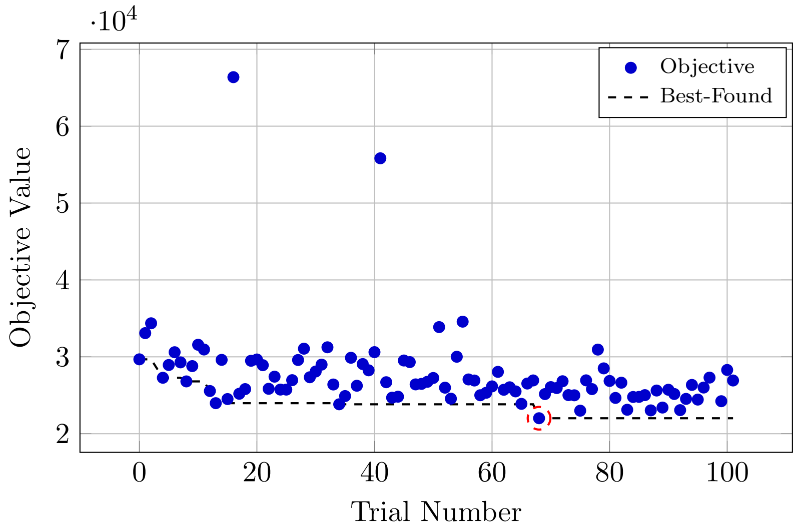


FIGURE 6.2: Optimisation History of the Optuna Parameter Tuning Process.

Figure 6.3 denotes the importance of tuning individual parameters. The external cooling factor (α_{ext}) is identified as the most influential parameter, with an importance score of approximately 0.45. This parameter guides the cooling schedule outside the SA phase, affecting the algorithm's ability to balance exploration and exploitation. A higher α_{ext} results in a slower decrease in initial temperatures, allowing for more extensive exploration but potentially delaying convergence. Conversely, a lower α_{ext} accelerates cooling, which may lead to premature convergence to a local optimum. The SA Activation Length (M) is the second most critical parameter, with an importance of 0.32. This parameter dictates when the SA phase is triggered, making it a key mechanism for introducing diversification at the appropriate moments. If M is too low, SA is invoked too frequently, disrupting intensification and potentially leading to inefficient searches. Conversely, if M is too high, the algorithm may risk stagnation in a local optimum before SA can provide effective diversification.

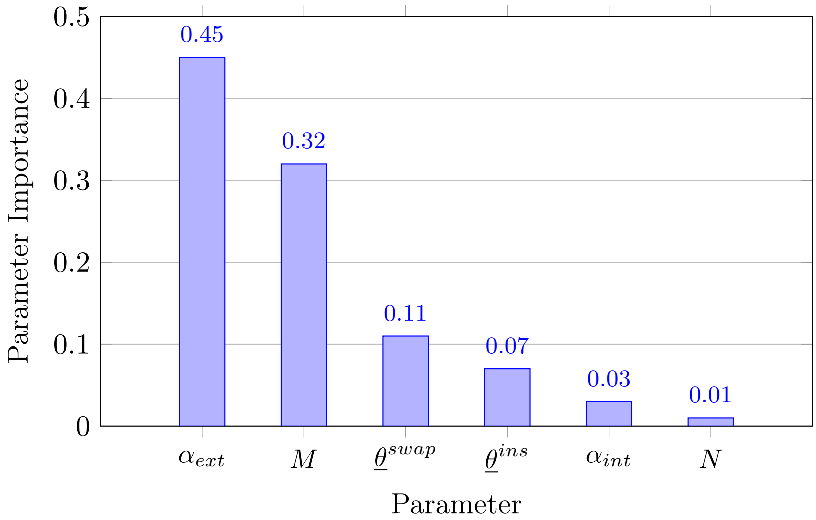


FIGURE 6.3: Parameter Importance according to Optuna

Although Optuna has accounted for parameter importance in the tuning process, future applications of the algorithm that require re-tuning should pay particular attention to α_{ext} and M. Given their substantial impact on algorithm performance, computational resources should be incentivised to tune these parameters over other parameters. Therefore, careful manual or automated fine-tuning of these two parameters is recommended in any future adaptation of the method to different problem instances.

The best trial achieved an objective value of $21,964$ with the configuration in Table 6.3. Because this configuration effectively balances diversification and intensification within the 3.5 hour execution time per trial, these values remain throughout all scenarios and analyses defined by the experimental design (Section 6.4).

TABLE 6.3: Optimised Parameter Configuration

Tabu Search (TS) Settings	
Minimum Tabu Tenure (Swap)	16
Maximum Tabu Tenure (Swap)	35
Minimum Tabu Tenure (Insertion)	5
Maximum Tabu Tenure (Insertion)	12
Simulated Annealing (SA) Settings	
SA Activation Iterations (N)	1
Reduction Threshold (τ)	0.04
SA Activation Duration (M)	14
Initial Temperature (T_0)	2600
Cooling Factor (External) (α_{ext})	0.91
Cooling Factor (Internal) (α_{int})	0.88

6.3 Case Study Validation

The input data used to validate the algorithm contains historical routes for a given set of orders through the CD in Venlo. This allows for the direct comparison of historical routes and the optimality of generated routes. The goal of this section is, therefore, to set a benchmark in algorithm performance. Consequently, we present the procedure for evaluating the historical routes, present the algorithm's solution, and briefly discuss the comparison. The evaluation criteria used to measure both solution qualities are the cost, loading efficiency, and service level (see Section 6.4.2)./hl Note that service levels are only reported in this thesis if they fall below 100%.

The historical planning forms the starting point for comparing the historical route planning versus the solution of the hybrid TS-SA algorithm. Historical routes can be reconstructed from the input data by grouping the records by trip number and considering the sequence of performed actions per location. More analysis on the historical routes reveals that the cost of the historical route planning (according to the objective function (8a) and the parameters defined in Table 6.2) are €32,199 and 70.7%, respectively. After analysing the historical routes, the algorithm is executed for 3.5 hours using the optimised configuration (Table 6.3). Figure 6.4 summarises the objective value evolution over the running time. The solution converges rapidly within the first hour, achieving over 75% of the total cost reduction within this period. At this point, the objective value is already approximately equal to that of the historical planning benchmark, suggesting that a shorter runtime could still yield substantial improvements while maintaining competitive solution quality.

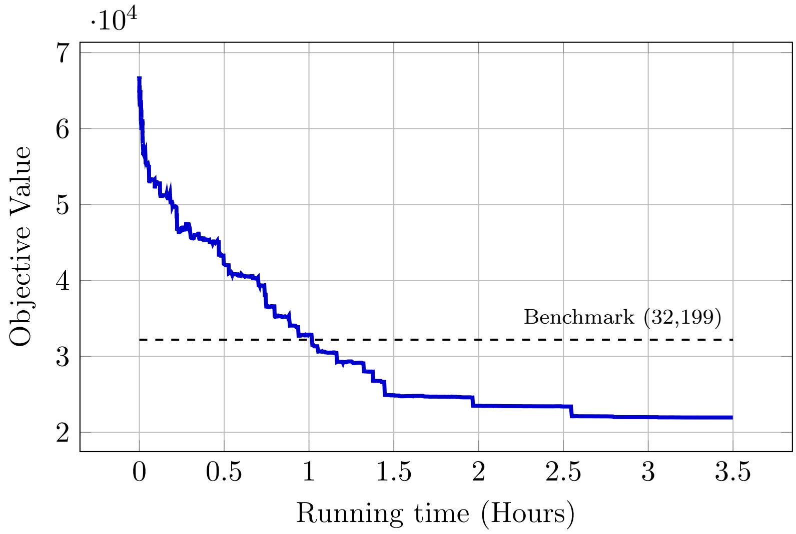


Figure 6.4: Objective Value Evolution

After executing the algorithm for 3.5 hours using the optimised parameters (see Table 6.3), the model achieves an objective value of 21,964. Compared to the historical planning, the number of vehicles required decreases from 22 to 12, contributing to a 31% reduction in the objective value. Meanwhile, the total travel time remains nearly unchanged, decreasing marginally from 10,199 to 9,964 minutes, resulting in only a 0.7% impact on the objective function. A 6.1% increase in loading efficiency suggests that fewer vehicles are required to transport the same orders, reducing fleet utilisation pressure.

These findings demonstrate the model's potential to enhance cost efficiency and fleet util-

TABLE 6.4: Comparison of Historical Planning and Hybrid TS-SA Model Performance

Scenario	Total Cost (€)	Cost Gap (%)	Loading Efficiency (%)	Efficiency Gap (%)
Historical Planning	32,199	–	70.7	–
Hybrid TS-SA Model (Baseline)	21,964	**-31.8**	76.8	+**8.1**

isation. However, since input parameters such as fixed and variable costs influence the algorithm's performance, future research could investigate robustness across varying operational settings.

6.4 Experimental Design

The following describes the experimental design to evaluate the effectiveness of the proposed algorithm. Using the configuration that follows from parameter tuning (see Table 6.3), the experimental design consists of test scenarios, sensitivity analyses on input parameters, and a description of the evaluation criteria that determine the quality of a solution. Altogether, these experiments determine the effectiveness and robustness of the proposed algorithm and provide insights into realistic problems.

6.4.1 Scenarios and Sensitivity Analysis

In addition to the base scenario, different scenarios are analysed to verify the algorithm's performance under different problem variations. Furthermore, they provide insights into routing intricacies and result from meetings with Wolter Koops [4, 5]. The following introduces these scenarios, which are conducted in Sections 7.1.1 and 7.1.2.

Scenario 1: Algorithmic Variation

This analysis evaluates the impact of an alternative fixed cost definition, where fixed costs are incurred each time a vehicle departs from the CD, rather than applying a single fixed cost per vehicle usage. The base scenario assumes that each vehicle incurs fixed costs only once per planning horizon, based on the premise that chartered vehicles must be available in advance. However, when vehicles are not leased in advance, a cost model that charges fixed costs each time a vehicle departs from the CD might better reflect the operational reality. To the best of our knowledge, this definition of the fixed cost is aligned with other implementations of fixed costs in the VRPCD context found in the literature [104].

Scenario 2: Impact of Compartmentalisation

This analysis evaluates whether the added complexity of incorporating multiple compartments in vehicle routing is justified in terms of operational efficiency and cost-effectiveness. While multi-compartment vehicles offer flexibility by allowing mixed-temperature shipments, they introduce additional constraints that increase computational complexity, routing challenges, and fleet management challenges [29]. By comparing the single-compartment and multi-compartment scenarios, this study assesses whether the benefits of compartmentalisation outweigh the potentially higher routing complexity and computational burden.

Sensitivity Analysis

To evaluate the robustness of the proposed optimisation model, a series of sensitivity experiments are conducted. Table 6.5 summarises the experimental scenarios, categorised by cost variations in euros (c_{var} and c_f), travel speed adjustments in km/h (v), and time window modifications in minutes (ΔTW). Each experiment is indicated using its parameter configuration using $E\left(c_{var}, c_f, \Delta TW, v\right)$

TABLE 6.5: Sensitivity experiments overview.

Experiment	c_{var}	c_f	ΔTW	v
Baseline	1.0	1000	0	72
$E(0.5, 800, -, -)$	0.5	800	-	-
$E(0.5, 1000, -, -)$	0.5	1000	-	-
$E(0.5, 1200, -, -)$	0.5	1200	-	-
$E(1.0, 800, -, -)$	1.0	800	-	-
$E(1.0, 1200, -, -)$	1.0	1200	-	-
$E(1.5, 800, -, -)$	2.0	800	-	-
$E(1.5, 1000, -, -)$	2.0	1000	-	-
$E(1.5, 1200, -, -)$	2.0	1200	-	-
$E(-, -, -, -30)$	-	-	-30 min	-
$E(-, -, -, +30)$	-	-	+30 min	-
$E(-, -, 67, -)$	-	-	-	67 km/h
$E(-, -, 77, -)$	-	-	-	77 km/h

Note: A dash ('-') symbol indicates that the parameter remains unchanged from the baseline scenario.

First, the impact of cost fluctuations on the objective function is examined. This analysis evaluates the influence of cost variations on the evaluation criteria and deviations from historical planning, offering insights into the algorithm's performance across different cost scenarios. This is important because the trade-off between the number of vehicles and total travel time depends on the ratio between fixed and variable costs.

Second, the effect of time window tightness is analysed to evaluate its impact on the trade-off between service level and cost improvements. Explicit time windows, which define the allowable delivery or pick up periods, directly influence operational feasibility and cost. Modifying these time windows can impact the ability to meet customer demands while also affecting overall cost efficiency. While relaxing time constraints may reduce costs, it could also decrease service reliability. Additionally, beyond direct adjustments to time windows, incorporating a flexibility margin could be explored to assess whether minor modifications create additional opportunities for cost reduction while maintaining service feasibility. This trade-off analysis provides insights into balancing cost efficiency and service quality within the proposed solution.

Finally, a sensitivity analysis is performed on travel speed. While the model assumes a default speed of 72 km/h, its impact on the solution is evaluated. Travel speed is a critical factor influencing the service level, as lower speeds may prevent the timely fulfilment of all orders within the given constraints. A reduced travel speed restricts the feasibility of integrating multiple orders into a single route, potentially increasing the number of required vehicles or resulting in unfulfilled deliveries. In contrast, a higher travel speed is expected to enhance route consolidation and overall efficiency by allowing for more opportunities to combine goods, expanding the search space. By analysing variations in travel speed, this sensitivity analysis provides insights into its effect on route feasibility and service performance. The assumed travel speed of 72 km/h is an estimation and may not fully reflect real-world conditions, making it essential to analyse its impact on routing feasibility and service performance.

6.4.2 Evaluation Criteria

The algorithm's performance is measured using KPIs, which capture operational efficiency and service effectiveness. Following Section 2.2, the selected KPIs include:

- Objective value: The total cost of servicing all transportation orders over the planning horizon (see Equation 8a). This includes:
  - Total travel time: The cumulative duration of vehicles in transit.
  - Fleet utilisation: The number of vehicles deployed in the optimised solution.
- Load efficiency: The degree to which vehicle capacity is utilised, measured as the ratio of total used capacity to total available capacity across all deployed vehicles.
- Service level: The percentage of customer orders fulfilled, considering constraints and vehicle availability.

6.5 Findings and Implications

This chapter contributes to Phase 4: Modelling and Implementation of the research framework by detailing the validation process and experimental design necessary to assess the model's reliability and performance. The corresponding research questions addressed in this chapter are 7, 8, 9, and 10.

7. *How can the input data be processed and validated to ensure the model operates reliably?*

The data validation process, outlined in Section 6.1, ensures that the input data used for model execution is consistent, complete, and representative of real-world operations. This step is essential for maintaining solution accuracy and preventing errors caused by missing or inconsistent data. The key findings from this validation process are:

- Filtering and preprocessing historical data removed incomplete or inconsistent records, ensuring all input values aligned with operational conditions.
- Checks were included in the input data validation process to ensure that all data corresponded to complete vehicle routes, preventing partial or fragmented trips from being used in the model.
- Extensive filtering of raw data was required before it could be used as input in the model, as the original dataset contained numerous inconsistencies, missing values, and irrelevant records that needed to be addressed.

The filtering process highlights the importance of improving raw data quality before applying the algorithm, as inaccuracies or incomplete records can significantly impact the validity and reliability of the model's output.

8. *What key parameters influence the model's performance, and how can they be effectively tuned?*

The Optuna parameter-tuning software was used to optimise the tunable input parameters, improving the objective value of the model by a further 7.5% decrease in the objective. The key parameters and their optimal configuration are presented in Table 6.3.

9. *How can the model be evaluated in a case study to assess its applicability to real-world routing scenarios?*

The case study validation process compared model-generated routes with historical trips planned by Wolter Koops. The key takeaways from this evaluation are:

- The model reduces total costs by 31.8% compared to manual planning while increasing the loading efficiency by 8.1%. The algorithm decreases the number of vehicles from 22 to 12 while maintaining similar cumulative minutes travelled.
- The case study results suggest that the model applies to real-world operations, provided that assumptions regarding speed, vehicle availability, and time constraints are adequately managed.

These findings emphasise the model's practical applicability, demonstrating its capability to optimise vehicle routing in a cross-docking supply chain.

10. *What experimental setup is required to systematically test the model's performance and robustness?*

The experimental design, described in Section 6.4, systematically evaluates the model under different parameter configurations. The experimental design details the following:

- Scenarios: Multiple test scenarios are defined to evaluate different routing conditions and validate model performance under various operational constraints. These include variations in fixed cost definitions, compartmentalisation, and other logistical factors.
- Sensitivity analysis: The model's robustness is assessed by systematically adjusting key parameters such as variable and fixed costs, travel speed, and time windows. This analysis provides insights into the impact of parameter variations on cost efficiency, routing feasibility, and service performance.
- Evaluation criteria: The model's effectiveness is measured using KPIs, including total cost, loading efficiency, and service level. These metrics ensure that the model is assessed based on operational efficiency and its ability to meet customer requirements.

This chapter ensures that the model's performance can be systematically assessed for grounded conclusions on its effectiveness and limitations.

7 Experimental Execution and Computational Results

This chapter evaluates the computational performance of the proposed hybrid TS-SA heuristic by exposing the model to test scenarios and sensitivity analyses. This chapter contributes to Phase 4: Modelling and Implementation of the research framework by executing the resulting experimental setup described in Section 6.4. All experiments were conducted on a high-performance computing server equipped with an AMD EPYC 9534 64-Core Processor (128 threads) and 1024GB RAM.

Section 7.1 executes the scenarios outlined in the experimental design. Section 7.2 presents the results of the sensitivity analysis performed on the cost, time, and time window parameters.

7.1 Test Scenario Execution

The test scenarios outlined in Section 6.4 are executed and analysed in this section. Each scenario is examined in detail, highlighting its modifications to the model and their impact on the solution. The results of the scenarios are then compared to assess their implications on model performance.

7.1.1 Scenario 1: Algorithmic Variation on the Fixed Cost

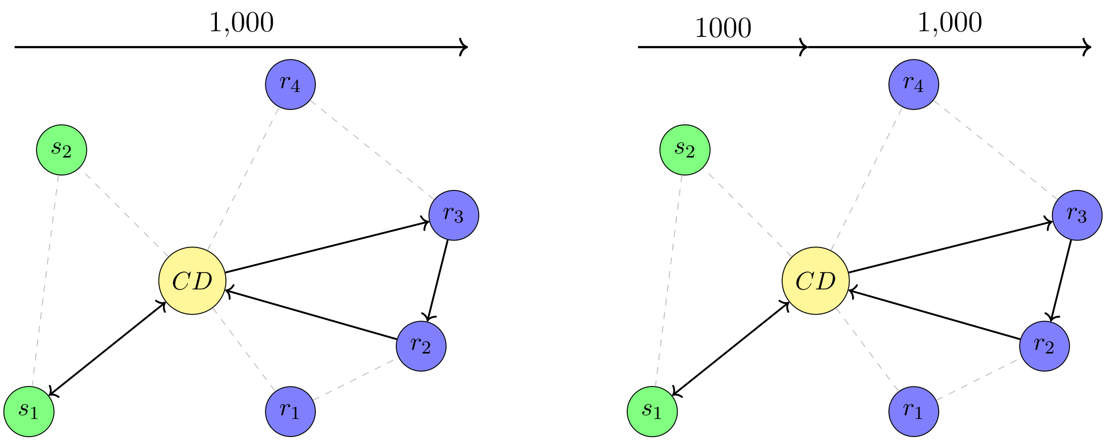
Scenario 1 represents the case where fixed costs are incurred each time a vehicle leaves from the CD, regardless of the type of route it makes. This variation of the base scenario requires a change in the definition of decision variable y^v, a binary decision variable that denotes whether a vehicle is used. In the current scenario, we generalise this definition to include the type of route a vehicle is used for, such that one vehicle may be used in two separate routes:

- y^v_π: Represents the case where vehicle v is used in a pick up route
- y^v_λ: Represents the case where vehicle v is used in a delivery route

After generalising the decision variable, the objective function becomes:

$$\min c_f \sum_{v\in\mathcal{V}} (y^v_\pi + y^v_\lambda) + c_{var} \sum_{(i,j)\in\mathcal{A},v\in\mathcal{V}} t_{ij}x^v_{ij} \tag{14}$$

To clarify these changes, we present a situation where the cost structure becomes evident. Consequently, Figure 7.1 compares both situations. Note that a fixed cost is incurred once in 7.1a because the vehicle depicted is used in the planning horizon. In 7.1b, however, the fixed cost is incurred twice: once when the vehicle departs the CD to pick up orders at supplier s_1, and a second time when the vehicle leaves the CD to deliver orders to retailers r_3 and r_4.



(A) Base Scenario

(B) Current Scenario

FIGURE 7.1: Comparison of vehicle routes and costs between the base scenario (a) and the current scenario (b). The timelines at the top indicate the difference in total fixed costs.

After running the algorithm for 3.5 hours with an identical configuration to the base scenario, an objective value of 30,407 is achieved, marking a 5.6% decrease in the objective function compared to the baseline. Additionally, the algorithm's proposed solution increases the loading efficiency by 6.9% .

7.1.2 Scenario 2: Impact of Compartmentalisation

In the base scenario, all vehicles have compartments to transport two different commodity types. Due to the complexity of modelling these constraints, the current scenario examines the impact of compartmentalisation on the model and its results.

Changes that need to be made can be summarised by looking at Constraints (8k) and (8l), where the right-hand side of the equation represents the number of compartments. The number of compartments may be set to any number, but it should be noted that the complexity increases due to an increase in the solution space. In the current scenario, we assess the impact of incorporating these compartments into the model. Therefore, we ran the model with only one compartment and evaluated the differences.

After running the algorithm for 3.5 hours, again with an identical configuration to the base scenario, an objective value of 26,104 is achieved. This marks an 18.9% decrease in the objective value. Furthermore, the load efficiency is 67.4%, which is a 12.2% decrease compared to the baseline.

Running the algorithm for 3.5 hours with three compartments per vehicle produced results similar to the base scenario, where vehicles have two compartments. This is because the input data contains only two product categories. The value of compartmentalization is expected to increase when more product categories are present.

7.1.3 Results and Comparison Between Scenarios

This section presents the results of the test scenarios, analysing their impact on the model's performance and solution quality. Each scenario is evaluated in terms of cost efficiency, convergence behaviour, and loading efficiency. The findings provide insights into the trade-offs introduced by different scenarios and offer recommendations for optimising vehicle routing and compartmentalisation strategies.

As depicted in Table 7.1, the baseline scenario shows superior results to scenarios 1 and 2. This is expected because both scenarios represent more constrained versions of the baseline. Regarding load efficiency, the scenario without compartments results in a decline of 12.2% compared to the baseline. Furthermore, a decline of 4.7% is observed compared to the historical planning. This result is anticipated, as the restriction to a single commodity type per vehicle, rather than two, necessitates the deployment of additional vehicles to accommodate the same demand.

TABLE 7.1: Historical Planning and TS-SA Model Performance (all scenarios)

Scenario	Total Cost (€)	Cost Gap (%)	Load Efficiency (%)	Efficiency Gap (%)
Historical Planning	32,199	–	70.7	–
Hybrid TS-SA Model (Base)	21,964	**-31.8**	76.8	+**8.1**
Hybrid TS-SA Model (S1)	30,407	**-5.6**	75.6	+**6.9**
Hybrid TS-SA Model (S2)	26,104	**-18.9**	67.4	**-12.2**

As Figure 7.2 shows, all scenarios exhibit a similar convergence pattern, characterised by rapid initial improvements that progressively diminish over time. The base scenario stabilises at a lower objective value than scenarios 1 and 2. This outcome is expected, as both scenarios introduce stricter constraints on the model. Scenario 2 achieves convergence in under two hours, significantly faster than the other scenarios. This aligns with expectations, as increasing the number of compartments expands the solution space considerably. Therefore, when planning orders with a single commodity type—or even two in some instances—it is advisable to limit the number of compartments to one to enhance computational efficiency.

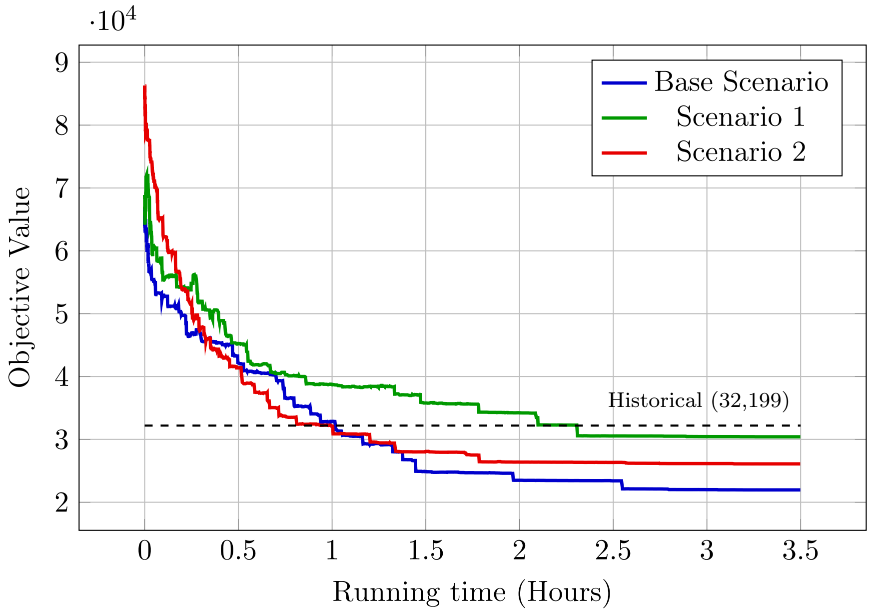


FIGURE 7.2: Objective Value Evolution over the Running Time for Scenarios 1 and 2

7.2 Sensitivity Analysis

A sensitivity analysis is conducted on key input parameters to assess the robustness of the proposed optimisation model under varying conditions. These parameters include the variable and fixed costs, the time windows, and the travel speed of a vehicle. This section presents the results of the experiments conducted, which are based on the sensitivity analysis detailed in Section 6.4. To this end, this section is structured by the different parameters on which the sensitivity analysis is performed: cost, time window tightness, and travel speed. Furthermore, it presents the performance of the best-found solution on the evaluation criteria of a given experiment $E\,(c_{var}, c_f, v, \Delta TW)$.

7.2.1 Cost Sensitivity

The sensitivity analysis results on the costs are detailed in Table 7.2. Following these results, a discussion is provided.

TABLE 7.2: Cost Sensitivity Analysis Results

Experiment	Total Cost (€)	Cost Gap (%)	Load Efficiency (%)	Efficiency Gap (%)
Baseline	21,964	76.8	-	-
$E(0.5, 800, -, -)$	17,434	**-20.6**	74.7	**-2.7**
$E(0.5, 1000, -, -)$	20,901	**-4.8**	70.6	**-8.1**
$E(0.5, 1200, -, -)$	23,996	+**9.3**	73.4	**-4.4**
$E(1.0, 800, -, -)$	21,652	**-1.4**	77.6	+**1.0**
$E(1.0, 1200, -, -)$	31,189	+**42.0**	71.3	**-7.1**
$E(1.5, 800, -, -)$	26,483	+**20.6**	75.9	**-1.2**
$E(1.5, 1000, -, -)$	33,404	+**52.1**	72.2	**-6.0**
$E(1.5, 1200, -, -)$	35,016	+**59.4**	74.7	**-2.7**

The results indicate a strong correlation between variable cost adjustments and the objective function value. A 50% decrease in the variable cost per minute travelled leads to an approximate 4.8% reduction in total cost, while a 50% increase results in a 52.1% increase in costs. This suggests that the objective function is more sensitive to increases in variable costs than to decreases, indicating a disproportionate impact of cost changes. Specifically, the relationship between cost changes and the objective function is asymmetric, with cost increases having a more significant effect than equivalent cost decreases.

Fixed costs also significantly and disproportionately influence total costs. A 20% decrease in fixed costs results in only a 1.4% decrease in the objective function, whereas a 20% increase in fixed costs leads to a 42% increase in the objective function. This highlights that the objective function is more sensitive to increases in fixed costs than to decreases.

The disproportionate sensitivity of the objective function to both fixed and variable cost changes underscores the significant impact of cost increases on total cost, while equivalent decreases have less effect. This asymmetry is important because it can influence strategic decisions, particularly in minimising fleet size and optimising vehicle utilisation. The findings suggest that managing cost increases should be prioritised as they significantly affect overall system performance.

In terms of decision-making, when fixed costs are high, there is a stronger incentive to minimise fleet size by maximising vehicle utilisation. In contrast, variable costs—such as fuel and distance-based expenses—directly impact route structuring. Lower variable costs encourage longer routes with more consolidated deliveries, as the cost per travelled minute becomes less significant. Conversely, higher variable costs lead to shorter, more direct routes to minimise total travel distance. Therefore, fixed costs shape vehicle allocation decisions, while variable costs drive route consolidation behaviour.

7.2.2 Travel Speed Sensitivity

The cost sensitivity analysis results are detailed in Table 7.2. Following these results, a discussion is provided. Note that the default travel speed in the baseline scenario is 72 km/h.

TABLE 7.3: Travel Speed Sensitivity Analysis Results

Experiment	Total Cost (€)	Cost Gap (%)	Load Efficiency (%)	Efficiency Gap (%)
Baseline	21,964	76.8	-	-
$E(-,-,62,-)$	26,775	+**22.0**	74.3	**-3.3**
$E(-,-,67,-)$	26,038	+**18.4**	72.0	**-6.3**
$E(-,-,77,-)$	21,633	**-1.5**	73.7	**-4.0**
$E(-,-,82,-)$	22,792	+**3.8**	70.9	**-7.7**

Not surprisingly, the experiments where the travel speed decreases show inferior results compared to the baseline scenario. For the $E(-,-,62,-)$ and (E $-,-,67,-)$ experiments, the cost increases by 22.0% and 18.4%, respectively, while the load efficiency decreases by 3.3% and 6.3%, respectively. A lower travel speed results in less feasible routes, allowing vehicles to visit fewer customers in one trip.

$E(-,-,77,-)$ shows a marginal decrease in the objective function, as expected, due to the increased travel speed. More interestingly, however, $E(-,-,82,-)$, where the travel speed is the highest, shows inferior results. Compared to the baseline scenario, the experiment shows an increase in costs of 3.8% and a load efficiency decrease of 7.7%. Therefore, we analyse these runs more closely to investigate the reason for this increase.

Upon closer examination of the runs, Figure 7.3 suggests that convergence occurs at later times as the travel speed increases. For example, in the case of $E(-,-,62,-)$, convergence appears to occur at approximately 1.6 hours of running time, whereas $E(-,-,82,-)$ only converges after 2.6 hours. This delay can be attributed to the increase in the number of feasible operations, which leads to a larger neighbourhood and, consequently, a longer convergence time. As the convergence time shifts, the diversification mechanism may fail to explore sufficiently when the algorithm becomes trapped in a local optimum, or it may not exploit solutions effectively to refine them.

The external cooling factor determines the rate at which the temperature decreases, which governs the balance between exploration (accepting worse solutions) and exploitation (focusing on improving solutions). Recall that the temperature regulates the acceptance of worse solutions, as defined by Equations 9 and 10, where the acceptance probability determines the likelihood of accepting a worse solution. Although the temperature scheme remained identical across the different experiments, the time to convergence varied significantly. This variation illustrates how different problem dynamics or configurations interact with the cooling schedule, resulting in earlier or later convergence.

To illustrate this further, we examine the acceptance probabilities under different cooling factors. The acceptance probability directly reflects the algorithm's willingness to explore

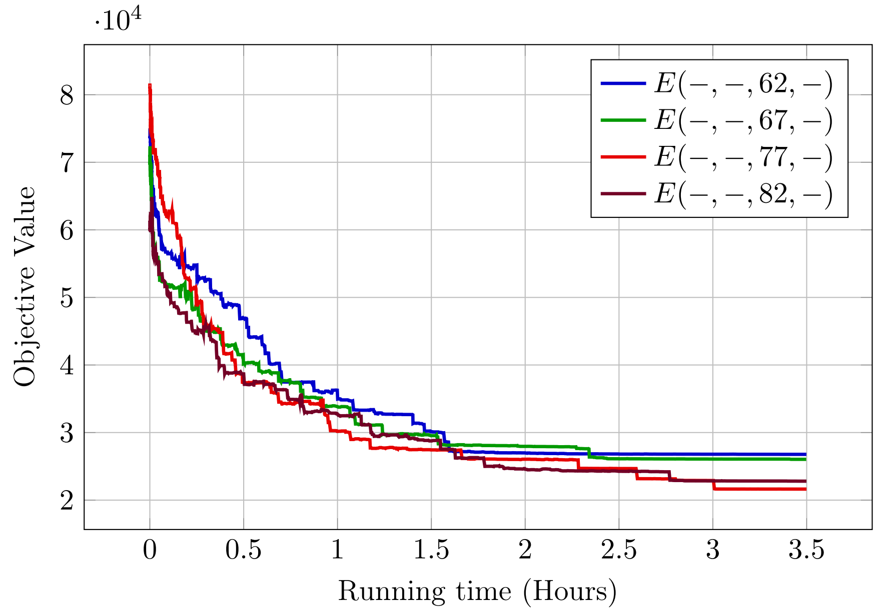


FIGURE 7.3: Objective Value over the Running Time for Different Travel Speeds

new areas of the solution space rather than exploiting known good solutions. A cooling factor that decreases the temperature too quickly may limit exploration, causing the algorithm to converge prematurely. On the other hand, a cooling factor that cools too slowly can result in prolonged exploration and unnecessary computational effort.

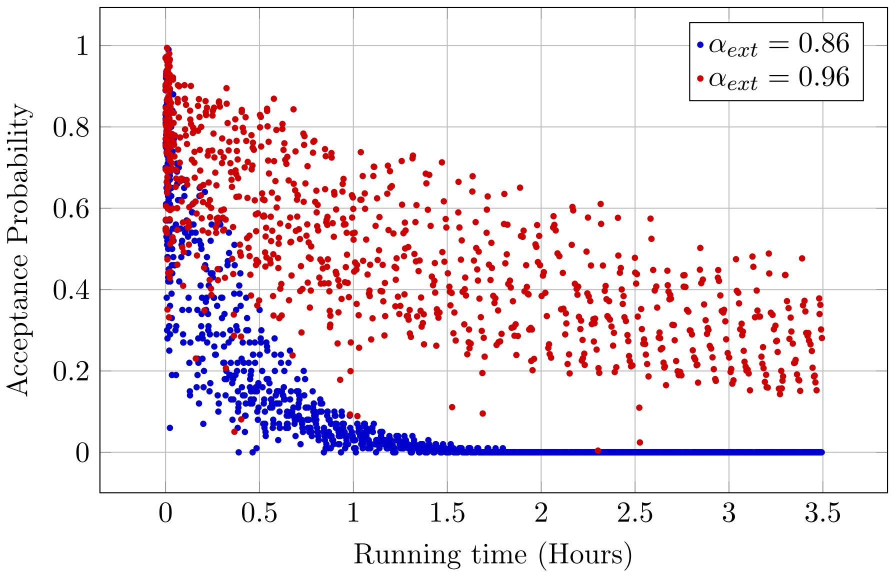


FIGURE 7.4: Average Acceptance Probability per Iteration over the Running Time

Setting the external cooling factor appropriately is crucial for regulating the exploration-exploitation trade-off, ensuring that the algorithm explores sufficiently to find high-quality solutions while converging within a reasonable time frame. To illustrate the exploration and exploitation behaviour under different external factors, Figure 7.4 depicts the average acceptance probability per iteration over the runtime of the algorithm when run with external cooling factors $\alpha_{ext} = 0.86$ and $\alpha_{ext} = 0.96$. This demonstrates how the external cooling factor affects the exploration-exploitation trade-off by accepting worse solutions, thereby reinforcing the importance of carefully tuning this parameter.

These findings emphasize the importance of tuning the parameters, particularly the cooling factor, as the algorithm converges at higher travel speeds during later stages. Recall that the external cooling factor has a considerable impact on the objective value, as shown in Section 6.2. Further analysis is required to establish the optimal parameter settings for different travel speeds.

7.2.3 Time Window Tightness Sensitivity

The sensitivity analysis results on time window tightness are detailed in Table 7.4. Following these results, a discussion is provided. Note that the baseline scenario uses the original time windows without any modifications.

TABLE 7.4: Time Window Tightness Sensitivity Analysis Results

Experiment	Total Cost (€)	Cost Gap (%)	Load Efficiency (%)	Efficiency Gap (%)
Baseline	21,964	76.8	-	-
$E(-,-,-,-60)$	27,433	**+24.9**	70.4	**-8.3**
$E(-,-,-,-30)$	25,266	**+15.0**	70.5	**-8.2**
$E(-,-,-,+30)$	25,130	**+14.4**	79.1	**+3.0**
$E(-,-,-,+60)$	24,282	**+10.6**	79.0	**+3.0**

As expected, experiments where the time windows are tightened (-60 and -30 minutes) show an increase in costs compared to the baseline scenario. In these cases, the cost increases by 24.9% and 15.0%, respectively, while the load efficiency decreases by 8.3% and 8.2%. This is primarily due to the reduced flexibility in scheduling, which forces vehicles to operate in a more constrained environment, leading to suboptimal routes and increased travel distances.

Conversely, when time windows are relaxed ($+30$ and $+60$ minutes), the results show slight improvements in cost and efficiency, but they do not outperform the baseline scenario. Compared to the stricter time window analyses, the cost reductions are only marginal, but load efficiency increases significantly. These results indicate that the solution remains trapped in a local optimum, as broader time windows should result in lower costs than the baseline. This suggests that the algorithm requires further tuning to fully exploit the benefits of broader time windows.

These findings underscore the need for further refinement in the solution approach. Simply relaxing constraints does not inherently lead to superior outcomes unless they are appropriately tuned. Figure 7.5 illustrates the trend presented in Section 7.2.2, where less constrained situations converge at later stages, resulting in the diversification mechanism being disabled early. The exploration-exploitation trade-off should, therefore, be tested more in search of the global optimum.

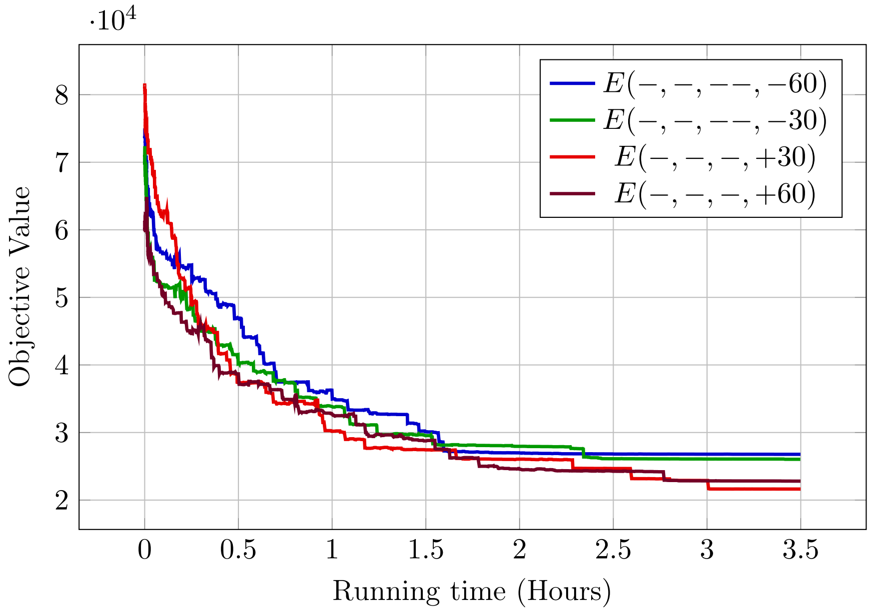


FIGURE 7.5: Objective Value Evolution over the Running Time for Different Time Window Tightness Sensitivities

8 Conclusions and Recommendations

This research develops a metaheuristic to optimise vehicle routing in temperature-controlled cross-docking environments, aiming to minimise the total cost of servicing orders with a fleet of homogeneous vehicles. The orders, originating from geographically dispersed suppliers and retailers, rely on a cross-docking strategy where goods are picked up from suppliers, consolidated, and then delivered to retailers. This approach is particularly suited to the operational structure, where suppliers and retailers are geographically dispersed, with a CD serving as an intermediary. The central research question addressed in this study is:

How can an optimisation algorithm be developed to minimise the total cost of servicing all suppliers and retailers in Wolter Koops' fleet routing operations?

Section 8.1 presents the findings in response to this central research question. Section 8.2 summarises the contributions of this study to science and practice. Section 8.3 provides suggestions for Wolter Koops to improve routing and consolidation decisions through the proposed algorithm. Finally, Section 8.4 presents the limitations of this study and the future research opportunities that result from these limitations. Note that these provide answers to the final research question 11 of the research framework.

8.1 Conclusions

This research aimed to optimise routing and consolidation decisions in a structured cross-docking environment by developing a tailored optimisation algorithm. The analysis of current vehicle routing practices at Wolter Koops revealed a highly constrained planning environment, with over 5,500 trips scheduled weekly. The study focused on data of historical routes of the EG planning department to validate the model. Several critical factors were identified that influence routing decisions, such as strict time windows, travel times, driver work regulations, commodity availability, and product compatibility. These factors highlight the inherent complexity of vehicle routing in cross-docking networks, particularly when temperature-controlled, multi-compartment vehicles are involved.

The literature review provided insights into existing VRP solutions and modelling approaches. While the VRPCD has been studied extensively, current approaches fail to reflect the operational realities faced by Wolter Koops. Existing algorithms do not integrate key constraints such as mandatory driver rest periods, direct supplier-to-retailer shipments, and multi-compartment vehicle compatibility considerations. This gap in the literature, combined with the identified operational challenges, defined this study's scope and novelty.

Given the combinatorial complexity of the VRPCD variant addressed, finding optimal solutions using exact methods is computationally infeasible for real-world instances. Even medium-sized problem instances lead to exponential growth in solution space, making exact approaches impractical for operational use. Consequently, metaheuristic algorithms were explored as a feasible alternative for solving the problem effectively within reasonable computation times.

After evaluating various metaheuristic approaches, TS was selected due to its promising results in solving other complex VRPs. However, further analysis showed that TS alone can struggle to avoid local optima in highly constrained search spaces. To overcome this limitation, a probabilistic exploration mechanism inspired by SA was integrated into the TS framework. The hybrid TS-SA algorithm combines the guidance of TS's memory-based search with SA's probabilistic exploration capabilities, allowing the search process to escape local optima and explore diverse regions of the solution space more effectively. The resulting hybrid TS-SA algorithm was developed and tested on historical data from Wolter Koops' EG department. Using a two-day dataset of 133 transport orders, the algorithm reduced the number of vehicles from 22 to 12, decreased total travel time from 10,199 minutes to 9,964 minutes, and reduced total transportation costs by 31.8% (from €32,199 to €21,964). Additionally, the average load efficiency increased by 8.1% compared to historical planning. These results demonstrate a substantial improvement over manual planning, achieved within approximately 3.5 hours of computing time.

Further validation was performed through multiple scenarios and sensitivity analyses. These included variations in fixed cost structures, removing multi-compartment capabilities, and adjusting travel speeds and time window tightness. In all cases, the hybrid TS-SA algorithm outperformed historical manual planning, confirming the robustness and adaptability of the proposed approach. Nevertheless, further experimentation is recommended to fine-tune parameter configurations and assess their performance under a broader range of operational conditions.

The study's objective—minimising routing and consolidation decisions through an algorithm—has been achieved, though the solution's optimality remains unverified due to the absence of an optimal benchmark. Importantly, this research represents a novel contribution to theory and practice by integrating multi-compartment vehicle constraints, temperature compatibility, strict time windows, cross-docking synchronisation, and driver rest regulations into a single VRPCD framework. To our knowledge, no prior work has addressed this combination of constraints, making this approach an innovative solution to a complex and underexplored problem in cross-docking logistics.

Despite the promising results, several limitations remain. Travel times were estimated using average speeds, which may not accurately reflect real-world variations such as traffic or unforeseen delays. Additionally, the algorithm was tested on a limited dataset from a single department over two days, which restricts the generalisability of the findings. Future research should expand testing across larger datasets and different logistics networks to validate the solution's scalability and adaptability.

In conclusion, this research demonstrates the effectiveness of hybrid metaheuristic optimisation techniques in addressing complex vehicle routing challenges in structured cross-docking environments. The developed hybrid TS-SA algorithm offers practical improvements for Wolter Koops and lays a foundation for future advancements in automated routing and consolidation strategies for multi-compartment routing in cross-docking logistics networks.

8.2 Contributions to Science and Practice

This section outlines the scientific and practical contributions of the study, concluding with a discussion on the generalisability of the findings.

8.2.1 Scientific Contributions

This research advances the scientific understanding of VRPs in cross-docking networks, particularly those involving multi-compartment vehicles and temperature-sensitive goods. While the literature offers numerous approaches to the VRPCDs, few models explicitly address the complexities of multi-compartment constraints in combination with strict time windows, rest periods, and cross-docking synchronisation.

Key scientific contributions include:

- The development of a hybrid TS-SA metaheuristic tailored to multi-compartment vehicle routing with cross-docking operations, integrating diversification and intensification strategies to enhance solution quality and robustness.
- A mathematical framework that models vehicle routing under compartment compatibility and temperature constraints, offering an extension to existing VRPCD formulations.
- Empirical evidence demonstrating the effectiveness and efficiency of hybrid metaheuristics in solving a complex VRPCD instance, validated through real-world data from logistics service provider Wolter Koops.

This research fills a gap in the VRP literature by addressing the underexplored intersection of cross-docking and multi-compartment routing, providing a foundation for future studies to expand upon. To our knowledge, we are the first to implement time windows, driver break times, a variable number of vehicles, and direct route possibilities in an instance of the VRPCD.

8.2.2 Practical Contributions

From a practical perspective, this research provides an implementable optimisation approach that has demonstrated substantial improvements in vehicle routing and consolidation planning within Wolter Koops' cross-docking operations. The hybrid TS-SA algorithm developed in this study has shown the potential to reduce transportation costs and improve operational efficiency significantly.

Key practical contributions include:

- A demonstrated cost reduction of 31.8% and an 8.1% increase in load efficiency compared to historical manual planning, based on a representative two-day dataset. These results illustrate the potential value of algorithmic decision support in complex routing environments.
- An application which integrates into Wolter Koops' current application environment, as illustrated by Figure 2.1. The integration readiness of the algorithm with existing TMSs enables planners to automate routing decisions and standardise planning processes currently reliant on manual expertise.
- A proof of concept for applying hybrid metaheuristic approaches to logistics networks involving multi-compartment vehicles, cross-docking synchronisation, and strict time windows. This underscores the potential of advanced optimisation techniques to enhance day-to-day logistics operations.
- Actionable recommendations for logistics service providers to enhance routing and consolidation processes. These are discussed in Section 8.3.

This research bridges the gap between theoretical optimisation methods and applied logistics management by providing an implementable solution validated with actual operational data.

8.2.3 Generalisability and Discussion

The hybrid TS-SA metaheuristic developed in this study addresses a complex variant of the VRP that can be generalised to other logistics networks featuring cross-docking and multi-compartment vehicle routing. While the case study focuses on Wolter Koops' operations in temperature-sensitive transport, the underlying optimisation approach and algorithmic structure can be adapted to:

- Other industries that involve commodity incompatibilities, such as pharmaceuticals, chemicals, or food and beverage logistics.
- Cross-docking facilities with similar consolidation and synchronisation requirements.
- Multi-compartment fleets, where compartmentalisation is necessary due to temperature or contamination concerns.

While the algorithm is generalisable across these supply chain characteristics, it is essential to note that it was validated using data from two days at Wolter Koops. Therefore, further testing is needed to assess the generalisability of the model in different settings and larger problem instances. As demonstrated in this study, parameter tuning is crucial for ensuring optimal performance in various operational settings and must be considered when implementing the algorithm.

The cross-docking structure used in this study involves a single centralised CD. The model is designed for structured cross-docking supply chains, characterised by a clear, hierarchical flow from suppliers (upstream) through the CD to retailers (downstream). This structure simplifies routing and consolidation decisions by maintaining distinct phases: vehicles pick up goods from suppliers or deliver goods to retailers. More details on routing possibilities can be found in Section 4.2.

While the model allows for direct routes involving a single pick up and a single delivery—as an exception to the general routing structure—it does not consider more complex combined pick up and delivery tours within the same route. Extending the model to support multiple pick ups and deliveries on a single trip or to operate in decentralised or multi-CD networks would introduce additional synchronisation and coordination complexities. Such scenarios would require adaptations to the current approach, particularly to handle dynamic consolidation, vehicle availability, and order allocations to a CD.

Reliability refers to the consistency of a measure [105]. While the two-day data sample provided consistent results, a more extended validation, incorporating data from different days, seasons, or scenarios, would provide more substantial evidence of the heuristic's overall reliability. Nonetheless, the consistency observed across the baseline, other scenarios, and sensitivity analyses within the two-day sample indicate that the algorithm is sufficiently robust to manage real-world logistics scenarios under stable conditions. However, parameter tuning requires careful attention to optimise performance across other scenarios.

8.3 Recommendations

The insights from this study lead us to recommend the following actions for Wolter Koops when implementing the algorithm, as well as for broader considerations:

- As the input data used in this study were obtained directly from master data, it is recommended to investigate the issues identified in the data preparation and filtering process (see Section 6.1). Specifically, missing elements and exceptions must be identified and addressed before providing the algorithm with invalid input data.
- Integrate the algorithm into the existing TMS to facilitate automated testing of routing decisions. Initially, we recommend running the model overnight to plan orders for the following day. After thorough testing and validation in a practical setting, it is suggested to reassess the planning horizon and consider more frequent updates to accommodate new orders as they arrive throughout the day. Implement the model in phases, starting with small-scale testing and gradually expanding the scope to larger instances. This phased approach will enable smoother integration and facilitate easier troubleshooting.
- It is recommended to conduct real-life tests to compare the algorithm with manual planning and determine optimal configurations under varying operational constraints, as the robustness of the algorithm depends on its ability to adapt to different operational conditions. By conducting real-life tests, Wolter Koops can assess the algorithm's performance in real-world scenarios and identify potential adjustments needed to optimise its effectiveness across various use cases, such as varying problem sizes, traffic conditions, and changing time windows. According to the Optuna parameter tuning results, the external cooling factor α_{ext} has a significant impact on the objective value (see Section 6.2). It is therefore recommended to start by tuning this parameter. This will help ensure the algorithm's robustness and practicality in day-to-day operations.

8.4 Limitations and Future Research

This section discusses the limitations of this study, categorising them into methodological and practical limitations. Furthermore, it outlines future research directions to address these limitations and improve the model's applicability and generalisability.

8.4.1 Methodological Limitations

This section presents the methodological limitations of this study, referring to constraints inherent to the research design, assumptions, or chosen modelling approach. These include the following:

- Travel times are estimated by retrieving the total distance between two locations from the Google Maps API and then converting it using an average travel speed.
- Due to time limitations, not all scenarios were tuned in terms of their parameter configuration. Since their solution space does not change in size, it is expected that the effect of further parameter tuning would be minimal. However, it is recommended to perform more testing on parameter configurations before implementing the algorithm.

- The validation was based on a specific Wolter Koops' EG department dataset spanning two days in January. Therefore, the model's performance and applicability may differ in other logistical contexts with varying operational constraints, demand patterns, or cost structures. Future research could assess the model's effectiveness across different datasets or companies to evaluate its broader applicability.
- Fixed costs c_f and variable cost c_v are based on an estimation and influence the results, as indicated by the sensitivity analysis in Section 7.2.1.

8.4.2 Practical Limitations

This section presents practical limitations that arise from real-world constraints, such as data availability, computational resources, and operational feasibility. These limitations are the following:

- The algorithm is tested on a dataset with 133 orders from the original dataset. The model was not analysed further for scalability. Testing the algorithm's scalability can future-proof it if this demand increases.
- The real-world applicability of the model depends on the ability to structure input data according to the conventions established in this thesis (see Section 6.1). Data challenges encountered during the preparation phase highlight the importance of data availability, consistency, and preprocessing. In practice, discrepancies in data formats or missing information may hinder implementation.
- The model is developed to solve the VRPCD in cross-docking supply chains and does not apply to mixed pick up and delivery routing. pick up and delivery trips with multiple pick up and delivery actions are not included, and direct shipments may only include one supplier-retailer combination.
- The model is designed specifically for route optimisation from a given CD and does not support multi-CD optimisation. It assumes that all orders are pre-assigned to a designated CD, meaning the allocation of orders across multiple CDs is not considered in the optimisation process. This assumption simplifies the problem but limits applicability in cases where order allocation between multiple CDs is a decision variable.
- The model does not account for vehicle cooling times, and loading and unloading times are generalised into a single average value. This reduces computational complexity but may affect accuracy in scenarios where loading and unloading times differ significantly or precise timing constraints are critical.

8.4.3 Future Research Directions

This subsection highlights key areas for future research that could build upon the current study. These directions aim to improve the model's effectiveness further and extend its applicability to real-world scenarios by addressing aspects not fully explored within the scope of this research.

- The current approach assumes that transport orders are pre-assigned to a specific CD without considering alternative allocations. Future research could explore methods for optimising CD assignments to improve efficiency and reduce transportation costs. This way, multi-CD routing problems can be solved by including the allocation decision.

- The current model assumes a static planning horizon, but real-world logistics require dynamic adaptation as new information becomes available. Implementing a real-time algorithm for the dynamic VRP could improve decision-making by continuously updating routes in response to demand fluctuations, delays, or unforeseen disruptions.
- The model assumes generalised travel and service times. A more detailed analysis of travel time variability and loading/unloading duration could improve accuracy by incorporating congestion, handling efficiency, and operational delays. The model could include stochastic times to reflect operational reality better.
- While tested for the current case study, the model's computational performance on even larger-scale datasets and different operational settings remains untested. Future work could assess its efficiency on larger problem instances and explore metaheuristic enhancements to improve scalability further. Furthermore, robustness across varying operational settings could be investigated.
- Scheduling methods could be applied to optimise dock usage, treating docks as servers and (un)loading and consolidation as jobs to be scheduled. Further research could explore the interaction between routing and scheduling by dynamically re-optimising the routing plan based on updated service time estimates at the CD [22, 23, 106, 107].
- Parameter tuning is required to guide the optimisation process effectively. Future research could explore self-tuning parameters to improve model robustness across varying scenarios and runtimes.

Bibliography

[1] Jan Van Belle, Paul Valckenaers, and Dirk Cattrysse. Cross-docking: State of the art. *Omega*, 40(6):827–846, 2012. ISSN 0305-0483. doi: 10.1016/j.omega.2012.01.005. Special Issue on Forecasting in Management Science.

[2] Tomáš Režnar, Jan Martinovič, Kateřina Slaninová, Ekaterina Grakova, and Vít Vondrák. Probabilistic time-dependent vehicle routing problem. *Central European Journal of Operations Research*, 25(3):545–560, 9 2017. ISSN 16139178. doi: 10.1007/s10100-016-0459-2.

[3] Wolter Koops. Homepage. Available online at https://www.wolterkoops.com/nl/, 2024. Accessed: May 3, 2024.

[4] Head of planning at Wolter Koops. Interview, 2024. Personal communication.

[5] IT Manager at Wolter Koops. Interview, 2024. Personal communication.

[6] Mendix. Mendix: Low-code development platform, 2024. URL https://www.mendix.com. Accessed: 15-10-2024.

[7] CAPE Groep. Homepage. Available online at https://capegroep.nl/, 2024. Accessed: May 3, 2024.

[8] Business Consultant at CAPE Groep. Interview, 2024. Personal communication.

[9] Hans Heerkens and Arnold van Winden. *Solving Managerial Problems Systematically*. Noordhoff Uitgevers, 2017. ISBN 9789001887957. Translated into English by Jan-Willem Tjooitink.

[10] Layla Martin. Hamiltonian cycles, 10 2023. URL https://www.youtube.com/watch?v=6PBndbvMHT4&list=PLluFr543UdZf4XT95DdWd7DLO2HPdD8UU.

[11] Koen Turnhout, Marjolein Jacobs, Irene Kamp, Charlie Mulholland, Andre Neuman, Sonja Rouwhorst, and Laura van der Vlies. Cmd methods pack: Find a combination of research methods that suit your needs. HAN University of Applied Sciences - Amsterdam University of Applied Sciences, the Netherlands. Available online at https://cmdmethods.nl/, 2015.

[12] G Dantzig and J Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959. doi: 10.1287/mnsc.6.1.80.

[13] Merrill M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956. doi: 10.1287/opre.4.1.61.

[14] Julia Rieck and Jürgen Zimmermann. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181:337–358, 12 2010. ISSN 02545330. doi: 10.1007/s10479-010-0748-4.

[15] B.L. Golden, S. Raghavan, and E.A. Wasil. *The vehicle routing problem: Latest advances and new challenges*, volume 43. Springer, 2008. doi: 10.1007/978-0-387-77778-8.

[16] Lawrence D. Bodin. A taxonomic structure for vehicle routing and scheduling problems. *Computers & Urban Society*, 1(1):11–29, 1975. ISSN 0305-7097. doi: 10.1016/0305-7097(75)90003-4.

[17] Lawrence Bodin and Bruce Golden. Classification in vehicle routing and scheduling. *Networks*, 11(2):97–108, 1981. doi: 10.1002/net.3230110204.

[18] M. Desrochers, J.K. Lenstra, and M.W.P. Savelsbergh. A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46(3):322–332, 1990. ISSN 0377-2217. doi: 10.1016/0377-2217(90)90007-X.

[19] Gilbert Laporte and Ibrahim Osman. Routing problems: A bibliography. *Annals of Operations Research*, 61:227–262, 01 1995. doi: 10.1007/BF02098290.

[20] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009. doi: 10.1287/trsc.1090.0301.

[21] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review, 11 2009. ISSN 03608352.

[22] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review, 9 2016. ISSN 03608352.

[23] Harilaos N. Psaraftis, Min Wen, and Christos A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 1 2016. ISSN 10970037. doi: 10.1002/net.21628.

[24] Thibaut Vidal, Gilbert Laporte, and Piotr Matl. A concise guide to existing and emerging vehicle routing problem variants, 10 2020. ISSN 03772217.

[25] Brenner Humberto Ojeda Rios, Eduardo C. Xavier, Flávio K. Miyazawa, Pedro Amorim, Eduardo Curcio, and Maria João Santos. Recent dynamic vehicle routing problems: A survey. *Computers and Industrial Engineering*, 160, 10 2021. ISSN 03608352. doi: 10.1016/j.cie.2021.107604.

[26] Haifei Zhang, Hongwei Ge, Jinlong Yang, and Yubing Tong. Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms. *Archives of Computational Methods in Engineering*, 29(1):195–221, 1 2022. ISSN 18861784. doi: 10.1007/s11831-021-09574-x.

[27] C.-Y Liong, I. Wan, and Khairuddin Omar. Vehicle routing problem: Models and solutions. *Journal of Quality Measurement and Analysis*, 4:205–218, 01 2008.

[28] Grigorios D. Konstantakopoulos, Sotiris P. Gayialis, and Evripidis P. Kechagias. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research*, 22(3):2033–2062, 7 2022. ISSN 18661505. doi: 10.1007/s12351-020-00600-7.

[29] Paolo Toth and Daniele Vigo. *Vehicle Routing - Problems, Methods, and Applications (2nd Edition)*. Society for Industrial and Applied Mathematics (SIAM), 2014. ISBN 978-1-611973-58-7. doi: 10.1137/1.9781611973594.

[30] Michael R. Garey and David S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness*. W.H. Freeman, 1979. ISBN 0716710447.

[31] P Toth and D Vigo. *The Vehicle Routing Problem*, chapter 1. An overview of Vehicle Routing Problems, pages 1–26. SIAM Publication, 2002. doi: 10.1137/1.9780898718515.ch1.

[32] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 04:66–74, 2012. ISSN 2160-5912. doi: 10.4236/iim.2012.43010.

[33] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21, 2013. ISSN 0377-2217. doi: 10.1016/j.ejor.2013.02.053.

[34] Raafat Elshaer and Hadeer Awad. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140:106242, 2020. ISSN 0360-8352. doi: 10.1016/j.cie.2019.106242. URL https://www.sciencedirect.com/science/article/pii/S0360835219307119.

[35] Alexander Gutiérrez-Sánchez and Linda Bibiana Rocha-Medina. Vrp variants applicable to collecting donations and similar problems: A taxonomic review. *Computers & Industrial Engineering*, 164:17, 2022. ISSN 0360-8352. doi: 10.1016/j.cie.2021.107887.

[36] Rahma Lahyani, Mahdi Khemakhem, and Frédéric Semet. Rich vehicle routing problems: From a taxonomy to a definition, 2 2015. ISSN 03772217.

[37] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. ieee trans syst man cybernetics - part b. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 26:29–41, 02 1996. doi: 10.1109/3477.484436.

[38] F. Stavropoulou. The Consistent Vehicle Routing Problem with heterogeneous fleet. *Computers and Operations Research*, 140, 4 2022. ISSN 03050548. doi: 10.1016/j.cor.2021.105644.

[39] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley Interscience Series in Discrete Mathematics and Optimization. Wiley, New York, 1990.

[40] Adam N. Letchford and Juan José Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105:251–274, 2 2006. ISSN 14364646. doi: 10.1007/s10107-005-0652-x.

[41] Tânia Rodrigues Pereira Ramos, Maria Isabel Gomes, and Ana Paula Barbosa Póvoa. Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*, 23:103–120, 3 2020. ISSN 1469848X. doi: 10.1080/13675567.2019.1630374.

[42] Hanif D. Sherali and Patrick J. Driscoll. On tightening the relaxations of miller-tucker-zemlin formulations for asymmetric traveling salesman problems. *Operations Research*, 50(4):656–669, 2002. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/3088515.

[43] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5):723–738, 2004. doi: 10.1287/opre.1040.0111.

[44] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977. doi: https://doi.org/10.1002/net.3230070203.

[45] Imdat Kara, Gilbert Laporte, and Tolga Bektas. A note on the lifted miller-tucker-zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158:793–795, 11 2004. ISSN 03772217. doi: 10.1016/S0377-2217(03)00377-1.

[46] C E Miller, A W Tucker, and R A Zemlin. Integer programming formulation of traveling salesman problems, 1960.

[47] Olli Bräysy, Michel Gendreau, and Geir Hasle. Genetic algorithms for the vehicle routing problem with time windows. *SINTEF Applied Mathematics, Research Council of Norway*, 2001. URL https://www.researchgate.net/publication/241100650.

[48] Marius M. Solomon and Jacques Derosiers. Time window constrained routing and scheduling problems. *Transportation Science*, 22:1–13, 1988. ISSN 00411655. doi: 10.1287/trsc.22.1.1.

[49] Eric Taillard, Philippe Badeau, Michel Gendrau, Francois Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Institute for Operations Research and the Management Sciences*, 31, 5 1997.

[50] M Desrochers, J K Lenstra, M W P Savelsbergh, and F Soumis. Vehicle routing with time windows: Optimization and approximation. *Elsevier Science Publishers B. V.*, 16, 1988.

[51] Jixian Xiao and Bing Lu. Vehicle routing problem with soft time windows. In *Advances in Computer Science and Information Engineering*, pages 317–322, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-30126-1. doi: 10.1007/978-3-642-30126-1_51.

[52] Wenjuan Gu, Claudia Archetti, Diego Cattaruzza, Maxime Ogier, Frédéric Semet, and M Grazia Speranza. Vehicle routing problems with multiple commodities: A survey. *European Journal of Operational Research*, pages 1–15, 2024. doi: 10.1016/j.ejor.2023.11.032Ãŕ.

[53] Ulrich Derigs, Jens Gottlieb, Jochen Kalkoff, Michael Piesche, Franz Rothlauf, and Ulrich Vogel. Vehicle routing with compartments: Applications, modelling and heuristics. *OR Spectrum*, 33:885–914, 10 2011. ISSN 01716468. doi: 10.1007/s00291-010-0194-3.

[54] Diego Cattaruzza, Nabil Absi, and Dominique Feillet. Vehicle routing problems with multiple trips. *A Quarterly Journal of Operations Research*, 14:223–259, 2016. doi: 10.1007/s10288-016-0306-2Ãŕ.

[55] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi: 10.1287/opre.2.4.393. URL https://doi.org/10.1287/opre.2.4.393.

[56] Abdellah El Fallahi, Christian Prins, and Roberto Wolfler Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35(5):1725–1741, 2008. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2006.10.006. Part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility.

[57] Young Hae Lee, Jung Woo Jung, and Kyong Min Lee. Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering*, 51(2):247–256, 2006. ISSN 0360-8352. doi: 10.1016/j.cie.2006.02.006. Special Issue: Logistics and Supply Chain Management.

[58] Vincent F. Yu, Parida Jewpanya, A.A.N. Perwira Redi, and Yu-Chung Tsao. Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks. *Computers & Operations Research*, 129:105205, 2021. ISSN 0305-0548. doi: 10.1016/j.cor.2020.105205.

[59] Aldy Gunawan, Audrey Tedja Widjaja, Pieter Vansteenwegen, and Vincent F. Yu. A matheuristic algorithm for the vehicle routing problem with cross-docking. *Applied Soft Computing*, 103:107163, 2021. ISSN 1568-4946. doi: 10.1016/j.asoc.2021.107163.

[60] Vitor A.A. Souza, Rafael A. Melo, and Geraldo R. Mateus. The vehicle routing problem with cross-docking and scheduling at the docking station: Compact formulation and a general variable neighborhood search metaheuristic. *Applied Soft Computing*, 161:111744, 2024. ISSN 1568-4946. doi: 10.1016/j.asoc.2024.111744.

[61] Philippe Grangier, Michel Gendreau, Fabien Lehuédé, and Louis-Martin Rousseau. An adaptive large neighborhood search for a vehicle routing problem with cross-dock under dock resource constraints. In *Odysseus 2015 - Workshop on Freight Transportation and Logistics*, May 2015.

[62] C.-J. Ting and T.-D. Chen. Integrate vehicle routing and truck sequencing in cross-docking operations. *13th IMHRC Proceedings*, 2014.

[63] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau, and G. Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2009. doi: 10.1057/jors.2008.108.

[64] Ali Baniamerian, Mahdi Bashiri, and Reza Tavakkoli-Moghaddam. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. *Applied Soft Computing*, 75:441–460, 2019. ISSN 1568-4946. doi: 10.1016/j.asoc.2018.11.029.

[65] Ching-Jong Liao, Yaoming Lin, and Stephen C. Shih. Vehicle routing with cross-docking in the supply chain. *Expert Systems with Applications*, 37(10):6868–6873, 2010. ISSN 0957-4174. doi: 10.1016/j.eswa.2010.03.035.

[66] Vincent F. Yu, Parida Jewpanya, and A.A.N. Perwira Redi. Open vehicle routing problem with cross-docking. *Computers & Industrial Engineering*, 94:6–17, 2016. ISSN 0360-8352. doi: 10.1016/j.cie.2016.01.018.

[67] Fernando Afonso Santos, Geraldo Robson Mateus, and Alexandre Salles da Cunha. A branch-and-price algorithm for a vehicle routing problem with cross-docking. *Electronic Notes in Discrete Mathematics*, 37:249–254, 2011. ISSN 1571-0653. doi: 10.1016/j.endm.2011.05.043. LAGOS'11 – VI Latin-American Algorithms, Graphs and Optimization Symposium.

[68] Christos D. Tarantilis. Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters*, 7(7):1583–1596, 2013. ISSN 1862-4480. doi: 10.1007/s11590-012-0558-5.

[69] Ali Esfahani and M.B. Fakhrzad. Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-heuristic algorithms. *International Journal of Engineering*, 27(7):1113–1126, 2014. ISSN 1025-2495. URL https://www.ije.ir/article_72345.html.

[70] A. Touihri, O. Dridi, and S. Krichen. A multi operator genetic algorithm for solving the capacitated vehicle routing problem with cross-docking problem. In *2016 IEEE Symposium Series on Computational Intelligence, SSCI*, pages 1–8. IEEE, 2016.

[71] P. Grangier, M. Gendreau, F. Lehuédé, and L.-M. Rousseau. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research*, 84:116–126, 2017.

[72] V.W. Morais, G.R. Mateus, and T.F. Noronha. Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications*, 41(16):7495–7506, 2014.

[73] P. Grangier, M. Gendreau, F. Lehuédé, and L.-M. Rousseau. The vehicle routing problem with cross-docking and resource constraints. *Journal of Heuristics*, pages 1–31, 2019.

[74] Rodolfo Dondo and Jaime Cerdá. A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. *Computers & Chemical Engineering*, 63:184–205, 2014. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2013.12.012.

[75] J.M. Urtasun and E. Montero. An study of operator design under an adaptive approach for solving the cross-docks vehicle routing problem. In *2019 IEEE Congress on Evolutionary Computation, CEC*, pages 2098–2105. IEEE, 2019.

[76] Rodolfo Dondo and Jaime Cerdá. The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. *Computers & Chemical Engineering*, 76:42–62, 2015. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2015.02.003.

[77] Rijksoverheid. Wat is er wettelijk geregeld voor de rijtijden en rusttijden bij wegvervoer? Available online at https://www.rijksoverheid.nl/onderwerpen/werktijden/vraag-en-antwoord/rijtijden-en-rusttijden-wegvervoer#%3A~%3Atext=Als%20chauffeur%20mag%20u%3A,weken%20maximaal%2090%20uur%20rijden., 2024. Accessed: January 29, 2025.

[78] Gilbert Laporte, Stefan Ropke, and Thibaut Vidal. Chapter 4: Heuristics for the vehicle routing problem. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 87–116. SIAM, 2014.

[79] Sachin Desale, Akhtar Rasool, Sushil Andhale, and Priti Rane. Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. *Int. J. Comput. Eng. Res. Trends*, 351(5):2349–7084, 2015.

[80] Jean-François Cordeau, Gilbert Laporte, Martin W.P. Savelsbergh, and Daniele Vigo. Chapter 6 vehicle routing. In *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 367–428. Elsevier, 2007. doi: 10.1016/S0927-0507(06)14006-2.

[81] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2005.09.012.

[82] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: 10.1126/science.220.4598.671.

[83] Gunter Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993. ISSN 0021-9991. doi: 10.1006/jcph.1993.1010.

[84] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986. ISSN 0305-0548. doi: 10.1016/0305-0548(86)90048-1. Applications of Integer Programming.

[85] Fred Glover, Manuel Laguna, and Rafael Marti. Principles of tabu search. *Approximation algorithms and metaheuristics*, 23:1–12, 2007.

[86] John Baxter. Depot location: A technique for the avoidance of local optima. *European journal of operational research*, 18(2):208–214, 1984.

[87] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997. ISSN 0305-0548. doi: 10.1016/S0305-0548(97)00031-2.

[88] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI, 1975.

[89] Fred Glover. Heuristics for integer programming using surrogate constraints. *Decision sciences*, 8(1):156–166, 1977.

[90] Fred Glover. Tabu search and adaptive memory programming — advances, applications and challenges. In *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies*, pages 1–75. Springer US, Boston, MA, 1997. ISBN 978-1-4615-4102-8. doi: 10.1007/978-1-4615-4102-8_1. URL https://doi.org/10.1007/978-1-4615-4102-8_1.

[91] Mauricio G.C. Resende, Celso C. Ribeiro, Fred Glover, and Rafael Martí. Scatter search and path-relinking: Fundamentals, advances, and applications. In *Handbook of Metaheuristics*, pages 87–107. Springer US, Boston, MA, 2010. ISBN 978-1-4419-1665-5. doi: 10.1007/978-1-4419-1665-5_4.

[92] J. Wang, A.K.R. Jagannathan, X. Zuo, and C.C. Murray. Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries. *Computers & Industrial Engineering*, 112:84–98, 2017.

[93] Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987. doi: 10.1287/opre.35.2.254.

[94] Farhad Kolahan, Ahmad Tavakoli, Behrang Tajdin, and Modjtaba Hosayni. Analysis of neighborhood generation and move selection strategies on the performance of tabu search, 2006. URL https://www.researchgate.net/publication/255599657.

[95] Belarmino Adenso-Díaz. Restricted neighborhood in the tabu search for the flowshop problem. *European Journal of Operational Research*, 62(1):27–37, 1992. ISSN 0377-2217. doi: /10.1016/0377-2217(92)90174-8.

[96] J-F Cordeau, G Laporte, and A Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8):928–936, 2001. doi: 10.1057/palgrave.jors.2601163.

[97] Jean-François Cordeau, Gilbert Laporte, and Les Cahiers du GERAD. Tabu search heuristics for the vehicle routing problem, 3 2002.

[98] E. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17(4):443–455, 1991. ISSN 0167-8191. doi: 10.1016/S0167-8191(05)80147-4.

[99] Michel Gendreau, François Guertin, Jean-Yves Potvin, and Éric Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390, 1999. doi: 10.1287/trsc.33.4.381.

[100] Michel Gendreau, Fred Glover, and Gary A. Kochenberger. An introduction to tabu search. In *Handbook of Metaheuristics*, chapter 2, pages 37–54. Springer US, Boston, MA, 2003. ISBN 978-0-306-48056-0. doi: 10.1007/0-306-48056-5_2.

[101] Hanne Pollaris, Gerrit K Janssens, Kris Braekers, and An Caris. Parameter tuning of a local search heuristic for a vehicle routing problem with loading constraints. *Information Technology & Management Science*, 23, 2020.

[102] Holger H. Hoos. *Automated Algorithm Configuration and Parameter Tuning*, pages 37–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-21434-9. doi: 10.1007/978-3-642-21434-9_3.

[103] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[104] Vincent F. Yu, Pham Tuan Anh, and Roberto Baldacci. A robust optimization approach for the vehicle routing problem with cross-docking under demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 173:103106, 2023. ISSN 1366-5545. doi: 10.1016/j.tre.2023.103106.

[105] Scribbr. Reliability and validity: Definition, types, and examples, 2023. URL https://www.scribbr.com/frequently-asked-questions/reliability-and-validity/#:~:text=Reliability%20refers%20to%20the%20consistency,they%20are%20supposed%20to%20measure. Accessed: 2025-03-19.

[106] Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225 (1):1–11, 2 2013. ISSN 03772217. doi: 10.1016/j.ejor.2012.08.015.

[107] Ninja Soeffker, Marlin W. Ulmer, and Dirk C. Mattfeld. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review, 5 2022. ISSN 03772217.

[108] W. W. Garvin, H. W. Crandall, J. B. John, and R. A. Spellman. Applications of linear programming in the oil industry. *Management Science*, 3(4):407–430, 1957. doi: 10.1287/mnsc.3.4.407.

[109] Bezalel Gavish and Stephen C Graves. The traveling salesman problem and related problems, 1978. URL https://www.researchgate.net/publication/37596296.

[110] Luis Gouveia. A result on projection for the vehicle routing problem. *European Journal of Operational Research*, 85(3):610–624, 1995. ISSN 0377-2217. doi: 10.1016/0377-2217(94)00025-8.

[111] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300–304, 1964. doi: 10.1287/opre.12.2.300.

[112] Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115:351–385, 10 2008. ISSN 00255610. doi: 10.1007/s10107-007-0178-5.

A Mathematical Modelling Techniques

A.1 Two-index vehicle flow formulation

Let x_{ij} be defined as a binary variable that assumes value 9 if and only if there is a route that goes from customer i to j directly, for $i, j \in \mathcal{V}$. For any $\mathcal{S} \subset \mathcal{V}$, let $\delta^+(\mathcal{S})$ (respectively, $\delta^-(\mathcal{S})$) denote the set of arcs (i, j) with $i \in \mathcal{S}$, $j \in \mathcal{V} \backslash \mathcal{S}$ (respectively, with $i \in \mathcal{V} \backslash \mathcal{S}$, $j \in \mathcal{S}$). With the decision variable and the parameters as defined at the start of this section, the two-index vehicle flow formulation of the CVRP is given by the following integer linear programming formulation model:

$$\min \quad \sum_{i,j \in \mathcal{V}} c_{ij} x_{ij} \tag{A.1}$$

$$\text{subject to} \quad \sum_{j \in \delta^+(i)} x_{ij} = 1 \qquad \forall\, i \in \mathcal{V} \tag{A.2}$$

$$\sum_{i \in \delta^-(j)} x_{ij} = 1 \qquad \forall\, j \in \mathcal{V}$$

$$\sum_{j \in \delta^+(0)} x_{0j} = |\mathcal{K}| \tag{A.3}$$

$$\sum_{(i,j) \in \delta^+(\mathcal{S})} x_{ij} \geq r\,(S) \qquad \forall\, \mathcal{S} \subseteq \mathcal{V},\ \mathcal{S} \neq \emptyset \tag{A.4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, (i, j) \in \mathcal{A} \tag{A.5}$$

In this formulation, equation (A.1) depicts the objective function of the CVRP, aiming to minimise the travel costs while serving the demand of all customers $\mathcal{V}_c$. Constraints (A.2) ensure that all customers are served exactly once and safeguard the vehicle flow, ensuring that every vehicle arriving at a customer also departs from that customer. Furthermore, Constraint (A.3) ensures that exactly $|\mathcal{K}|$ routes are constructed. If more vehicles than needed are available (i.e., $|\mathcal{K}| > r\,(\mathcal{V})$), the equalities can be replaced with inequalities of type "$\leq$". Note that fleet size minimisation and routing costs are conflicting objectives, implying that a solution with $|\mathcal{K}| = r\,(\mathcal{V})$ may have higher routing costs than one where more routes are allowed. These objectives can be integrated by adding fixed costs for the routes, altering the cost coefficients c_{oi} [29]. Constraints (A.4) simultaneously serve as capacity constraints and subtour elimination constraints, ensuring that enough vehicles are used to satisfy the demand of the customers $\mathcal{S} \subseteq \mathcal{V}$. For more detailed explanations, one may refer to [10, 29, 41]. Lastly, Equations (A.5) are the integrality conditions on the x-variables.

A.2 Commodity-flow formulations

The first commodity-flow formulation was proposed by [108] in an oil delivery problem and later extended by [109]. It is an alternative to the vehicle flow formulations, which require an exponential number of constraints to enforce correct routing restrictions [40]. Unlike vehicle flow formulations, commodity flow formulations keep track of the load delivered.

Hence, they contain a set of continuous variables representing the flow of one or more commodities between the depot and the customers. This is in addition to the variables used in the vehicle flow formulations. The following explains the two-index one-commodity flow formulation. Note that in this formulation, vehicles are not differentiated; only one type of commodity is delivered from the depot to the customers.

According to [40, 109, 110], the two-index one-commodity flow is obtained by defining decision variables f_{ij} for all $i, j \in \mathcal{V}$, with the following interpretation: if $x_{ij} = 0$, then $f_{ij} = 0$, but if $x_{ij} = 1$, then f_{ij} represents the amount of load delivered by the vehicle when it leaves vertex i. Now, the one-commodity flow, mixed integer linear programming formulation is obtained by replacing Equations (1d) in the two-index vehicle flow formulation with:

$$f(\delta^+(\{i\})) = f(\delta^-\{i\}) + q_i \qquad \forall\, i \in \mathcal{V}_c \tag{A.6}$$

$$0 \leq f_{ij} \leq Qx_{ij} \qquad \forall\, i, j \in \mathcal{A} \tag{A.7}$$

Constraints (A.6) ensure that q_i units of flow are delivered at vertex i. Assuming that $q_0 = -\sum_{i \in \mathcal{V}_c} q_i$, then $-q_0$ units of flow are collected at the depot (in other words, the demand at node i is negative for the supply node). Constraints (A.7) are bounds on the f-variables.

A.3 Set-partitioning formulation

This formulation of the CVRP Balinski and Quandt [111], Baldacci et al. [112] associates a binary variable with each feasible route. In the set partitioning formulation, a variable is defined for each feasible route that a vehicle can take. Based on the formulation used by [40], consider Ω as the set of feasible routes. Each route r in Ω is associated with a binary variable z_r, which takes the value 1 if that route is selected and 0 otherwise. Additionally, constants a_{ir} are defined, with $a_{ir} = 1$ if customer i is served by route r, and $a_{ir} = 0$ otherwise. Let c_r represent the optimal cost associated with route r. Based on these definitions, the formulation can be expressed as integer linear programming formulation:

$$\min \quad \sum_{r \in \Omega} c_r z_r \tag{A.8}$$

$$\text{subject to} \quad \sum_{r \in \Omega} z_r = K \tag{A.9}$$

$$\sum_{r \in \Omega} a_{ir} z_r = 1 \qquad \forall\, i \in \mathcal{V}_c \tag{A.10}$$

$$z_r \in \{0, 1\} \qquad \forall\, r \in \Omega \tag{A.11}$$

Equation (A.8) depicts the objective function of the formulation. Similar to the other formulations, it aims to minimise the total cost incurred by visiting all customers exactly once. Constraint (A.9) guarantees that the number of feasible routes selected equal the number of vehicles available K. Also, if fewer than $(|\mathcal{K}|$ suffice to serve all customers, the equalities in (A.10) can be replaced with inequalities of type "$\leq$", and ensure that all customers are supplied by only one selected route. Finally, Equations (A.11) are the integrality conditions imposed on the binary z-variables. Note that a_{ir} represents constants and therefore does not require such an integrality definition.

B Toy Problem Coordinates

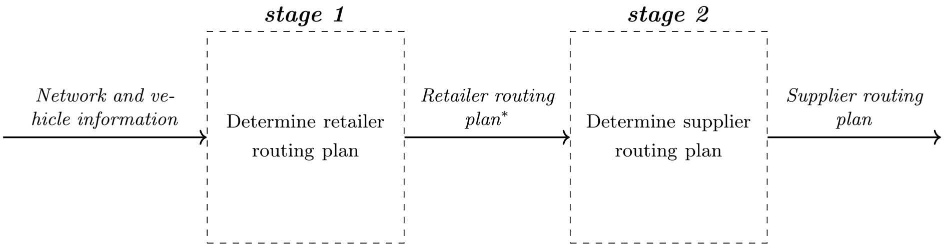
For conciseness, the coordinates of the graph used in the toy problem instance in Section 4.5 were omitted. Therefore, we include them here for reproducibility:

TABLE B.1: Toy problem node characteristics

Node	*x-coor*	*y-coor*
CD	10	10
s_1	6	8
s_2	7	12
r_1	11	8
r_2	14	9
r_3	15	11
r_4	12	13

C Initial Solution Generation

Given the interdependence between delivering goods to retailers and picking up goods from suppliers at the CD, the initial solution construction follows a two-stage approach. The first stage serves as input for the second stage. Figure C.1 summarises the two stages and their relationship. Note that the greedy insertion heuristic is similar in both stages, although the retailer routing plan serves as input for the supplier routing plan. The greedy heuristic is detailed in Algorithm 2.



* *The departure times from the retailer routing plan are essential for calculating the latest allowable arrival times of goods at the CD, factoring in handling time at the CD.*

FIGURE C.1: Outline of the two-stage approach employed in the constructive heuristic

C.1 Stage 1: Retailer/Delivery Routes

The input for the first stage includes network and vehicle data, such as node locations, demands, travel times, and vehicle capacities. The greedy insertion heuristic generates a feasible routing plan from the CD to all retailers and back, iterating to find the insertion with the least incremental cost. Consequently, the insertion that minimises the increase in travel time is selected as the optimal choice in each iteration.

C.2 Stage 2: Supplier/Pick Up Routes

The output of the first stage serves as the *input* for the second stage, where the routing for servicing suppliers is determined. In this stage, restrictions on arrival times at the CD are considered, as certain goods must be loaded onto vehicles that depart at predetermined times. Consequently, it is necessary to ensure that goods arrive at the CD before their respective outbound vehicles finish loading. In addition to this temporal constraint, network information (i.e., node locations, demands, travel times) and vehicle characteristics (i.e., capacity, vehicle availability) also serve as inputs for the second stage.

Changes in the procedure for composing the routes in the second stage, compared to the first stage are:

- UnvisitedLocations is set to $\mathcal{S}$ to route the vehicles to the suppliers. Thus, UnvisitedLocations $\leftarrow \mathcal{S}$.

Algorithm 2: Two-stage greedy insertion heuristic

Data: $\mathcal{R}$, $\mathcal{S}$, $\mathcal{V}$, Q, D_{rs} for all $r \in \mathcal{R}, s \in \mathcal{S}, t_{ij}$ for all $i, j \in \mathcal{N}$

Result: Constructed routes from suppliers to CD to retailers

Initialise: Routes $\leftarrow \{\varnothing\}$, UnvisitedLocations $\leftarrow$ Locations of unmet demands (stage 1) or suppliers (stage 2), UnvisitedOrders $\leftarrow$ Corresponding unmet demands, AvailableVehicles $\leftarrow \mathcal{V}$

while *UnvisitedOrders* **do**
- $v \leftarrow Rnd(AvailableVehicles)$ ▷ Random vehicle selection
- **while** *True* **do** ▷ Infinite loop for insertion
  - **Initialise:** BestCost $\leftarrow \infty$, BestPosition $\leftarrow$ null, Location $\leftarrow$ null
  - **for** *index*, $m \in$ *enumerate(UnvisitedLocations)* **do** ▷ Loop over unvisited locations
    - **for** $p \in$ *CurrentRoute* **do** ▷ Loop over potential insertion positions
      - InsertionCost$(m, p) \leftarrow t_{i_{p-1}m} + t_{mi_p} - t_{i_{p-1}i_p}$
      - **if** *InsertionCost* $<$ *BestCost* ***and*** *TimeWindowsSatisfied* ***and*** $\hat{Q}_v - UnvisitedOrders[index] \geq 0$ **then**
        - BestCost $\leftarrow$ InsertionCost, BestPosition $\leftarrow p$, Bestocation $\leftarrow m$, BestDemandIndex $\leftarrow$ index
    - **if** *BestPosition NOT null* **then** ▷ Check for feasible insertion
      - InsertLocation(CurrentRoute, Bestlocation, BestPosition)
      - UpdateRemainingCapacity: $\hat{Q}_v \leftarrow \hat{Q}_v - UnvisitedOrders[BestDemandIndex]$
      - Remove(UnvisitedLocations, BestDemandIndex), Remove(UnvisitedOrders, BestDemandIndex) ▷ Remove met demand and location
    - **else**
      - $AvailableVehicles.remove(v)$
      - Routes$[v] \leftarrow$ CurrentRoute ▷ Store current route
      - Determine times for this route
      - break ▷ No possible insertions, exit loop
    - **if** *NOT Bestlocation or UnvisitedOrders* $== \varnothing$ **then**
      - break

return Routes ▷ Return constructed routes

- Ensure that all goods picked up at suppliers arrive in time at the CD. The steps taken to evaluate the latest allowable time at which a vehicle can return to the CD are outlined by Algorithm 3, which computes these times for any given route that services a set of suppliers. Note that this procedure should be called when checking the time window of any potential route if an insertion is evaluated.

Algorithm 3: Procedure for computing latest allowable time of arrival

Data: Route, $\mathcal{R}$, D_{rs} for all $r \in \mathcal{R}, s \in \mathcal{S}$
Result: Latest allowable time for a vehicle to arrive at the CD
Initialise: LatestAllowableTimes $\leftarrow [\varnothing]$
for *supplier* $\in$ *Route* **do**
    for *retailer* $\in \mathcal{R}$ ***where*** $D_{retailer,supplier} > 0$ **do**
        for *vehicle* $\in$ *RetailerRoutes* ***where*** *retailer* $\in$ *RetailerRoutes*[*vehicle*] **do**
            DepartureTime $\leftarrow$ DepartureTimes[vehicle] [0]
            LatestAllowableTime $\leftarrow$ DepartureTime - HandlingTime
            LatestAllowableTimes.append(LatestAllowableTime)
return min{LatestAllowableTimes} ▷ Return minimum latest allowable time

D Data Filtering Process

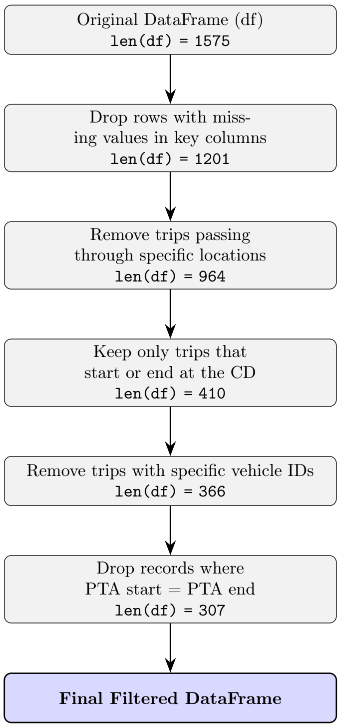


Figure D.1: Data Filtering Process