Faculdade de Engenharia da Universidade do Porto

Multi-Objective Optimization of Container Transportation in Multi-Depot Logistics Networks

Sebastião Lucena e Vale de Sousa Pinto



Master in Mechanical Engineering
Production Engineering & Management

Supervisor: Catarina Moreira Marques

Supervisor: Jorge Pinho de Sousa

July 3, 2026

Multi-Objective Optimization of Container Transportation in Multi-Depot Logistics Networks

Sebastião Lucena e Vale de Sousa Pinto

Master in Mechanical Engineering
Production Engineering & Management

July 3, 2026

Abstract

As the complexity of global supply chains grows, maritime transport, and container terminals in particular, are under increasing pressure to more effectively connect sea-side operations and land-side distribution. In this context, hinterland container transportation planning and the dispatching of container trucks represents a key operational challenge. This challenge is even greater because modern container networks typically encompass multiple geographically distributed terminals and depots, with vehicles and containers spread across various yards and inland facilities.

This work, developed within the NEXUS project and motivated by the hinterland network of the Port of Sines, tackles this problem by proposing a bi-objective Mixed-Integer Linear Programming model that jointly minimizes total transportation time and total transportation cost. The model determines vehicle-to-task assignments, routing sequences, and the timing of every operation, while accounting for a heterogeneous fleet of conventional and electric trucks, multiple depots, time windows at every node, driver working time regulations, hazardous material certification requirements, and the simultaneous management of full and empty containers. The ε-constraint method is used to generate the Pareto front between the two objectives, giving decision-makers a full picture of the available trade-offs rather than a single solution.

For a representative 10-client instance, the resulting Pareto front comprises 24 non-dominated solutions, ranging from a time-optimal solution using 6 conventional vehicles to a cost-optimal solution that deploys the entire available electric fleet across 8 vehicles in total, confirming that minimizing cost favors electric vehicles despite their lower speed, at the expense of fleet size and route consolidation. Scalability experiments show that the model can be solved to a tight optimality gap for instances of up to approximately 15 clients, beyond which solution quality degrades sharply. A sensitivity analysis further shows that tightening delivery time windows is the most computationally demanding operational factor examined, while a more restrictive vehicle certification ratio allowed the model to be solved faster.

Keywords: Global supply chains, Logistics Networks, Container Terminals, Truck scheduling, Multi-objective optimization

Resumo

À medida que a complexidade das cadeias de abastecimento globais aumenta, o transporte marítimo, e os terminais de contentores em particular, estão sob pressão crescente para ligar de forma mais eficiente as operações do lado do mar às redes de distribuição terrestre. Neste contexto, o planeamento do transporte terrestre de contentores ("hinterland") e a programação dos camiões de contentores constitui um desafio operacional fundamental. Este desafio é ainda maior porque as redes modernas de contentores normalmente abrangem múltiplos terminais e depósitos geograficamente distribuídos, com veículos e contentores dispersos por diversos parques e instalações no interior do território.

Este trabalho, desenvolvido no âmbito do projeto NEXUS e motivado pela rede de hinterland do Porto de Sines, aborda este problema através da proposta de um modelo de Programação Linear Inteira Mista biobjetivo que minimiza simultaneamente o tempo total de transporte e o custo total de transporte. O modelo determina a atribuição de tarefas a veículos, as sequências de rotas e o instante de cada operação, considerando uma frota heterogénea de camiões convencionais e elétricos, múltiplos depósitos, janelas temporais em todos os nós, regulamentação do tempo de trabalho dos condutores, requisitos de certificação para o transporte de mercadorias perigosas, e a gestão simultânea de contentores cheios e vazios. O método de restrição-ε é utilizado para gerar a fronteira de Pareto entre os dois objetivos, fornecendo aos decisores uma visão completa dos compromissos disponíveis, em vez de uma única solução.

Para uma instância representativa de 10 clientes, a fronteira de Pareto obtida é composta por 24 soluções não dominadas, que vão desde uma solução ótima em tempo, que utiliza 6 veículos convencionais, até uma solução ótima em custo, que utiliza toda a frota elétrica disponível, num total de 8 veículos, confirmando que a minimização do custo favorece os veículos elétricos, apesar da sua menor velocidade, à custa da dimensão da frota e da consolidação das rotas. As experiências de escalabilidade mostram que o modelo pode ser resolvido com um gap de otimalidade reduzido para instâncias com até aproximadamente 15 clientes, acima do qual a qualidade das soluções degrada-se acentuadamente. Uma análise de sensibilidade mostra ainda que o estreitamento das janelas temporais de entrega é o fator operacional computacionalmente mais exigente entre os analisados, enquanto um rácio de certificação de veículos mais restritivo permitiu, surpreendentemente, resolver o modelo mais rapidamente.

Palavras-chave: Cadeias de Abastecimento Globais, Logística de Hinterland, Transporte de Contentores, Problema de Roteamento com Múltiplos Depósitos, Otimização Multiobjetivo

Acknowledgments

I would like to thank my supervisor, Professor Catarina Moreira Marques, for her help and guidance throughout the development of this thesis.

I would also like to thank Professor Jorge Pinho de Sousa for his valuable advice throughout my time at INESC TEC.

I am also grateful to INESC TEC for welcoming me during this period and for providing the environment and resources that made this work possible.

Finally, I would like to thank my family and friends for their constant support and patience throughout this journey.

Sebastião Sousa Pinto

"All models are wrong,
but some are useful"

George E. P. Box

Contents

1 Introduction **1**
- 1.1 Motivation and Industrial Context . . . 1
- 1.2 Research Objectives . . . 3
- 1.3 Methodology . . . 4
- 1.4 Structure . . . 4

2 State of the Art **7**
- 2.1 The Vehicle Routing Problem and Its Multi-Depot Extension . . . 7
- 2.2 Mathematical Programming Formulations . . . 8
- 2.3 Solution Approaches . . . 9
  - 2.3.1 Exact Methods . . . 9
  - 2.3.2 Heuristics and Metaheuristics . . . 10
  - 2.3.3 Multi-Objective Optimization Approaches . . . 10
- 2.4 Problem Variants and Extensions . . . 11
  - 2.4.1 Arc Routing . . . 12
  - 2.4.2 Multi-Objective Extensions . . . 12
  - 2.4.3 Optimization under Uncertainty . . . 13
  - 2.4.4 Time Windows and Service Quality . . . 14
- 2.5 Emerging Trends: Machine Learning for Routing . . . 14
- 2.6 Summary and Research Gaps . . . 15

3 Problem Definition **17**
- 3.1 Operational Context . . . 17
  - 3.1.1 Multi-Depot Structure . . . 19
  - 3.1.2 Heterogeneous Fleet . . . 20
  - 3.1.3 Full and Empty Containers . . . 20
  - 3.1.4 Hazardous Materials . . . 20
  - 3.1.5 Time Windows . . . 21
  - 3.1.6 Driver Working Time Regulations . . . 21
- 3.2 Problem Description and Assumptions . . . 21

4 Mathematical Model **25**
- 4.1 Network Nodes . . . 25
- 4.2 Parameters . . . 26
- 4.3 Decision Variables . . . 28
- 4.4 Objective Functions . . . 28
- 4.5 Constraints . . . 29
  - 4.5.1 Routing and Flow Conservation . . . 29

4.5.2 Time Constraints . . . 30
4.5.3 Hazardous Material and Capacity Constraints . . . 31
4.5.4 Container Load Flow Constraints . . . 31
4.5.5 Load State Variable Constraints . . . 32
4.6 Multi-Objective Solution Strategy . . . 33

5 Computational Experiments and Results **35**
5.1 Computational Setup . . . 35
5.2 Instance Generation . . . 36
5.3 Pareto Front Generation . . . 39
5.4 Scalability Analysis . . . 40
5.5 Sensitivity Analysis . . . 41
5.5.1 Base Instance and Experimental Design . . . 42
5.5.2 Time Window Tightness . . . 42
5.5.3 Hazardous Material Proportion . . . 43
5.5.4 Vehicle Certification . . . 43
5.5.5 Fleet Size . . . 44
5.6 Computational Results and Discussion . . . 45
5.6.1 Approximation to the Pareto front . . . 46
5.6.2 Scalability . . . 49
5.6.3 Sensitivity Analysis . . . 53
5.6.4 Synthesis . . . 55

6 Conclusion **57**
6.1 Summary and Contributions . . . 57
6.2 Sustainable Development Goals . . . 59
6.3 Limitations . . . 60
6.4 Future Work . . . 61

References **63**

A Instance Data **67**
A.1 Cumulative Instance Sequence . . . 67
A.2 Independent Instances . . . 69
A.3 Example Solution Routes . . . 72

B Full Pareto Front Results **73**
B.1 Cumulative Instance Sequence . . . 73
B.2 Independent instances . . . 77

C Conference Submission **83**

List of Figures

3.1 Schematic representation of a container terminal and its hinterland distribution chain, from quayside to final clients via dry ports and logistics platforms, adapted from Steenken et al. (2004). 17
3.2 Hinterland distribution network of the Port of Sines, showing maritime links, rail corridors, and logistics platforms across the Iberian Peninsula (APS – Ports of Sines and the Algarve Authority, 2026). 18
3.3 Standard ISO shipping container sizes: 20-foot (TEU) and 40-foot (FEU). Source: based on ISO 668:2020 (International Organization for Standardization, 2020). 19
3.4 Port logistics network: depots, ports, and clients with container transport flows. 23

5.1 Bounding rectangle approximating the Portuguese territory, with coordinates $(x, y) \in [0, 210] \times [0, 550]$. 36
5.2 Pareto front between total transportation time and total transportation cost, 10-client instance A. 46
5.3 Number of conventional and electric vehicles used along the Pareto front, 10-client instance A. 47
5.4 Vehicle routes for the time-optimal and cost-optimal solutions, 10-client instance A. Squares denote depots, triangles denote ports, circles denote clients. Dashed lines indicate electric vehicles. 48
5.5 Optimality gap of all non-dominated solutions, by number of clients. 50
5.6 Pareto fronts of the cumulative-sequence instance, by number of clients. . . 51

List of Tables

3.1 Problem-specific features of the container truck routing problem. 19

5.1 Fixed coordinates and processing times for ports and depots. 38
5.2 Modified time windows, Time Window Tightness variant. 43
5.3 Modified hazardous material requirements, Hazardous Material Proportion variant. 44
5.4 Vehicle certification and propulsion type, Certification Ratio variant. 45
5.5 Vehicle certification and propulsion type, Fleet Size variant. 45
5.6 Endpoint solutions of the Pareto front, 10-client instance A. 47
5.7 Model size statistics for the cumulative instance sequence. 50
5.8 Variability of solve difficulty across independent instances of the same size. 52
5.9 Sensitivity analysis: objective values, gap, and run time by variant. 53
5.10 Sensitivity analysis: fleet composition by variant. 54

6.1 Alignment of this thesis with the United Nations Sustainable Development Goals. 59

A.1 Client node data for the cumulative instance sequence. 68
A.2 Client node data, 5-client instance A. 69
A.3 Client node data, 5-client instance B. 69
A.4 Client node data, 10-client instance A. 69
A.5 Client node data, 10-client instance B. 70
A.6 Client node data, 15-client instance A. 70
A.7 Client node data, 15-client instance B. 71
A.8 Vehicle routes, time-optimal solution, 10-client instance A. 72
A.9 Vehicle routes, cost-optimal solution, 10-client instance A. 72

B.1 Full Pareto front, 5-client cumulative. 73
B.2 Full Pareto front, 8-client cumulative. 73
B.3 Full Pareto front, 10-client cumulative. 74
B.4 Full Pareto front, 12-client cumulative. 74
B.5 Full Pareto front, 15-client cumulative. 75
B.6 Full Pareto front, 20-client cumulative. 76
B.7 Full Pareto front, 5-client instance A. 77
B.8 Full Pareto front, 5-client instance B. 77
B.9 Full Pareto front, 10-client instance A. 77
B.10 Full Pareto front, 10-client instance B. 78
B.11 Full Pareto front, 15-client instance A. 79
B.12 Full Pareto front, 15-client instance B. 80

Acronyms list

ACO Ant Colony Optimization

ADR European Agreement concerning the International Carriage of Dangerous Goods by Road

AFV Alternative Fuel Vehicle

BTS Bureau of Transportation Statistics

CVRP Capacitated Vehicle Routing Problem

DFJ Dantzig-Fulkerson-Johnson

DSS Decision Support System

FEU Forty-foot Equivalent Unit

GAT Graph Attention Network

GVRP Green Vehicle Routing Problem

MDARPFED Multi-Depot Arc Routing Problem with Flexible End Depot

MDHVRP Multi-Depot Heterogeneous Vehicle Routing Problem

MDOVRP Multi-Depot Open Vehicle Routing Problem

MDVRP Multi-Depot Vehicle Routing Problem

MILP Mixed-Integer Linear Programming

MIP Mixed Integer Programming

MOGA Multi-Objective Genetic Algorithm

MOGVRPTW Multi-Objective Green Vehicle Routing Problem with Time Windows

MOILS Multi-Objective Iterated Local Search

MOPSO Multi-Objective Particle Swarm Optimization

MTZ Miller-Tucker-Zemlin

NP Nondeterministic Polynomial (time)

NSGA-III Non-Dominated Sorting Genetic Algorithm III

SPD-MDVRP Simultaneous Pickup and Delivery Multi-Depot Vehicle Routing Problem

TEU Twenty-foot Equivalent Unit

VNS Variable Neighborhood Search

VRP Vehicle Routing Problem

Chapter 1

Introduction

This chapter introduces the work developed in this thesis. Section 1.1 describes the motivation and industrial context that led to this research. Section 1.2 states the research objectives. Section 1.3 outlines the methodology followed. Section 1.4 describes the structure of the remainder of the thesis.

1.1 Motivation and Industrial Context

This dissertation is developed by Sebastião Sousa Pinto, in the second semester of the academic year 2025/2026, in the context of the Master's degree in Mechanical Engineering, in the area of Production Management, under the supervision of Professors Catarina Moreira Marques and Jorge Pinho de Sousa.

The way goods move around the world has changed dramatically in recent decades. Containers have become the backbone of international trade, and the ports that handle them are under constant pressure to do more, faster, and at lower cost. Much of the attention in port research and investment has gone to what happens at the waterfront, loading and unloading ships, coordinating cranes, managing schedules. But once a container leaves the terminal gate, a whole separate logistical challenge begins, one that is arguably just as complex and considerably less studied.

This landward side of port operations is known as hinterland logistics. It covers everything that happens between the port and the final destination of a container, whether that is a warehouse, a factory, an inland terminal, or a distribution center. In practice, this journey typically unfolds in two steps. First, containers travel from the port to one of a set of inland terminals, by train or by truck, depending on what is available and what makes sense given the timing and cost. From those inland terminals, smaller truck fleets then take over, delivering containers directly to clients or collecting them for the return trip. Although each transportation task is relatively straightforward when considered in

isolation, coordinating large numbers of such tasks across multiple terminals, heterogeneous vehicle fleets, different operators, and time-sensitive delivery schedules gives rise to a highly complex operational planning problem.

In this regard, the Port of Sines provides a clear real-world example of the operational complexity associated with container transportation planning. It is the largest deep-water container terminal in Portugal and one of the most important Atlantic ports in Europe, handling traffic that connects South America, Africa, and Asia to European markets. Its hinterland network reaches across Portugal and into Spain, passing through inland terminals at Bobadela, Entroncamento, and Porto (APS – Ports of Sines and the Algarve Authority, 2026). Every day, decisions have to be made about which trucks go where, which containers they carry, when they depart, and how to bring empty containers back to where they are needed. These decisions are often made with limited analytical support, and poor planning can result in delays, unnecessary trips, and increased operational costs.

What makes the problem particularly difficult is that it involves a large number of interacting constraints that must all be respected simultaneously. Drivers are subject to legal working time limits. Containers can be full or empty, and managing both types correctly, including repositioning empty containers to loading points, is essential to avoid unnecessary empty travel and wasted capacity. Tasks come with time windows imposed by terminals and clients. Vehicles operate out of multiple depots spread across the network. And all of this has to be coordinated daily, under time pressure, with the information available at the start of each planning cycle. A formal characterization of the problem is provided in Chapter 3.

This thesis was developed within the NEXUS project (PRR NEXUS project (Reference 53)), a collaborative initiative involving the Port of Sines Authority, infrastructure operators, transport companies, and several Portuguese universities. The project aims to develop decision-support tools to improve the efficiency and sustainability of port and hinterland logistics operations. Within this context, the present work focuses on the truck distribution problem, namely the routing and scheduling of container trucks operating between ports and the containers' final clients. To address this problem, a multi-objective optimization approach is proposed and formulated as a Mixed-Integer Linear Programming (MILP) model.

The multi-objective approach is particularly appropriate for this problem because different performance criteria lead to fundamentally different operational decisions. Solutions that minimize total travel time may differ substantially from those that minimize total transportation cost, and improving one objective often comes at the expense of the other. Consequently, decision-makers must evaluate the trade-offs between competing objectives rather than focusing exclusively on a single performance measure. These trade-offs become increasingly significant as the scale and complexity of the transportation network grow. The main objective of this thesis is therefore to characterize these trade-offs and provide a decision-support framework capable of identifying efficient routing and scheduling

solutions considering different (and possibly conflicting) objectives.

A preliminary version of the work presented in this thesis has been submitted to and accepted for presentation at the Optimization 2026 conference, organized by APDIO, the Portuguese Operations Research Society, to be held in Lisbon, Portugal, in July 2026. The accepted abstract is reproduced in Appendix C.

1.2 Research Objectives

The central objective of this thesis is to develop a decision support system for the operational planning of container truck fleets in port hinterland networks, capable of simultaneously optimizing two competing performance criteria: total transportation time and total transportation cost. To achieve this, the work pursues a set of more specific objectives that together define the scope and contribution of the research.

The first objective is to review the relevant literature and characterize the problem in its operational and academic context. This means surveying the main families of vehicle routing problems and their extensions, identifying the solution methods that have been applied to similar problems, and mapping the research gaps that motivate the contributions of this thesis.

The second objective is to build a mathematical model that accurately captures the operational reality of the problem. This means going beyond simplified routing formulations and explicitly accounting for the features that make container drayage genuinely difficult in practice, namely the simultaneous management of full and empty containers, the heterogeneity of the vehicle fleet, the presence of hazardous material constraints, time windows at every node, driver working time regulations, and the multi-depot structure of the network. The result is a Mixed-Integer Linear Programming formulation that is both faithful to the real problem and tractable enough to produce useful solutions on instances of meaningful size.

The third objective is to frame the planning problem as a multi-objective optimization task and to generate the Pareto front between the two objectives of interest. Rather than collapsing the problem into a single weighted criterion and producing a single solution, the aim is to map out the full range of trade-offs available to a decision-maker, showing clearly what must be sacrificed in terms of cost to gain a given improvement in time, and vice versa. This provides a richer and more honest picture of the problem than any single-objective approach could offer.

The fourth objective is to analyze and interpret the results in a way that is meaningful from an operational standpoint. Computational experiments are designed not only to validate the model but to extract practical insights about how different problem parameters, such as fleet composition, network structure, and time window tightness, affect the shape and range of the trade-off between time and cost. The goal is that the findings should be

useful not just to researchers but to logistics planners and fleet managers working in real hinterland contexts.

The fifth and final objective is to contribute, within the broader NEXUS project framework, a well-documented and reusable optimization module that can serve as a foundation for future extensions of the decision support system, whether those extensions involve larger instances, additional objectives, stochastic demand, or integration with the upstream intermodal planning problem.

1.3 Methodology

The work developed in this thesis followed a Operations Research methodology consisting of problem characterization, mathematical modeling, computational optimization, and experimental evaluation. The starting point was a thorough review of the specialized academic literature, covering the main families of vehicle routing problems, multi-objective optimization methods, and the specific characteristics of container drayage and port hinterland logistics. This review provided the theoretical foundation on which the mathematical model was built and helped identify the gaps that this work aims to address.

In parallel, the problem was grounded in a real industrial context through the NEXUS project, a collaborative initiative involving the Port of Sines authority, infrastructure operators, transport companies, and several Portuguese universities. The project provided the operational framing for this work, including a clear understanding of the structure of the hinterland network, the nature of the transportation tasks performed, and the constraints that logistics operators face on a daily basis. This industrial context ensured that the modeling choices made throughout the thesis reflect genuine operational needs rather than purely theoretical assumptions. Although inspired by the operational characteristics of the Port of Sines hinterland network, the computational experiments are conducted on synthetically generated instances to ensure controlled and reproducible testing across a range of operational scenarios while addressing the limited availability of real operational data.

The solution approach is based on mathematical programming. The problem is formulated as a Mixed-Integer Linear Programming model and solved using the Gurobi Optimizer. To handle the multi-objective nature of the problem, the epsilon-constraint method is used to generate the Pareto front between total transportation time and total transportation cost. Computational experiments are conducted on synthetically generated instances and the results are analyzed both in terms of computational performance and practical interpretability.

1.4 Structure

The remainder of this thesis is organized as follows.

Chapter 2 reviews the relevant literature, starting from the classical Vehicle Routing Problem and progressively moving toward the more complex variants that are closest to the problem addressed here, including multi-depot routing, time windows, and green routing. It also covers the main solution methods found in the literature, both exact and heuristic, and closes by identifying the research gaps that motivated this work.

Chapter 3 presents a formal characterization of the problem addressed in this thesis, describing the operational context, the entities involved, and the assumptions underlying the model. It provides the conceptual foundation on which the mathematical formulation of Chapter 4 is built.

Chapter 4 presents the mathematical model developed to tackle the problem. It introduces the full Mixed-Integer Linear Programming formulation, describing the sets, parameters, decision variables, objective functions, and constraints in detail, along with the reasoning behind the main modeling choices.

Chapter 5 covers both the implementation of the model and the computational results. It begins by describing the experimental setup, the instance generation procedure, and the strategy used to generate the Pareto front through the epsilon-constraint method. It then presents and discusses the results of the computational experiments, analyzing the trade-offs between transportation time and cost, examining how different problem parameters influence the shape of the Pareto front, and drawing practical conclusions relevant to logistics planners operating in hinterland contexts.

Chapter 6 closes the thesis with a summary of the main contributions, a critical reflection on the limitations of the work, and a set of directions for future research that naturally follow from what was accomplished here.

Chapter 2

State of the Art

This chapter reviews the literature relevant to the problem addressed in this thesis. It begins with the classical Vehicle Routing Problem and its multi-depot extension 2.1, followed by a discussion of the main mathematical formulations proposed for this class of problems 2.2. It then reviews the solution approaches found in the literature, covering exact methods, heuristics and metaheuristics, and multi-objective optimization techniques 2.3. The chapter proceeds to examine the main problem variants and extensions that are closest to the problem addressed here, including arc routing, multi-objective applications, optimization under uncertainty, and time windows with service quality considerations 2.4. It then discusses emerging machine learning approaches to routing problems 2.5. The chapter closes with a summary of the literature reviewed and an identification of the research gaps that motivate the contributions of this thesis 2.6.

2.1 The Vehicle Routing Problem and Its Multi-Depot Extension

Physical distribution is one of the most expensive and operationally demanding components of any logistics system, and the Vehicle Routing Problem (VRP) lies at the heart of efforts to optimize it. Originally introduced by Dantzig and Ramser (1959), the VRP asks how a fleet of vehicles, departing from a single depot, should be routed to serve a set of customers with known demands while respecting vehicle capacity and minimizing total travel cost. Despite its straightforward statement, the problem is classified as NP-hard, so the computational effort required to guarantee an optimal solution grows rapidly with instance size, rendering exact methods impractical for all but the smallest cases (Montoya-Torres et al., 2015).

Most real distribution networks, however, are not built around a single warehouse. Companies typically operate several facilities spread across a region, which gives rise to the Multi-Depot Vehicle Routing Problem (MDVRP). Each vehicle must begin and end its tour at the same depot, and every customer is assigned to exactly one facility. This

additional assignment dimension makes the MDVRP substantially harder than its single-depot counterpart, since the algorithms and formulations developed for the classical VRP cannot be straightforwardly extended and new modeling strategies must be designed from the ground up (Montoya-Torres et al., 2015; Baldacci et al., 2004).

2.2 Mathematical Programming Formulations

Several mathematical formulations have been proposed for the MDVRP, each reflecting different trade-offs between model strength and computational tractability. The main families in the literature are two-index vehicle flow formulations, three-index vehicle flow formulations, two-commodity flow formulations, and set partitioning formulations (Baldacci et al., 2004; Golden et al., 1977).

Baldacci et al. (2004) introduced a two-commodity flow formulation for the capacitated VRP in which each route is represented by two complementary flows along every arc: one tracking the cargo already carried, the other the remaining empty capacity. Together, these flows enforce capacity constraints and produce linear relaxations that are tighter than those of classical three-index models. Montoya-Torres et al. (2015) subsequently extended this framework to the multi-depot setting by adding a third index for vehicle type, introducing depot copies to handle multiple facilities, and incorporating a binary variable that links each vehicle to its assigned customers. The resulting formulation is better suited to branch-and-bound procedures precisely because its relaxation is stronger.

The three-index formulation of Golden et al. (1977) remains widely used. Montoya-Torres et al. (2015) adapted it for the multi-depot context by partitioning vehicles according to their home depot and adding constraints that prevent them from operating outside their assigned facility. A challenge shared by all these models is the elimination of subtours, partial cycles that satisfy flow conservation on paper but carry no operational meaning. Four approaches have been studied in depth: Dantzig, Fulkerson and Johnson (DFJ) constraints (Dantzig et al., 1954), which produce tight relaxations but may require an exponential number of inequalities; Miller, Tucker and Zemlin (MTZ) constraints (Miller et al., 1960), which are compact but weaker; transit load constraints based on the cargo carried along each arc; and arrival time constraints that exploit temporal progression to exclude cycles. Montoya-Torres et al. (2015) tested five variants of the three-index formulation against their two-commodity flow model on standard benchmark instances and found competitive solution quality at considerably lower computational cost.

More recently, Lalla-Ruiz and Mes (2021) extended the formulation landscape to the Multi-Depot Open Vehicle Routing Problem (MDOVRP), in which vehicles are not required to return to their starting depot after completing their routes. They developed a new two-index integer programming model with improved subtour elimination and flow conservation constraints, comparing it against existing three-index and alternative two-index formulations using CPLEX. Their analysis showed that the choice of formulation

has a pronounced effect on solver performance, particularly for larger instances, and that alternative flow constraints and load-tracking variables each offer distinct computational advantages depending on instance characteristics.

The model developed in this thesis builds on the two-commodity flow formulation introduced by Baldacci et al. (2004), extended to explicitly track full and empty container flows separately rather than a single aggregate commodity, and further adapted to incorporate multiple depots, time windows, and heterogeneous vehicle characteristics within the routing and load-tracking structure.

2.3 Solution Approaches

This section reviews the main solution approaches found in the literature for the MDVRP and its variants, covering exact methods 2.3.1, heuristics and metaheuristics 2.3.2, and multi-objective optimization techniques 2.3.3.

2.3.1 Exact Methods

Exact solution methods, including Mixed Integer Linear Programming (MILP), branch-and-bound, branch-and-cut, and column generation, are capable of certifying optimality and computing provable bounds on solution quality. In practice, however, their applicability is limited to small and medium-sized instances. Montoya-Torres et al. (2015) estimated that exact approaches account for roughly 25% of the MDVRP literature, with branch-and-bound methods developed for both symmetric and asymmetric problem variants, and set-partitioning-based branch-and-cut-and-price schemes capable of handling instances with up to 200 customers.

A compelling real-world application of exact modeling is the work of Mogale et al. (2025), who built a Mixed Integer Programming (MIP) model for a UK retailer seeking to optimize its northern distribution network without opening new facilities. The case study, carried out for a company referred to as APN (Alpha Partnership Network), covered a network of 51 Alpha & Partners stores, 349 Beta & Partner stores, 21 customer delivery hubs, 11 distribution centers, cross-docking facilities, transportation hubs, and third-party logistics providers. The model incorporated driver costs, inter-depot routing, a heterogeneous fleet, dynamic demand, multiple product types, and route duration constraints, a combination the authors noted was underrepresented in the literature, since fewer than 10% of studies published between 2009 and 2017 explicitly addressed multi-depot configurations. The solver achieved a reduction of approximately £1.2 million in annual transport costs by recommending route consolidation between distribution centers and identifying the most cost-efficient changeover point in the network, while confirming that the company's internal fleet was sufficient to meet demand without recourse to third-party services.

2.3.2 Heuristics and Metaheuristics

Given the computational intractability of exact methods for large instances, the bulk of the MDVRP literature relies on heuristic and metaheuristic approaches. Montoya-Torres et al. (2015) found that approximately 75% of published work falls into this category, ranging from classical constructive heuristics such as the savings method to sophisticated population-based and neighborhood search algorithms. Common metaheuristics applied to the MDVRP include Tabu Search, Genetic Algorithms, Simulated Annealing, Variable Neighborhood Search (VNS), Ant Colony Optimization (ACO), and record-to-record travel. Hybrid and matheuristic variants that combine exact solver components with metaheuristic search have also been developed to exploit the complementary strengths of both paradigms.

A representative example of multi-objective metaheuristic design is the work of Galindres-Guancha et al. (2018), who proposed a Multi-Objective Iterated Local Search (MOILS) for a bi-objective MDVRP aimed at simultaneously minimizing total distance and the standard deviation of route lengths, the latter serving as a practical measure of operational fairness and workload balance. The algorithm intensifies its search using VNS with both intra-route and inter-route neighborhood moves, maintaining and updating the Pareto front throughout the procedure. Their results showed that achieving a more balanced workload distribution does not need to come at a prohibitive cost in total distance.

Surekha and Sumathi (2011) proposed a Multi-Objective Genetic Algorithm (MOGA) for the Simultaneous Pickup and Delivery MDVRP (SPD-MDVRP), an extension in which each customer location may involve both a delivery and a pickup requirement that must be handled within the same route. Solutions are encoded as chromosomes representing depot assignments and visit sequences, Pareto-front selection identifies non-dominated individuals, and specialized crossover and mutation operators handle the routing structure. Experiments conducted on two-depot instances with 100 iterations confirmed that the multi-objective formulation, which minimizes total distance and number of vehicles simultaneously, outperformed a single-objective genetic algorithm, with the Pareto-front approach providing decision-makers with a meaningful way to navigate the trade-off between resource utilization and service coverage.

2.3.3 Multi-Objective Optimization Approaches

Solving a multi-objective optimization problem requires a strategy for handling the conflicting nature of the objectives, since no single solution generally optimizes all criteria simultaneously. Several classical scalarization techniques have been developed to transform a multi-objective problem into one or more single-objective problems that can be solved with standard mathematical programming tools.

The weighted sum method, introduced by Zadeh (1963), combines all objectives into a single function by multiplying each one by a user-defined weight and summing the results.

By systematically varying the weights and re-solving the resulting single-objective problem, a set of Pareto-optimal solutions can be approximated (Miettinen, 1999). This approach has been applied directly within a Mixed-Integer Linear Programming context by Song et al. (2020), who studied a real-life multi-objective vehicle routing and loading problem with time windows in which three conflicting objectives, total travel distance, number of routes, and load-mixing across pallets, were combined into a single weighted cost function to obtain a solution without requiring preferential information from the decision-maker in advance. The method is simple to implement and guarantees Pareto optimality when all weights are strictly positive, but it requires the decision-maker to specify weights in advance, which can be difficult to interpret meaningfully, and it is known to perform poorly at capturing solutions in non-convex regions of the Pareto front (Miettinen, 1999).

The lexicographic method, formalized by Fishburn (1974), ranks the objectives by importance and optimizes them sequentially: the highest-priority objective is optimized first, and each subsequent objective is optimized subject to the constraint that the optimal value of all higher-priority objectives must be preserved. This approach is conceptually simple and always yields a Pareto-optimal solution, but it can require solving a large number of single-objective problems to obtain a single point on the Pareto front, and it does not naturally produce the full trade-off curve.

Goal programming, introduced by Charnes and Cooper (1961), takes a different approach by allowing the decision-maker to specify an aspiration level, or target value, for each objective, and then minimizing the total deviation from these targets. This reframes the multi-objective problem as a single minimization of weighted or prioritized deviations, and is particularly well suited to situations where target performance levels are known in advance, although the choice of targets directly affects the quality and Pareto-optimality of the resulting solution.

Finally, the ε-constraint method, proposed by Haimes et al. (1971), designates one objective as the primary criterion to be minimized while converting all remaining objectives into constraints bounded by parameters ε. By systematically varying these bounds and re-solving the resulting single-objective problem, the entire Pareto front can be traced out, including solutions in non-convex regions where the weighted sum method fails. This property, together with its straightforward implementation within standard mixed-integer programming solvers, makes the ε-constraint method particularly well suited to problems formulated as Mixed-Integer Linear Programs, and it is the method adopted in this thesis to generate the Pareto front between total transportation time and total transportation cost, as described in Section 4.6.

2.4 Problem Variants and Extensions

This section reviews the main problem variants and extensions that are closest to the problem addressed in this thesis. It covers arc routing problems, in which service is re-

quired along edges rather than at point locations 2.4.1, applications of multi-objective optimization methods to vehicle routing, particularly in the context of environmental and emission-aware objectives 2.4.2, optimization under uncertainty, addressing the robustness of routing solutions to demand variability and other sources of risk 2.4.3, and the incorporation of time windows alongside multiple, often conflicting, service quality criteria 2.4.4.

2.4.1 Arc Routing

Most VRP research treats customers as point locations to be visited. In applications such as street sweeping, waste collection, and road maintenance, however, service is required along entire road segments rather than at individual points, giving rise to arc routing problems. Kraiem et al. (2025) introduced the Multi-Depot Arc Routing Problem with Flexible End Depot (MDARPFED), in which vehicles may conclude their shift at a depot different from the one where they started. The authors developed a MILP formulation for a setting that distinguishes between two arc types: Type H arcs that can only be serviced during night shifts, and Type R arcs that may be serviced in any shift, a distinction arising from real-world street-sweeping contracts that impose scheduling constraints driven by traffic and noise regulations. Solved with IBM ILOG CPLEX on benchmark instances derived from the Pr06 instance of Cordeau, the model reduced total travel time by between 2% and 12% compared to a single-depot configuration, confirming both the practical value of flexible depot assignment and the computational difficulty of the model, which could only be solved to optimality for instances with up to 18 nodes.

2.4.2 Multi-Objective Extensions

Building on the methods introduced in Section 2.3.3, a substantial body of research applies multi-objective optimization techniques to vehicle routing problems, particularly in the context of environmental and emission-related objectives. These Green Vehicle Routing Problem (GVRP) formulations add an environmental objective to the classical cost minimization framework, creating tensions that are addressed using the scalarization techniques discussed previously.

Kabadurmuş et al. (2019) studied a single-depot multi-objective GVRP with Alternative Fuel Vehicles (AFVs) and refueling stations, formulated as a MILP with two objectives: minimizing total CO2 emissions proportional to distance traveled, and minimizing the maximum route duration across all vehicles. The Pareto front was generated using the ε-constraint method, with a notable finding that tighter time limits force the use of more vehicles on shorter routes, which paradoxically increases total emissions, a result with direct implications for fleet management policy.

Wang et al. (2019) extended the GVRP to a multi-depot capacitated setting, formulating a bi-objective MILP that minimizes both total economic cost, comprising facility opening costs, fixed vehicle costs, and distance-dependent fuel expenditure, and total CO2

emissions, monetized via a carbon price. The environmental objective was modeled as a function of both distance and gross vehicle weight, reflecting the well-established relationship between load and fuel consumption. The Pareto front was approximated using a hybrid ACO-VNS metaheuristic, producing a set of Pareto-optimal solutions from which the decision-maker can select the preferred balance between profitability and environmental impact.

Dao and Nguyen Thi (2018) developed a multi-criteria MILP for the Multi-Depot Heterogeneous VRP (MDHVRP) with an explicit environmental dimension. In addition to minimizing total transportation cost, the model simultaneously targets multiple pollutant emissions, including CO and NO2 equivalents, modeled as linear functions of distance and vehicle-specific emission factors. Because the fleet is heterogeneous, the choice of vehicle type directly affects both cost and environmental performance, creating a richer trade-off surface. The Pareto front was populated using a large neighborhood search metaheuristic iterating over weight vectors, and computational experiments on standard benchmarks showed that substantial emission reductions are achievable with only marginal cost increases, an insight of significant practical relevance for fleet procurement and routing decisions.

2.4.3 Optimization under Uncertainty

Real-world distribution operations are subject to demand uncertainty, making robust formulations increasingly important. Xu et al. (2022) formulated a multi-objective MDVRP for refined oil distribution with soft time windows and supply constraints, optimizing three objectives simultaneously: minimizing total operational cost, minimizing penalties for late deliveries, and maximizing the aggregate satisfaction of service stations, with higher weights assigned to priority stations. A minimum safety stock constraint ensured that no station fell below its critical inventory level. A robust extension was developed by introducing demand uncertainty through a robustness control parameter that trades increased cost against protection from demand deviations. The problem was solved using Multi-Objective Particle Swarm Optimization (MOPSO), and results confirmed that the robust formulation meaningfully improves plan resilience.

A related concern for hazardous materials logistics motivates the work of Zhou et al. (2021), who studied the Half-Open Multi-Depot Heterogeneous VRP for Hazardous Material Transportation. In this variant, vehicles depart from depots but may terminate at any permitted facility, reducing deadhead travel and the associated exposure risk along return legs. The model integrates strategic facility location decisions with operational routing across multiple planning periods, allowing simultaneous optimization of depot opening costs, transport costs, vehicle utilization, and penalties for unmet demand. A hybrid optimization approach was used to manage the combined strategic-operational complexity.

2.4.4 Time Windows and Service Quality

The incorporation of time windows, specifying intervals within which customers must be served, adds a further layer of realism to routing models and is particularly relevant in distribution contexts where punctuality is a firm requirement. Li et al. (2025) studied the multi-objective GVRP with time windows (MOGVRPTW), formulating a model with three objectives: minimizing total cost (fixed vehicle costs, fuel expenditure, and penalties for early and late arrivals), minimizing carbon emissions as a function of load-dependent fuel consumption, and maximizing customer satisfaction through a piecewise function that rewards on-time deliveries within a preferred sub-interval. The problem was solved with an improved version of the Non-Dominated Sorting Genetic Algorithm III (NSGA-III), a state-of-the-art evolutionary algorithm for many-objective optimization. The results demonstrated that simultaneously balancing service quality, cost, and environmental impact is achievable through reference-point-based Pareto search.

2.5 Emerging Trends: Machine Learning for Routing

Alongside classical optimization, a growing body of research is exploring end-to-end machine learning approaches that learn to construct good routes directly from problem instances, without the need to formulate and solve a mathematical program explicitly. These methods are particularly attractive for real-time applications where solution speed is paramount, and the field has expanded rapidly enough in recent years to warrant a dedicated survey by Bogyrbayeva et al. (2024), who provide a comprehensive overview of the main learning paradigms applied to vehicle routing, including supervised, reinforcement, and hybrid learning approaches.

An early and influential contribution to this line of work is the model proposed by Nazari et al. (2018), who introduced an end-to-end reinforcement learning framework for the Capacitated VRP. Rather than training a separate policy for each problem instance, the authors propose a single parameterized stochastic policy trained via policy gradient methods, capable of producing near-optimal solutions for a broad range of instances of similar size without retraining. Their approach was shown to outperform classical heuristics and commercial solvers such as Google's OR-Tools on medium-sized instances, while maintaining comparable computation time after training, and it also demonstrated the ability to handle split deliveries.

Building on this foundation, more recent work has explored graph-based representations of the routing problem. Wang et al. (2025) proposed a Constraint-Based Adaptive Hypergraph Learning framework for the CVRP. The approach draws conceptually on a set partitioning formulation, but replaces the solver with a graph neural network that learns to partition customers into routes. The key innovation is the use of hyperedges, connections that link more than two nodes simultaneously, to capture high-order relationships among groups of customers that simple pairwise graph edges cannot represent. Hyperedges are

constructed dynamically during training based on node similarity and are penalized when they induce constraint violations. The model uses a Graph Attention Network (GAT) encoder with multi-head attention and an autoregressive dual-pointer decoder, trained via policy gradient reinforcement learning. On small and medium benchmark instances, the approach outperformed several competing neural combinatorial optimization methods, though sensitivity to the hyperparameters governing hyperedge reconstruction remains a challenge for practical deployment.

Closer to the specific combination of concerns addressed in this thesis, Zhou et al. (2025) proposed a multiobjective edge-based learning algorithm for the VRP with time windows, directly combining the multi-objective and time-window themes discussed throughout this chapter with a learning-based solution approach. Their method learns edge-level representations of the routing graph to simultaneously guide route construction across multiple conflicting objectives, illustrating how machine learning techniques are increasingly being extended beyond single-objective routing to the same kind of multi-criteria settings traditionally addressed by exact and metaheuristic methods.

2.6 Summary and Research Gaps

The literature reviewed above reveals several consistent patterns. First, the MDVRP remains an active and expanding research frontier, with mathematical formulations continuing to be refined (Montoya-Torres et al., 2015; Baldacci et al., 2004; Lalla-Ruiz and Mes, 2021) and new problem variants, including open routes, arc routing, hazardous materials, and flexible depot assignment, steadily broadening the scope of multi-depot optimization. Second, metaheuristics and multi-objective evolutionary algorithms dominate applied research, driven by the computational intractability of exact methods for large instances and by the increasing prevalence of problems involving two or more conflicting objectives. Third, environmental concerns have moved from the periphery to the center of the field, with green routing formulations now constituting a substantial share of recent publications and emission-minimization objectives appearing alongside or even replacing cost minimization (Kabadurmuş et al., 2019; Wang et al., 2019; Dao and Nguyen Thi, 2018; Li et al., 2025).

Despite these advances, certain areas remain underexplored. Real-world case studies with full multi-depot configurations account for fewer than 10% of published studies (Montoya-Torres et al., 2015; Mogale et al., 2025), and integrated models that simultaneously address facility location decisions, demand uncertainty, fleet heterogeneity, and multiple environmental objectives are still rare. Machine learning-based approaches (Wang et al., 2025; Nazari et al., 2018; Zhou et al., 2025) represent an emerging paradigm that may substantially alter the solution landscape for routing problems in the coming years, though considerable work remains before these methods are reliable enough for routine deployment in complex operational settings.

More specifically, very few studies combine, within a single optimization framework, the full set of operational features that characterize real-world container drayage: a multi-depot fleet structure, vehicle heterogeneity, hazardous material handling restrictions, time windows at every node, driver working-time regulations, and the simultaneous management of full and empty container flows. Existing multi-objective formulations typically address a subset of these features in isolation, most commonly pairing multi-depot structure with environmental objectives or with uncertainty, but rarely integrating the operational detail required to capture the day-to-day reality of hinterland truck distribution.

This thesis aims to address this gap by developing a bi-objective Mixed-Integer Linear Programming model for the operational planning of container truck fleets in multi-depot hinterland networks, explicitly incorporating all of the features identified above, and by generating Pareto-optimal solutions between total transportation time and total transportation cost using the ε-constraint method. A formal characterization of the problem addressed is presented in Chapter 3, followed by the full mathematical formulation in Chapter 4.

Chapter 3

Problem Definition

This chapter provides a formal characterization of the container truck routing and scheduling problem addressed in this thesis. It begins by describing the operational context in which the problem arises, covering the key features of the logistics environment that motivate the modeling choices made in Chapter 4 (Section 3.1). It then presents a structured description of the problem itself, including the decisions to be made, the entities involved, and the assumptions adopted to make the problem tractable (Section 3.2). Together, these elements provide the conceptual foundation on which the mathematical formulation of Chapter 4 is built.

3.1 Operational Context

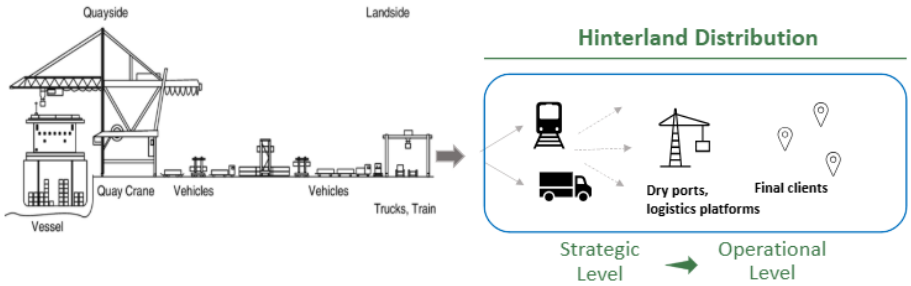


Figure 3.1: Schematic representation of a container terminal and its hinterland distribution chain, from quayside to final clients via dry ports and logistics platforms, adapted from Steenken et al. (2004).

The problem addressed in this thesis is rooted in the capillary distribution leg of port hinterland logistics, represented in figure 3.1, which corresponds to the final stage of the container supply chain. Hinterland logistics can be broadly divided into two components: the first leg, which handles the movement of containers between the port and a set of inland terminals by rail or road; and the second leg, known as capillary distribution, which covers

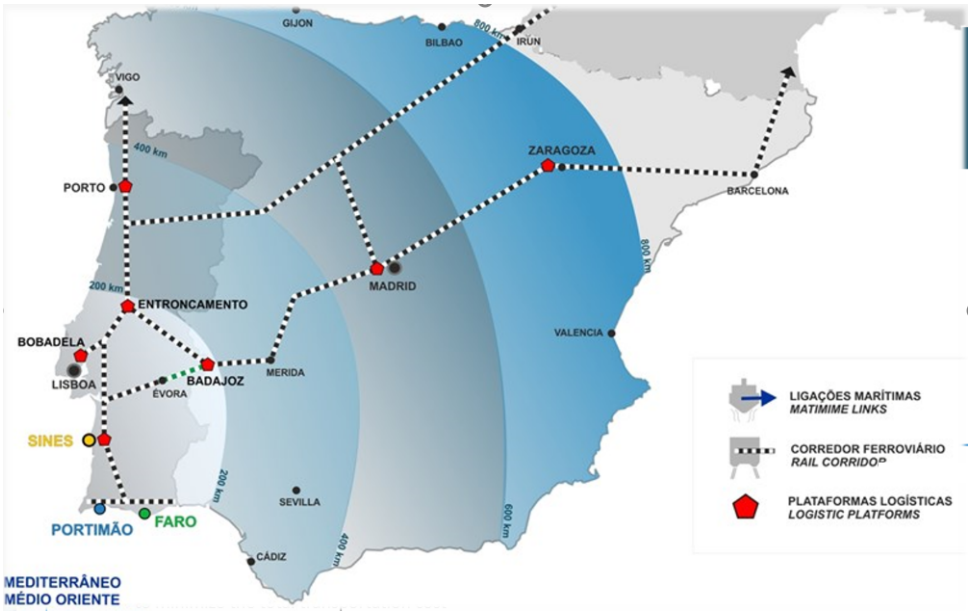


Figure 3.2: Hinterland distribution network of the Port of Sines, showing maritime links, rail corridors, and logistics platforms across the Iberian Peninsula (APS – Ports of Sines and the Algarve Authority, 2026).

the pickup and delivery of containers between those inland terminals and the final clients, performed exclusively by truck fleets. This thesis focuses on the direct distribution of containers from ports to final clients, and is motivated by the operational context of the Port of Sines hinterland network (APS – Ports of Sines and the Algarve Authority, 2026), which extends across Portugal through inland terminals at Bobadela, Entroncamento, and Porto, as shown in figure 3.2. Each day, a fleet of trucks must be routed from their depots to the port to collect containers, deliver them to clients scattered across the territory, manage the return of empty containers back to the port, and return to their depots by the end of the working day. The operational planning problem therefore consists of assigning transportation tasks to vehicles and determining the sequence and timing of their movements while accounting for the operational characteristics and constraints that define the container drayage system. These elements are described in the following sections and form the basis of the mathematical formulation presented in Chapter 4.

Central to this problem is the shipping container itself. Containers are standardized steel boxes designed for intermodal transport, meaning they can be transferred seamlessly between ships, trains, and trucks without unpacking their contents. Standardized under ISO 668 (International Organization for Standardization, 2020), two sizes dominate global trade and are relevant to this thesis: the 20-foot container, known in the industry as a Twenty-foot Equivalent Unit (TEU), measuring approximately 6.1 m in length; and the 40-foot container, known as a Forty-foot Equivalent Unit (FEU), measuring approximately 12.2 m. Both share the same width (2.44 m) and standard height (2.59 m), making them stackable and interchangeable across terminal infrastructure. The 20-foot container is

typically used for heavier, denser cargo, while the 40-foot container is preferred for lighter, bulkier goods. These two sizes translate directly into the capacity units used in the model developed in Chapter 4: a 20-foot container occupies 1 unit of vehicle capacity and a 40-foot container occupies 2 units. Figure 3.3 illustrates the two container sizes and their relative dimensions.

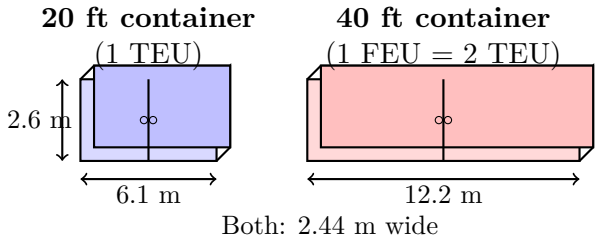


Figure 3.3: Standard ISO shipping container sizes: 20-foot (TEU) and 40-foot (FEU). Source: based on ISO 668:2020 (International Organization for Standardization, 2020).

Table 3.1 summarizes the key operational features of the problem addressed in this thesis. Each feature is described in detail in the following subsections.

Table 3.1: Problem-specific features of the container truck routing problem.

Feature	**Description**
Multi-depot structure	Vehicles are distributed across multiple depots located near the coastline. Each vehicle starts and ends its route at its assigned depot.
Heterogeneous fleet	Two vehicle types are considered: conventional diesel trucks (100 km/h) and battery-electric trucks (70 km/h), differing in speed and operating cost.
Full and empty containers	Vehicles manage both full containers (collected at ports, delivered to clients) and empty containers (returned to ports after client delivery).
Hazardous materials	Some tasks require ADR-certified vehicles. Only certified vehicles may be assigned to hazardous tasks.
Time windows	Every node has an associated time window within which service must begin, reflecting terminal operating hours and client delivery commitments.
Driver working time	Each driver is limited to a maximum of 480 minutes per working day, bounding the total elapsed time from depot departure to return.

3.1.1 Multi-Depot Structure

The truck fleet does not operate from a single central facility. Instead, vehicles are distributed across multiple depots, each located near the coastline, reflecting the proximity of fleet operators to the port and inland terminal infrastructure. In the instances generated

for this thesis, depots are placed within 20 kilometers of the western coast, at varying latitudes across the Portuguese territory. Each vehicle begins and ends its working day at its assigned depot, and the model allows different vehicles to start and end at different depots, capturing the distributed nature of real hinterland fleet operations.

3.1.2 Heterogeneous Fleet

The vehicle fleet is heterogeneous in terms of propulsion technology. Two vehicle types are considered: conventional diesel-powered trucks and battery-electric trucks. These two types differ in their operating cost per kilometer and in their travel speed: conventional trucks are assumed to travel at 100 km/h, while electric trucks travel at 70 km/h, reflecting the more conservative operating speeds and additional time overhead associated with current electric heavy vehicle technology (Öko-Institut, 2024). This speed difference creates a direct tension between the two objectives of the model, since electric vehicles are cheaper to operate but slower, while conventional vehicles are faster but more costly. Both vehicle types share the same load capacity of $Q = 2$ container units and the same maximum working time.

3.1.3 Full and Empty Containers

Each transportation task involves moving a container between a port and a client location. Containers may be either full, carrying goods, or empty. At the start of each route, all vehicles depart their depots without any containers on board. Full containers are collected at port nodes and delivered to clients. Upon delivery, some clients retain the empty container after unloading the goods, while others return it to the vehicle. In the latter case, the vehicle accumulates an empty container that must be returned to a port before the end of the route. Port nodes are therefore dual-purpose: they serve as collection points for full containers and as deposit points for empty containers.

3.1.4 Hazardous Materials

Some transportation tasks involve containers carrying hazardous materials, such as chemical products or flammable goods. In practice, the transportation of such cargo requires the driver to hold a specific certification, in the European Union, this corresponds to the ADR certificate (European Parliament and Council of the European Union, 2008), which qualifies drivers to handle and transport dangerous goods by road. Since in this model each vehicle is operated by a single dedicated driver throughout the planning horizon, the vehicle and its driver are treated as a single entity, and the certification is therefore attributed to the vehicle rather than to the driver separately. A task requiring hazardous material handling can only be assigned to a certified vehicle. This constraint adds a combinatorial layer to the assignment problem, since the set of vehicles eligible for certain tasks is restricted.

3.1.5 Time Windows

Every node in the network is associated with a time window specifying the earliest and latest times at which a vehicle may begin service there. Time windows at client nodes reflect delivery commitments and scheduling agreements with the final recipients of the containers, while time windows at port nodes reflect terminal operating hours. Depot and port nodes are assumed to be available throughout the entire planning horizon, imposing no temporal restriction on vehicle departure or return. The presence of time windows at every node adds a scheduling dimension to the routing problem that goes beyond simple distance or cost minimization, since the sequence in which tasks are executed must be compatible with the temporal constraints at every location along the route.

3.1.6 Driver Working Time Regulations

Each vehicle is operated by a single driver, and the total working time of each driver is bounded by a maximum of 480 minutes per day, corresponding to a standard eight-hour working shift. This constraint limits the total elapsed time from when a vehicle departs its depot to when it returns, including all travel, service, and waiting time along the route. The working time limit is a binding constraint in practice, particularly for vehicles assigned to tasks spread across long distances, and it influences both the number of vehicles needed and the structure of the routes produced by the model.

3.2 Problem Description and Assumptions

The problem involves a set of trucks that must perform a collection of transportation tasks, each consisting of picking up a full container from a port and delivering it to a final client. Trucks begin and end their working day at one of several depots. The model determines, for each vehicle, the depot it departs from and returns to, the sequence of port visits and client deliveries it performs, and the timing of each operation. A vehicle is not restricted to using the same port throughout its route: for example, a vehicle may collect containers at one port, deliver them to clients, and then visit a different port to collect a second load before continuing its route. Any empty containers accumulated during the route must be returned to a port before the vehicle returns to its depot, so that depots are always entered and exited without any containers on board.

The model is subject to a number of simplifying assumptions that are standard in the vehicle routing literature and that make the problem tractable for the instance sizes considered in this thesis. First, all parameters are assumed to be deterministic: travel times, service durations, time windows, and demands are all known with certainty at the start of the planning horizon. Second, the set of transportation tasks to be performed is fully known at the beginning of the day, meaning no new tasks arrive dynamically during execution. Third, travel times between any pair of nodes are fixed and do not vary

with time of day, traffic conditions, or vehicle load, meaning there is no congestion effect. Fourth, the planning horizon corresponds to a single working day, and vehicles are not carried over between planning cycles. Fifth, each vehicle is operated by a single dedicated driver throughout the day, so vehicle and driver are treated as a single entity with a unified working time limit.

With these assumptions, the model simultaneously optimizes two competing performance criteria: total transportation time, measured as the sum of travel times across all arcs used by all vehicles, and total transportation cost, measured as the total fuel and energy expenditure across the fleet. The multi-objective nature of the problem means that no single solution minimizes both criteria simultaneously, and the model is therefore solved using the ε-constraint method to generate the Pareto front between the two objectives, as described in Chapter 4. Figure 3.4 is a generalized representation of the problem.

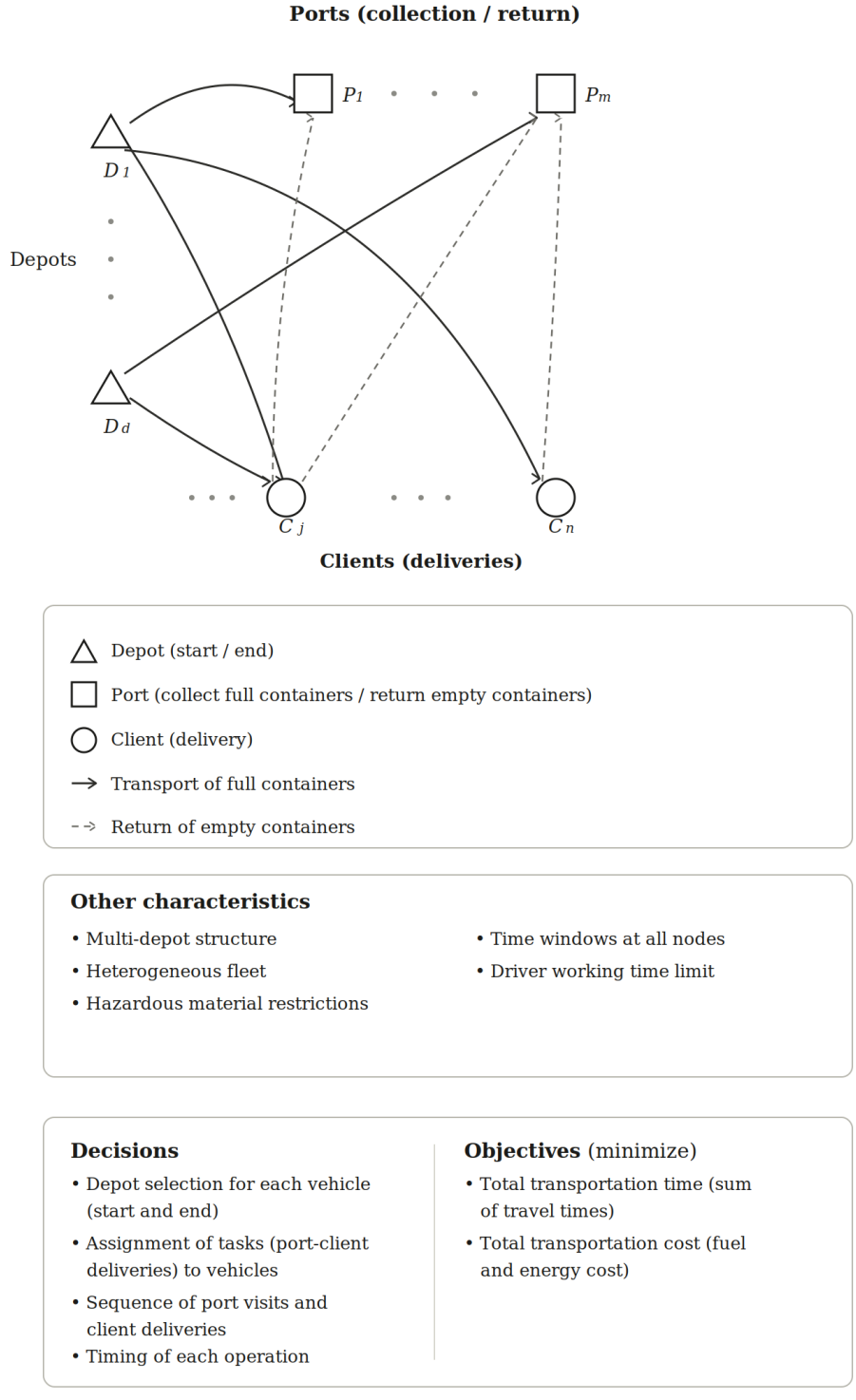


Figure 3.4: Port logistics network: depots, ports, and clients with container transport flows.

Chapter 4

Mathematical Model

This chapter presents the mathematical formulation developed to model the container truck routing and scheduling problem described in Chapter 3. The model belongs to the family of Multi-Depot Vehicle Routing Problems with Time Windows and Pickup and Delivery constraints, formulated as a Mixed-Integer Linear Program. The following sections introduce the network nodes 4.1, parameters 4.2, and decision variables 4.3 that define the model, followed by the objective functions 4.4 and constraints 4.5, each accompanied by an explanation of their role and motivation. The chapter closes with a discussion of the multi-objective formulation and the solution strategy adopted to generate the Pareto front between the two competing objectives 4.6.

4.1 Network Nodes

The model is defined over a set of nodes representing all relevant locations in the network, and a set of vehicles that perform the transportation tasks. The nodes are partitioned into three distinct subsets, each playing a different role in the model.

- C — set of delivery locations, i.e., client nodes where containers are picked up or delivered.
- D — set of depots, where vehicles begin and end their routes.
- P — set of ports, which serve a dual purpose: vehicles may visit a port to pick up full containers, to deposit empty containers accumulated during the route, or to drop off any unused full containers before returning to the depot. This dual-purpose design means that a single port visit can simultaneously handle loading and unloading operations, making the model more flexible and realistic.
- $N = C \cup D \cup P$ — the complete set of all nodes in the network.
- K — set of vehicles available for routing.

- $A = \{(i,j) \in N \times N : i \neq j\}$ — set of directed arcs connecting all pairs of distinct nodes.

4.2 Parameters

The parameters capture all the fixed input data of the problem, including travel times, vehicle characteristics, time window bounds, demand values, and cost coefficients.

- t_{ijk}, $i,j \in N$, $k \in K$ — travel time from node i to node j by vehicle k.
- $e_k \in \{0,1\}$, $k \in K$ — equals 1 if vehicle k is electric.
- a_i, $i \in N$ — earliest service time at node i.
- b_i, $i \in N$ — latest service time at node i.
- p_i, $i \in N$ — processing time at node i.
- d_i, $i \in C$ — demand at task i.
- $\delta_i \in \{0,1\}$, $i \in C$ — return indicator; equals 1 if the container is returned after service at node i.
- $h_i \in \{0,1\}$, $i \in C$ — equals 1 if task i requires hazardous material handling capability.
- $\gamma_k \in \{0,1\}$, $k \in K$ — equals 1 if vehicle k is certified to handle hazardous materials.
- Q — vehicle capacity, measured in container units. Since a 40-foot container occupies 2 units and a 20-foot container occupies 1 unit, the capacity is fixed at $Q = 2$ for all vehicles.
- c^0, c^1, c^2 — fuel consumption rates for non-electric vehicles traveling with 0, 1, and 2 units of full load, respectively. Fuel consumption rates c^0, c^1, and c^2 are derived from empirical data published by the U.S. Bureau of Transportation Statistics (BTS), which reports commercial truck fuel consumption as a function of average operating weight (U.S. Bureau of Transportation Statistics, 2024). Three representative weight classes are defined, corresponding to the three load states of the model. An empty truck is assumed to weigh approximately 35,000 lbs (15,876 kg), reflecting a typical tractor-chassis combination with no container on board. A truck carrying a single 20-foot container is assumed to weigh approximately 61,000 lbs (27,669 kg), combining the tractor-chassis weight with the container tare weight of roughly 5,070 lbs (International Organization for Standardization, 2020) and a representative payload. A truck carrying a single 40-foot container is assumed to weigh approximately 88,000 lbs (39,916 kg), consistent with a loaded 40-foot container whose tare weight is approximately 8,160 lbs (International Organization for Standardization, 2020)

and a typical gross vehicle weight approaching the federal limit. Reading the corresponding fuel consumption values from the BTS curve for these three weight classes and converting from gallons per 100 miles to liters per kilometer yields $c^0 = 0.311$, $c^1 = 0.362$, and $c^2 = 0.401$ liters per kilometer, respectively. These values reflect the well-established relationship between vehicle weight and fuel consumption, and are consistent with figures reported in the broader freight transport literature (International Road Union, 2024).

- c^{fuel} — cost of fuel per kilometer for non-electric vehicles, set to $c^{fuel} = 2$ monetary units per kilometer. This figure reflects the full variable operating cost of a diesel heavy-duty truck in a European hinterland context, encompassing fuel expenditure, which at a European average diesel price of approximately €1.57 per liter International Road Union (2024) and a consumption rate of around 0.38 liters per kilometer yields a fuel cost of roughly €0.60 per kilometer, together with tolls, which on major European corridors range from €0.15 to €0.26 per kilometer for heavy vehicles International Road Union (2024), maintenance, insurance, and other distance-dependent costs, bringing the total variable cost to approximately €1.50 to €2.00 per kilometer, consistent with figures reported in the freight transport literature (International Road Union, 2024).

- c^{elec} — cost per kilometer for electric vehicles, set to $c^{elec} = 0.2$ monetary units per kilometer. This is derived from a representative energy consumption of approximately 1.1 kWh per kilometer for a heavy-duty battery-electric truck (Basma et al., 2021), combined with a depot charging electricity price of approximately €0.14 to €0.20 per kWh reported for European operators in 2024 (International Road Union, 2024), yielding an energy cost of roughly €0.15 to €0.22 per kilometer. The ratio between c^{fuel} and c^{elec} reflects the well-documented cost advantage of electric propulsion on a per-kilometer basis, which is one of the key drivers of the trade-off between the two objective functions explored in this thesis.

- α — speed conversion factor, set to $\alpha = 100/70$, reflecting a speed of 100 km/h for non-electric vehicles with time expressed in minutes.

- T^{max} — maximum permitted working time per vehicle, in minutes.

- F^{max} — maximum permitted total fuel and energy consumption cost across all vehicles.

- M — a sufficiently large constant used in big-M constraints to conditionally activate or deactivate inequalities.

4.3 Decision Variables

The decision variables capture both the routing choices made for each vehicle and the temporal and load state information along each route. They can be grouped into three categories: binary routing variables, continuous time variables, and load tracking variables.

- $x_{ijk} \in \{0,1\},\ i,j \in N,\ k \in K$ — equals 1 if vehicle k travels from node i to node j.
- $u_k \in \{0,1\},\ k \in K$ — equals 1 if vehicle k is used in the solution.
- $y_{dk} \in \{0,1\},\ d \in D,\ k \in K$ — equals 1 if vehicle k starts its route at depot d.
- $z_{dk} \in \{0,1\},\ d \in D,\ k \in K$ — equals 1 if vehicle k ends its route at depot d.
- $arr_{ijk} \geq 0,\ i,j \in N,\ k \in K$ — arrival time of vehicle k at node j when traveling via arc (i,j). Arrival times are indexed to arcs rather than nodes, which allows constraint (4.8) to determine directly, from the arc-indexed variables x_{ijk} and x_{jlk}, whether two arcs are used in sequence by the same vehicle, without requiring an auxiliary visit indicator. The variable is only meaningful when $x_{ijk} = 1$.
- $s_k \geq 0,\ k \in K$ — start time of vehicle k.
- $f_k \geq 0,\ k \in K$ — finish time of vehicle k.
- $q^F_{ijk} \geq 0,\ i,j \in N,\ k \in K$ — full container load carried by vehicle k on arc (i,j).
- $q^E_{ijk} \geq 0,\ i,j \in N,\ k \in K$ — empty container load carried by vehicle k on arc (i,j).
- $z^0_{ijk}, z^1_{ijk}, z^2_{ijk} \in \{0,1\},\ i,j \in N,\ k \in K$ — equal to 1 if arc (i,j) is traveled by vehicle k with 0, 1, or 2 units of full load, respectively.

4.4 Objective Functions

Two objective functions are considered, each capturing a different dimension of operational performance that matters to a logistics operator.

The first objective minimizes the total transportation time, measured as the sum of travel times across all arcs used by all vehicles:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} t_{ijk} x_{ijk} \tag{4.1}$$

The second objective minimizes the total energy and fuel consumption cost. For non-electric vehicles, this cost depends on the load carried on each arc, since heavier loads consume more fuel. This is captured through the load state variables z^0, z^1, and z^2, which indicate whether an arc is traveled with zero, one, or two units of full load, each

associated with a different consumption rate. For electric vehicles, the cost depends only on the distance traveled, independent of the load carried:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} \left[(1 - e_k) c^{fuel} \alpha t_{ijk} \left(c^0 z^0_{ijk} + c^1 z^1_{ijk} + c^2 z^2_{ijk} \right) + e_k c^{elec} t_{ijk} x_{ijk} \right] \tag{4.2}$$

These two objectives are the source of the central trade-off explored in this thesis. Electric vehicles have lower consumption costs per kilometer, which makes them attractive from a cost minimization perspective. However, they are operationally slower than conventional vehicles: in this model, conventional trucks are assumed to travel at 100 km/h while electric trucks travel at 70 km/h, reflecting the additional time overhead associated with charging stops and the more conservative operating speeds typical of current electric heavy vehicles. This speed difference means that solutions favoring electric vehicles tend to perform worse in terms of total transportation time. Conversely, conventional vehicles are faster and more operationally flexible, making them the preferred choice when time is the primary concern, but at a higher fuel cost. This tension between time and cost is what makes the multi-objective formulation both necessary and meaningful, and it is explored in detail through the Pareto front generated in Chapter 5.

A natural question that arises from this formulation is the degree of independence between the two objectives, that is, whether they are truly conflicting across the full feasible region or only partially so. Analyzing this formally falls outside the scope of this work but is identified as a direction for future research in Section 6.4.

4.5 Constraints

The constraints are organized into five groups: routing and flow constraints 4.5.1, time constraints 4.5.2, hazardous material and capacity constraints 4.5.3, container load flow constraints 4.5.4, and load state variable constraints 4.5.5.

4.5.1 Routing and Flow Conservation

These constraints ensure that every customer node is visited exactly once, that vehicle routes are continuous, and that each vehicle begins and ends at exactly one depot.

Each customer node must be visited exactly once across all vehicles:

$$\sum_{k \in K} \sum_{j:(j,i) \in A} x_{jik} = 1 \qquad \forall i \in C \tag{4.3}$$

Flow conservation is enforced at all customer and port nodes, ensuring that a vehicle that arrives at a node also departs from it:

$$\sum_{j:(i,j)\in A} x_{ijk} = \sum_{j:(j,i)\in A} x_{jik} \qquad \forall i \in C \cup P, \forall k \in K \tag{4.4}$$

Each vehicle that is used must start at exactly one depot:

$$\sum_{d\in D} y_{dk} = u_k \qquad \forall k \in K \tag{4.5}$$

Each vehicle that is used must end at exactly one depot:

$$\sum_{d\in D} z_{dk} = u_k \qquad \forall k \in K \tag{4.6}$$

Flow balance at depot nodes ensures consistency between the starting and ending depot assignments and the arcs traveled:

$$\sum_{j:(d,j)\in A} x_{djk} - \sum_{j:(j,d)\in A} x_{jdk} = y_{dk} - z_{dk} \qquad \forall d \in D, \forall k \in K \tag{4.7}$$

4.5.2 Time Constraints

These constraints enforce the correct propagation of time along routes, define the start and finish times of each vehicle, impose time windows at service nodes, and limit total working time.

Since arrival times are indexed to arcs, time propagation is enforced over pairs of consecutive arcs sharing a common intermediate node. If vehicle k travels arc (i,j) and subsequently arc (j,l), the arrival time at j via (i,j) plus the service time at j plus the travel time from j to l must not exceed the arrival time at l via (j,l). The double big-M term deactivates the constraint whenever either arc is unused:

$$arr_{ijk} + p_j + t_{jlk} \leq arr_{jlk} + M(1 - x_{ijk}) + M(1 - x_{jlk}) \qquad \forall (i,j),(j,l) \in A, \forall k \in K \tag{4.8}$$

Time windows are imposed at all visited nodes. Since arrival times are indexed to arcs, the constraint is deactivated simply by $x_{ijk} = 0$, without requiring a sum of visit indicators. This formulation is therefore valid for client nodes, depot nodes, and port nodes alike, without requiring a sum of visit indicators.

$$a_j - M(1 - x_{ijk}) \leq arr_{ijk} \leq b_j - p_j + M(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in K \tag{4.9}$$

The start time of each vehicle is defined as its arrival time at its origin depot d before departing along the first arc of its route. Since each vehicle departs from exactly one depot by constraint (4.5), exactly one of these constraints will be active per vehicle:

$$s_k \leq arr_{djk} + M(1 - x_{djk}) \qquad \forall d \in D, \forall (d,j) \in A, \forall k \in K \tag{4.10}$$

$$s_k \geq arr_{djk} - M(1 - x_{djk}) \qquad \forall d \in D, \forall (d,j) \in A, \forall k \in K \tag{4.11}$$

The finish time of each vehicle is defined as its arrival time at its ending depot d via the last arc of its route, plus the depot processing time p_d. Since each vehicle ends at exactly one depot by constraint (4.6), exactly one of these constraints will be active per vehicle:

$$f_k \leq arr_{idk} + p_d + M(1 - x_{idk}) \qquad \forall d \in D, \forall (i,d) \in A, \forall k \in K \tag{4.12}$$

$$f_k \geq arr_{idk} + p_d - M(1 - x_{idk}) \qquad \forall d \in D, \forall (i,d) \in A, \forall k \in K \tag{4.13}$$

The total working time of each vehicle must not exceed the permitted maximum. Note that $f_k \geq s_k$ is implied by constraint (4.8) propagating time forward along used arcs combined with $p_d \geq 0$, so no explicit constraint is needed:

$$f_k - s_k \leq T^{max} u_k \qquad \forall k \in K \tag{4.14}$$

4.5.3 Hazardous Material and Capacity Constraints

If a task requires hazardous material handling, only a capable vehicle may be assigned to it:

$$\sum_{j:(j,i) \in A} x_{jik} \leq \gamma_k + (1 - h_i) \qquad \forall i \in C, \forall k \in K \tag{4.15}$$

The total load carried by vehicle k on any arc must not exceed the vehicle capacity. When $x_{ijk} = 0$ the right-hand side is zero, forcing both loads to zero on unused arcs. When $x_{ijk} = 1$ it imposes the standard capacity limit Q. Since loads are naturally bounded by Q, this uses Q as a tight bound rather than a large M:

$$q^F_{ijk} + q^E_{ijk} \leq Q x_{ijk} \qquad \forall (i,j) \in A, \forall k \in K \tag{4.16}$$

4.5.4 Container Load Flow Constraints

These constraints model the evolution of full and empty container loads along each vehicle's route. They ensure that vehicles depart and arrive at depots carrying no containers, that full and empty container flows are correctly conserved at customer nodes, and that port visits reset the empty load to zero.

Vehicles depart every depot carrying no full containers:

$$q^F_{djk} = 0 \qquad \forall d \in D, \forall (d,j) \in A, \forall k \in K \tag{4.17}$$

Vehicles depart every depot carrying no empty containers:

$$q^E_{djk} = 0 \qquad \forall d \in D, \forall (d,j) \in A, \forall k \in K \tag{4.18}$$

Vehicles arrive at every depot carrying no full containers:

$$q^F_{idk} = 0 \qquad \forall d \in D, \forall (i,d) \in A, \forall k \in K \tag{4.19}$$

Vehicles arrive at every depot carrying no empty containers:

$$q^E_{idk} = 0 \qquad \forall d \in D, \forall (i,d) \in A, \forall k \in K \tag{4.20}$$

Flow conservation for full containers at customer nodes: the net consumption of full containers at node i equals the delivery demand d_i, multiplied by the visit indicator to deactivate the constraint for vehicles that do not visit node i:

$$\sum_{j:(j,i)\in A} q^F_{jik} - \sum_{j:(i,j)\in A} q^F_{ijk} = d_i \sum_{j:(j,i)\in A} x_{jik} \qquad \forall i \in C, \forall k \in K \tag{4.21}$$

Flow conservation for empty containers at customer nodes: the net generation of empty containers at node i equals $d_i\delta_i$, the number of containers returned after delivery, multiplied by the visit indicator:

$$\sum_{j:(i,j)\in A} q^E_{ijk} - \sum_{j:(j,i)\in A} q^E_{jik} = d_i\delta_i \sum_{j:(j,i)\in A} x_{jik} \qquad \forall i \in C, \forall k \in K \tag{4.22}$$

Every arc departing a port carries zero empty containers. Full containers on port departure arcs are not fixed by this constraint: the solver decides how many to load, determined implicitly by the flow conservation constraints and the capacity constraint (4.16). This makes ports dual-purpose: a vehicle visiting a port can pick up full containers, deposit empty containers accumulated during the route, and deposit any unused full containers it cannot deliver, all in a single visit:

$$q^E_{pjk} = 0 \qquad \forall p \in P, \forall (p,j) \in A, \forall k \in K \tag{4.23}$$

4.5.5 Load State Variable Constraints

These constraints link the binary load state variables z^0, z^1, z^2 to the actual full load on each arc, enabling the fuel consumption calculation. Unlike the previous version of the model, since load variables are now defined on arcs, these constraints apply uniformly across all arcs with no special cases for port or depot nodes.

Exactly one load state must be active per traveled arc:

$$z^0_{ijk} + z^1_{ijk} + z^2_{ijk} = x_{ijk} \qquad \forall (i,j) \in A, \forall k \in K \tag{4.24}$$

If the arc is traveled with zero full load, q^F_{ijk} must equal zero:

$$q^F_{ijk} \leq Q(1 - z^0_{ijk}) \qquad \forall (i,j) \in A, \forall k \in K \tag{4.25}$$

If the arc is traveled with one unit of full load, q^F_{ijk} must equal one:

$$q^F_{ijk} \geq z^1_{ijk} \qquad \forall (i,j) \in A, \forall k \in K \tag{4.26}$$

$$q^F_{ijk} \leq 1 + Q(1 - z^1_{ijk}) \qquad \forall (i,j) \in A, \forall k \in K \tag{4.27}$$

If the arc is traveled with two units of full load, q^F_{ijk} must equal two:

$$q^F_{ijk} \geq 2z^2_{ijk} \qquad \forall (i,j) \in A, \forall k \in K \tag{4.28}$$

4.6 Multi-Objective Solution Strategy

The two objective functions defined in Section 4.4 capture fundamentally different aspects of operational performance. Minimizing total travel time pushes the model toward faster, more direct routes, which tend to consume more fuel and incur higher costs. Minimizing total cost, on the other hand, encourages more economical routing, which may result in longer travel times. These objectives pull in opposite directions, and no single solution can simultaneously optimize both.

To address this, the problem is formulated as a bi-objective optimization problem and solved using the ε-constraint method. This approach fixes one objective as the primary criterion to be minimized, while the other is converted into a constraint bounded by a parameter ε. By varying ε systematically across its feasible range, a set of Pareto-optimal solutions is generated, each representing a different balance between time and cost. Together, these solutions form the Pareto front, which gives a decision-maker a complete picture of the available trade-offs rather than a single answer. The implementation details of this procedure are described in Chapter 5.

This type of approach positions the model as a true decision support tool rather than a prescriptive optimization engine. Rather than imposing a single solution, the ε-constraint method produces a structured set of alternatives spanning the full range of available trade-offs, allowing the decision-maker to evaluate and compare options in a discretionary way. The choice of which solution to implement ultimately rests with the planner, who can weigh operational priorities, such as fleet availability, client urgency, or sustainability targets, against the time and cost implications made explicit by the Pareto front. This is particularly valuable in a hinterland logistics context, where operational conditions and stakeholder priorities may shift from day to day, and where no single fixed weighting of objectives can capture the full complexity of real planning decisions.

Chapter 5

Computational Experiments and Results

This chapter describes the computational experiments conducted to evaluate the mathematical model presented in Chapter 4, and presents and discusses the results obtained. The chapter is organized as follows. Section 5.1 describes the computational setup and the tools used to implement and solve the model. Section 5.2 details the instance generation procedure. Section 5.3 explains the implementation of the ε-constraint method used to generate the Pareto front. Section 5.4 discusses the scalability experiments conducted to assess how solution time and quality evolve as the number of clients increases, using additional independent instances at selected instance sizes to assess the variability of these results. Section 5.5 presents a sensitivity analysis examining how specific operational factors, namely time window tightness, hazardous material requirements, and fleet composition, affect the computational difficulty of the problem. Finally, Section 5.6 presents and discusses the results.

5.1 Computational Setup

The mathematical model was implemented in Python 3.11.14, distributed via Anaconda, using the Gurobi Optimizer version 12.0.3 as the underlying solver. Gurobi is a state-of-the-art commercial solver for mixed-integer linear programming problems, widely used in both academic research and industrial applications, and was selected here for its performance on combinatorial optimization problems of the type addressed in this thesis. All experiments were run on a machine equipped with an 11th Generation Intel Core i5-1135G7 processor running at 2.40 GHz, 16 GB of RAM, under a 64-bit Windows 11 operating system. A time limit of 3600 seconds was imposed on each solver call to prevent excessively long runs on larger instances. This setup, including the solver configuration and time limit, was used consistently across all computational experiments described in this chapter.

5.2 Instance Generation

Since no real operational dataset was available for the Port of Sines hinterland network at the level of detail required by the model, all instances were generated synthetically. The generation procedure was designed to reflect the structural characteristics of a realistic hinterland distribution network, following an approach inspired by the NEXUS project network described in Chapter 3.

The Portuguese territory is approximated by a rectangle with dimensions 210 km (width, West-East) by 550 km (height, South-North). All node locations are expressed as coordinates (x, y) within this rectangle, where $x \in [0, 210]$ represents the West-East position and $y \in [0, 550]$ represents the South-North position, as shown in figure 5.1.

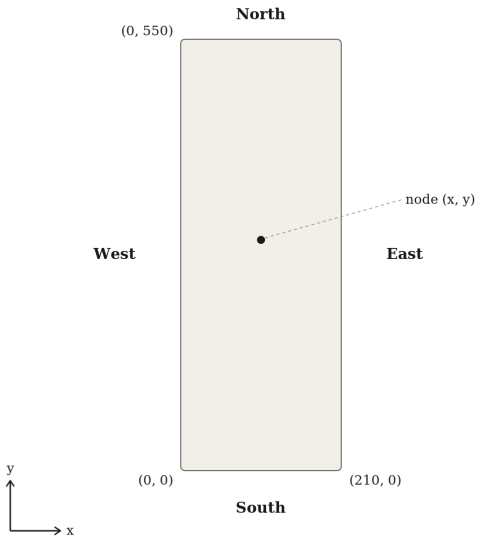


Figure 5.1: Bounding rectangle approximating the Portuguese territory, with coordinates $(x, y) \in [0, 210] \times [0, 550]$.

The network consists of three types of fixed infrastructure nodes and one type of randomly generated node. The distinction between fixed and randomly generated refers to whether node locations are held constant across all instances: fixed nodes, ports and depots, are generated once with structural constraints reflecting the real operational context (ports on the coastline, depots nearby) and then kept identical across every instance and sensitivity variant. Randomly generated nodes represent client locations, i.e., the warehouses, factories, or distribution centers to which containers must be delivered. Since no real client dataset was available, their locations are drawn independently for each instance from a uniform distribution over the full territory rectangle, approximating the geographic spread of real hinterland clients. These are not auxiliary modeling nodes, they are the

actual demand points of the problem, each corresponding to a transportation task that must be assigned to a vehicle and scheduled within the planning horizon.

- **Port nodes** P: two ports are included in all instances, both placed at $x = 0$, reflecting the fact that container terminals are located on the coastline. Their y coordinates were generated randomly and are fixed across all instances.

- **Depot nodes** D: four depots are included in all instances. Their x coordinates are drawn randomly from the interval $[0, 20]$, placing them close to the coast and therefore close to the ports, which reflects the operational reality of hinterland logistics where vehicle fleets are typically stationed near the terminal facilities. Their y coordinates are drawn randomly from a uniform distribution over $[0, 550]$, distributing depots across the full length of the territory. The depot locations are fixed and remain the same across all instances, including all additional instances and sensitivity variants described below.

- **Client nodes** C: client locations are generated randomly within the full rectangle, with both x and y coordinates drawn from a uniform distribution over $[0, 210]$ and $[0, 550]$ respectively. For the main sequence of instances used to assess scalability, referred to hereafter as the *cumulative sequence*, the client nodes are built cumulatively: the 5-node instance defines a base set of clients, and each subsequent instance size is obtained by adding new randomly generated nodes to the existing set. For example, the 8-node instance retains all 5 clients from the previous instance and adds 3 new ones, the 10-node instance retains all 8 and adds 2 more, and so on. This cumulative approach ensures that results across different instance sizes are directly comparable, since larger instances are strict extensions of smaller ones rather than entirely different problems. The instance sizes tested in this cumulative sequence are $|C| \in \{5, 8, 10, 12, 15, 20, 25\}$.

The exact coordinates of all ports and depots, fixed across every instance and sensitivity variant used in this thesis, are listed in Table 5.1. Ports have a processing time of 15 minutes, representing the time required to load or unload containers during a visit. Depots have a processing time of zero, since vehicles only depart from and return to their assigned depot at the start and end of the working day, with no loading or unloading operations taking place there.

In addition to the cumulative sequence, two further independent instances were generated at each of the sizes $|C| \in \{5, 10, 15\}$, so that three instances are available at each of these sizes: the cumulative instance itself, plus two additional independent instances. This allows the variability of computational results to be assessed across different random realizations of the same instance size. Unlike the cumulative instance, each additional instance is generated independently, with fresh random draws for client coordinates, demands, time

Table 5.1: Fixed coordinates and processing times for ports and depots.

Node type	**Node ID**	**Coordinates** (x, y)	**Processing time (min)**
Port	P_1	$(0, 413)$	15
Port	P_2	$(0, 148)$	15
Depot	D_1	$(6, 203)$	0
Depot	D_2	$(16, 12)$	0
Depot	D_3	$(19, 396)$	0
Depot	D_4	$(9, 460)$	0

windows, and hazardous material requirements, while depot and port locations remain fixed as described above.

The full coordinate, time window, demand, and hazardous material data for every client node across the cumulative sequence and the additional independent instances are provided in Appendix A.

Travel times t_{ijk} are computed from the Euclidean distance between each pair of nodes, scaled by the speed of the vehicle type. Non-electric vehicles travel at 100 km/h and electric vehicles at 70 km/h, as discussed in Chapter 4. Since the model works in minutes, the appropriate conversion factor is applied.

For each instance, the remaining parameters are generated as follows:

- **Time windows** $[a_i, b_i]$ are defined within a planning horizon of 480 minutes, corresponding to a standard 8-hour working day. For each client node, the earliest service time a_i is drawn randomly from the interval $[0, 125]$ and the latest service time b_i is drawn randomly from the interval $[350, 480]$. This design produces relatively wide time windows, giving vehicles considerable scheduling flexibility, which is intentional at this stage in order to focus the analysis on the routing and cost trade-offs rather than on feasibility issues arising from tight temporal constraints. Terminal and depot nodes are assumed to be available throughout the entire planning horizon, so their time windows span the full 480 minutes.

- **Demands** d_i are randomly assigned a value of either 1 or 2 for each client node, corresponding to a 20-foot or a 40-foot container respectively, with equal probability.

- **Hazardous material requirements** h_i are assigned randomly to each client node, with a 25% probability of a task being hazardous and a 75% probability of it being non-hazardous. Of the 15 vehicles, 11 are certified to handle hazardous materials: 4 of the 5 electric vehicles, and 7 of the 10 conventional vehicles, meaning that constraint (4.15) is active and may restrict the assignment of certain tasks to certain vehicles, adding a layer of combinatorial complexity to the routing decisions.

- **Fleet composition** consists of 15 vehicles in total, of which 10 are conventional non-electric trucks and 5 are electric trucks. All vehicles share the same maximum capacity $Q = 2$ and the same maximum working time $T_{\max} = 480$ minutes, corresponding to the full planning horizon. In practice, the optimal solutions never require the full fleet, confirming that the fleet size is sufficiently large to guarantee feasibility across all instance sizes tested.

Finally, for the sensitivity analysis described in Section 5.5, a set of modified instances is derived from one of the 10-client instances described above. Each variant alters a single parameter relative to this base instance, namely the width of the time windows, the proportion of hazardous tasks, the number of hazardous-certified vehicles, or the total fleet size, while keeping all other parameters unchanged, in order to isolate the effect of each factor on the computational difficulty of the problem.

5.3 Pareto Front Generation

As discussed in Section 2.3.3 and formalized in Section 4.6, the ε-constraint method is used to generate the Pareto front between the two objectives of the model. The method works by designating one objective as the primary criterion to be minimized, while the other is converted into an inequality constraint bounded by a parameter ε. By solving the resulting single-objective problem for a sequence of decreasing values of ε, a set of Pareto-optimal solutions is obtained, each corresponding to a different balance between the two objectives.

In this implementation, total transportation time is taken as the primary objective, and total transportation cost is bounded above by ε. The range of ε is determined by first solving two single-objective problems: one minimizing total transportation time without any cost restriction, which yields the minimum achievable time and its associated cost; and one minimizing total transportation cost without any time restriction, which yields the minimum achievable cost. These two solutions define the endpoints of the Pareto front and establish the interval within which ε is varied.

Starting from the cost value associated with the time-optimal solution, ε is decreased by 0.01 monetary units at each iteration, and the model is solved again. This small, fixed decrement is applied uniformly across all instances and is deliberate for the following reason. When minimizing time alone, the model may produce multiple solutions with identical total time but different costs. One possible reason is that the time objective is insensitive to the order in which nodes of equal travel distance are visited. A concrete example of this arises when a vehicle must visit two client nodes before returning to a port: the two possible orderings may yield the same total travel time but different fuel costs, because fuel consumption depends on the load carried on each arc, and the longer arc is cheaper to travel when the vehicle is less loaded. By decreasing ε by a small amount at each step, the method provides a high-resolution approximation of the efficient

frontier, ensuring that for each time value on the Pareto front, the solution returned is also cost-optimal among all solutions achieving that time, and reducing the risk of missing non-dominated points to a minimum.

Since the lower bound on cost is known from the cost-minimization problem solved at the outset, the constraint cost $\leq \varepsilon$ never renders the problem infeasible: the procedure stops naturally when ε reaches the minimum achievable cost, at which point no further improvement is possible.

This procedure is applied independently to every instance described in Section 5.2, including the full cumulative sequence used for the scalability analysis in Section 5.4 and the independent instances generated at fixed instance sizes, in order to obtain a complete Pareto front for each one.

5.4 Scalability Analysis

This section describes two complementary experiments designed to assess the scalability of the proposed model. The first examines how solution time and quality evolve as the size of the network grows, using the cumulative instance sequence introduced in Section 5.2. The second examines how much these results vary across different independently generated instances of the same size, using the additional independent instances generated at $|C| \in \{5, 10, 15\}$. Both experiments apply the same ε-constraint procedure described in Section 5.3, with all problem characteristics other than instance size and specific instance held at the default values described in Section 5.2; the effect of varying other problem characteristics is examined separately in the sensitivity analysis of Section 5.5.

The first experiment examines how solution time and quality evolve as the number of clients increases, using the cumulative instance sequence described in Section 5.2, with $|C| \in \{5, 8, 10, 12, 15, 20, 25\}$. Since the underlying decision variables, constraints, and nonzero coefficients of the model all scale with the size of the node set, as discussed in Chapter 4, larger instances are expected to be substantially more demanding to solve, and this experiment quantifies that growth directly.

For each instance size, the full ε-constraint procedure described in Section 5.3 is executed, and the following metrics are recorded at every step of the resulting Pareto front: solver status (optimal, time limit reached, or infeasible), solve time, the optimality gap reported by Gurobi upon termination, and the objective values obtained for both total transportation time and total transportation cost. The number of variables, constraints, and nonzero coefficients in the model is also recorded once per instance size, since these quantities remain constant across all ε values for a given instance, as discussed in Section 5.3.

One of the purposes of the experiment is to identify the instance size beyond which the model can no longer be reliably solved to optimality within the imposed time limit, and to characterize how the computational burden of generating a full Pareto front grows as the

underlying network expands. The results of this experiment are presented and discussed in Section 5.6.

The second experiment examines how much the computational results obtained for a given instance size depend on the particular instance generated, rather than on the size itself. For this purpose, two additional independent instances were generated at each of the sizes $|C| \in \{5, 10, 15\}$, as described in Section 5.2, so that three instances are available at each of these sizes: the corresponding instance from the cumulative sequence described above, plus two additional independently generated instances. Unlike the cumulative sequence, where larger instances are strict extensions of smaller ones, each instance is generated with entirely fresh random draws for client coordinates, demands, time windows, and hazardous material requirements, while depot and port locations remain fixed.

For each of the three instances at a given size, the full ε-constraint procedure described in Section 5.3 is executed independently, and the same metrics recorded above are collected: solver status, solve time, optimality gap, and objective values at each step of the Pareto front. Comparing these metrics across the three instances at each size makes it possible to assess whether the solve times and gaps observed for a given instance size in the cumulative sequence above are representative of that size in general, or whether they are specific to the particular client configuration used in the cumulative sequence.

This experiment is particularly relevant given that the difficulty of the underlying problem depends not only on the number of clients, but also on factors such as their spatial distribution and the specific combination of demands and time windows assigned to them, none of which are controlled for in the cumulative sequence alone. The results of this experiment are presented and discussed in Section 5.6.

Together, these two experiments provide complementary evidence regarding the computational scalability of the proposed formulation: the first isolates the effect of increasing problem size, whereas the second evaluates the extent to which computational performance depends on the specific structure of an instance of a given size.

The results of these experiments are presented and discussed in Section 5.6.2.

5.5 Sensitivity Analysis

While the scalability analysis presented in Section 5.4 examines how computational difficulty grows with the number of clients, it does not isolate the effect of other operational characteristics of the problem on that same difficulty. This section addresses that question directly, by fixing the instance size and systematically varying individual problem characteristics, one at a time, relative to a single base instance, as described in Section 5.5.1. The characteristics examined are the tightness of the time windows (Section 5.5.2), the proportion of hazardous tasks (Section 5.5.3), the proportion of hazardous-certified vehicles (Section 5.5.4), and the total fleet size (Section 5.5.5). This allows the effect of each characteristic to be isolated from the effects of instance size and from the effects of the

other characteristics, providing insight into which operational constraints are the most significant drivers of computational difficulty in practice.

5.5.1 Base Instance and Experimental Design

The sensitivity analysis is conducted using the 10-client instance A (Table A.4, Section 5.2) as a base instance. A fixed instance size of 10 clients was chosen as a compromise between the two extremes observed in the scalability analysis: small enough that the base case is solved efficiently, leaving sufficient room to observe an increase in difficulty as each characteristic is varied, yet large enough that the resulting problem is non-trivial and the effects of each modification are meaningful.

Unlike the scalability analysis, where the full ε-constraint procedure is executed to generate a Pareto front for each instance, the sensitivity analysis considers two single-objective solves per variant: one minimizing total transportation time without any cost restriction, and one minimizing total transportation cost without any time restriction. This choice is deliberate: the purpose of this analysis is not to characterize the trade-off between time and cost under each modified condition, but rather to assess how each characteristic affects the computational difficulty of solving the model under each objective individually, and whether that effect is consistent across both objectives or specific to one of them. This is sufficient for the purposes of this analysis and avoids the substantially larger computational effort that would be required to generate a full Pareto front for every combination of characteristics tested.

Each variant described in the following subsections modifies exactly one characteristic of the base instance, while keeping all other parameters, including client locations, demands, and the unmodified characteristics, identical to the base case. For each variant, the same metrics collected throughout the scalability analysis are recorded, separately for the time-minimization and cost-minimization solves: solver status, solve time, the optimality gap reported by Gurobi upon termination, and the resulting objective value. These metrics are compared against the corresponding values obtained for the unmodified base instance, allowing the effect of each characteristic on both objectives to be assessed directly. The results of this analysis are presented and discussed in Section 5.6.

5.5.2 Time Window Tightness

In the base instance, time windows are generated by drawing the earliest service time a_i from the interval $[0, 125]$ and the latest service time b_i from the interval $[350, 480]$ independently for each client, as described in Section 5.2. This produces relatively wide time windows, with a width of up to 480 minutes in the most permissive case.

To assess the effect of tighter scheduling constraints on computational difficulty, this variant instead draws a_i uniformly from the interval $[0, 280]$ for each client, and sets $b_i = a_i + 200$. This guarantees a fixed window width of 200 minutes for every client, regardless

of where the window falls within the planning horizon, reducing the scheduling flexibility available to the solver compared to the base instance. All other parameters, including client locations, demands, hazardous material requirements, and fleet composition, remain identical to the base instance described in Section 5.5.1. The resulting time windows for each client are reported in Table 5.2.

Table 5.2: Modified time windows, Time Window Tightness variant.

Client	a_i	b_i
C1	234	434
C2	23	223
C3	137	337
C4	8	208
C5	82	282
C6	175	375
C7	46	246
C8	152	352
C9	43	243
C10	276	476

5.5.3 Hazardous Material Proportion

In the base instance, hazardous material requirements are assigned randomly to each client node with a 25% probability of a task being hazardous, as described in Section 5.2. Given that 11 of the 15 vehicles in the fleet are certified to handle hazardous materials, this proportion results in a relatively loose combinatorial restriction on task assignment.

To assess the effect of a higher proportion of hazardous tasks on computational difficulty, this variant instead assigns hazardous material requirements with a 50% probability for each client node, doubling the expected number of hazardous tasks relative to the base instance. Since constraint (4.15) restricts hazardous tasks to certified vehicles only, a higher proportion of hazardous tasks increases the combinatorial pressure on the subset of certified vehicles, potentially forcing solutions that would otherwise be avoided under the base instance. All other parameters, including client locations, demands, time windows, the number of hazardous-certified vehicles, and fleet composition, remain identical to the base instance described in Section 5.5.1. The resulting hazardous material requirement for each client is reported in Table 5.3.

5.5.4 Vehicle Certification

In the base instance, 11 of the 15 vehicles in the fleet are certified to handle hazardous materials, as described in Section 5.2, leaving a relatively wide pool of vehicles available for the assignment of hazardous tasks under constraint (4.15).

Table 5.3: Modified hazardous material requirements, Hazardous Material Proportion variant.

Client	Hazardous Material
C1	No
C2	Yes
C3	No
C4	Yes
C5	No
C6	No
C7	Yes
C8	No
C9	Yes
C10	Yes

To assess the effect of a more restrictive certification ratio on computational difficulty, this variant reduces the number of hazardous-certified vehicles from 11 to 6, split proportionally to the overall fleet composition, with 4 certified conventional vehicles and 2 certified electric vehicles, matching the underlying 2:1 ratio of conventional to electric vehicles in the fleet. This ensures that the reduction in certified vehicles does not disproportionately remove one vehicle type from the pool available for hazardous tasks beyond what is already implied by its share of the fleet, isolating the effect of certification scarcity itself from any confound related to vehicle type availability. All other parameters, including client locations, demands, time windows, the proportion of hazardous tasks, and total fleet composition, remain identical to the base instance described in Section 5.5.1. The resulting vehicle certification and propulsion assignment is reported in Table 5.4.

5.5.5 Fleet Size

In the base instance, the fleet consists of 15 vehicles in total, of which 10 are conventional non-electric trucks and 5 are electric trucks, as described in Section 5.2. As noted previously, the optimal solutions for the base instance never require the full fleet, indicating that fleet size is not a binding constraint under the base configuration.

To assess the effect of a more constrained fleet on computational difficulty, this variant reduces the total fleet size from 15 to 8 vehicles, with 5 conventional non-electric trucks and 3 electric trucks. This reduction approaches more closely the number of vehicles actually used in the base instance's optimal solutions, increasing the likelihood that vehicle availability becomes a binding constraint and that the solver must work harder to find a feasible assignment of tasks to vehicles. Since constraint (4.15) restricts hazardous tasks to certified vehicles, the certification ratio is adjusted accordingly: of the 8 vehicles in this variant, 5 are certified to handle hazardous materials, split into 3 certified conventional vehicles and 2 certified electric vehicles, keeping a split proportionate to the underlying

Table 5.4: Vehicle certification and propulsion type, Certification Ratio variant.

Vehicle	Hazardous-capable	Electric
1	Yes	Yes
2	Yes	Yes
3	No	Yes
4	No	Yes
5	Yes	No
6	Yes	No
7	Yes	No
8	Yes	No
9	No	No
10	No	No
11	No	No
12	No	No
13	No	No
14	No	No
15	No	Yes

composition of the reduced fleet. All other parameters, including client locations, demands, time windows, and the proportion of hazardous tasks, remain identical to the base instance described in Section 5.5.1. The resulting vehicle certification and propulsion assignment is reported in Table 5.5.

Table 5.5: Vehicle certification and propulsion type, Fleet Size variant.

Vehicle	Hazardous-capable	Electric
1	Yes	Yes
2	Yes	Yes
3	No	Yes
4	Yes	No
5	Yes	No
6	Yes	No
7	No	No
8	No	No

5.6 Computational Results and Discussion

This section presents and discusses the computational results obtained through the experimental procedures described in Sections 5.3, 5.4, and 5.5. Section 5.6.1 reports the Pareto front generated for a representative instance using the ε-constraint method, illustrating the trade-off between total transportation time and total transportation cost that motivates the multi-objective formulation of Chapter 4. Section 5.6.2 reports the scalability results obtained from the cumulative instance sequence and the independent

instances introduced in Section 5.2, focusing on how solve time, optimality gap, and problem size evolve as the number of clients increases, and on how much these metrics vary across different independently generated instances of the same size. Section 5.6.3 reports the outcome of the sensitivity analysis described in Section 5.5, comparing the base instance against each modified variant across the four operational factors considered, namely time window tightness, hazardous material proportion, vehicle certification ratio, and fleet size. Section 5.6.4 synthesizes these findings in light of the research objectives set out in Chapter 1.

5.6.1 Approximation to the Pareto front

This section presents the approximation to the Pareto front obtained for the 10-client instance A (Table A.4), the same base instance used throughout the sensitivity analysis of Section 5.5. The front was generated using the ε-constraint procedure described in Section 5.3, with total transportation time taken as the primary objective and total transportation cost bounded above by ε.

Figure 5.2 shows the resulting Pareto front for the 10-client instance A, comprising 24 non-dominated solutions, with total transportation time ranging from 2161.61 to 2828.45 minutes and total transportation cost ranging correspondingly from 2487.14 down to 1636.48 monetary units.

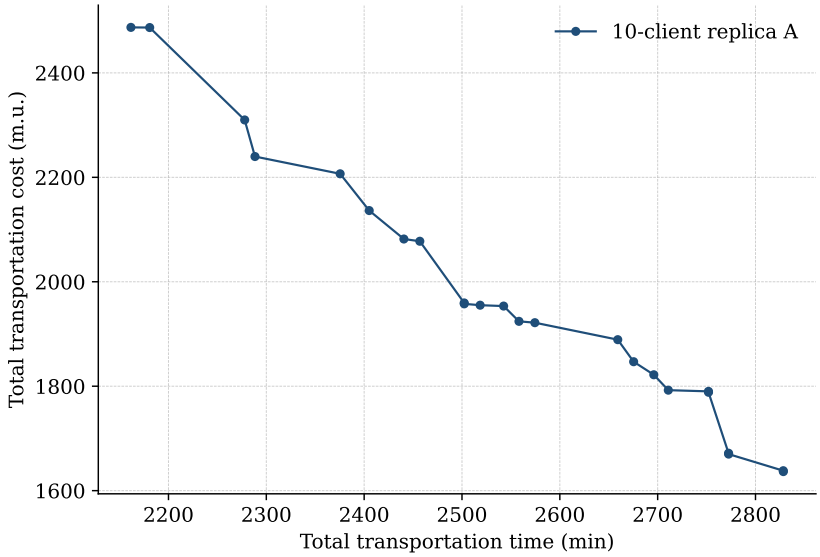


Figure 5.2: Pareto front between total transportation time and total transportation cost, 10-client instance A.

Table 5.6 reports the two endpoint solutions of the front in detail, together with the fleet composition used by each. The cost-optimal endpoint was solved to proven optimality (gap of 0%) in 2908 seconds, while the time-optimal endpoint reached the time limit of

3600 seconds with a remaining gap of 4.95%. The complete set of 24 non-dominated solutions underlying this front is reported in Appendix B.

Table 5.6: Endpoint solutions of the Pareto front, 10-client instance A.

Solution	Time (min)	Cost (m.u.)	Gap (%)	Run time (s)	Conv.	Elec.
Time-optimal	2161.61	2487.14	4.95	3600	6	0
Cost-optimal	2828.45	1636.48	0.00	2908	3	5

To characterize how vehicle allocation shifts along the front, the fleet composition of each non-dominated solution, namely the number of conventional and electric vehicles used, was also recorded. Figure 5.3 shows this breakdown across all 24 solutions, plotted against the same total transportation time axis as Figure 5.2 for direct comparison. The number of electric vehicles used increases monotonically along the front, from 0 at the time-optimal endpoint to 5, the entire available electric fleet, at the cost-optimal endpoint. The number of conventional vehicles used follows the opposite trend overall, decreasing from 6 to 3 vehicles, though not strictly monotonically. The total number of vehicles used also increases along the front, from 6 at the time-optimal endpoint to 8 at the cost-optimal endpoint.

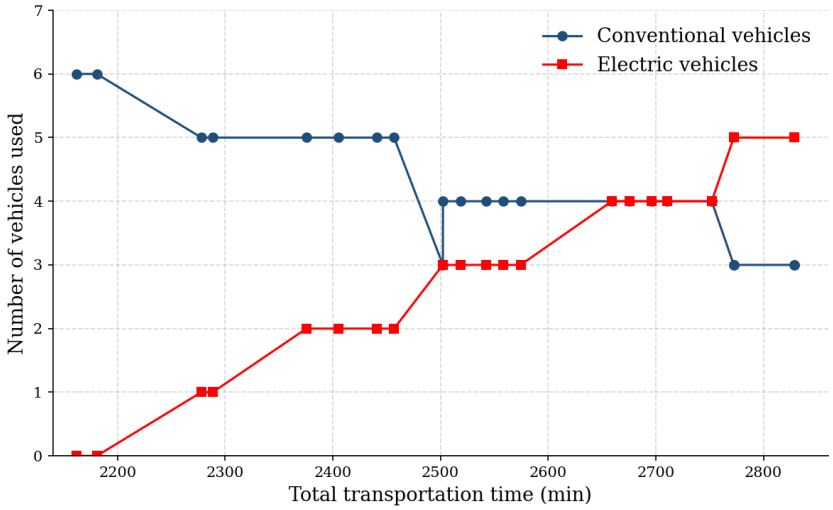


Figure 5.3: Number of conventional and electric vehicles used along the Pareto front, 10-client instance A.

To make this shift in vehicle allocation concrete, Figure 5.4 shows the actual routes generated for both endpoint solutions. In the time-optimal solution, all 6 vehicles used are conventional, and routes tend to favor the shorter, more direct port-to-client legs near the coast. In the cost-optimal solution, 5 of the 8 vehicles used are electric, and the work is spread across two additional vehicles: the average route length per vehicle drops from 600.4 km to 486.6 km, even though the total distance traveled across the fleet

increases slightly, from 3602.7 km to 3893.0 km. This is consistent with the lower travel speed of electric vehicles (70 km/h against 100 km/h for conventional trucks): splitting the workload across more, shorter routes allows each vehicle to remain within its working-time limit despite traveling more slowly.

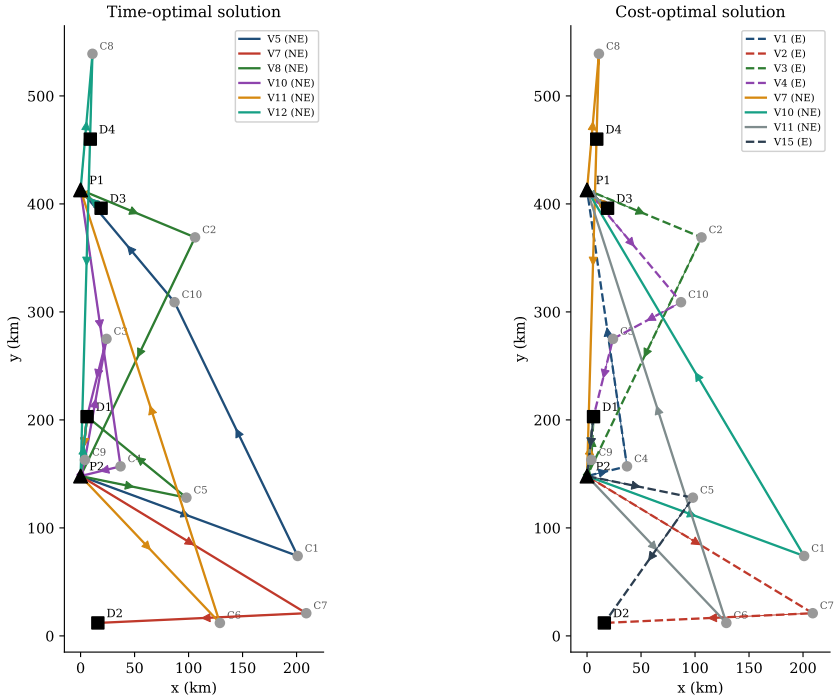


Figure 5.4: Vehicle routes for the time-optimal and cost-optimal solutions, 10-client instance A. Squares denote depots, triangles denote ports, circles denote clients. Dashed lines indicate electric vehicles.

The full route sequences corresponding to both solutions are listed in Appendix A.3.

The Pareto front above confirms the central premise motivating the multi-objective formulation of this thesis: total transportation time and total transportation cost cannot be minimized simultaneously, and the 24 non-dominated solutions trace out a range of intermediate compromises between the two. The mechanism behind this trade-off is made explicit by the fleet composition data reported alongside the front. As ε is relaxed and cost is allowed to decrease, the model progressively substitutes electric vehicles for conventional ones, reaching full use of the available electric fleet (5 of 5 vehicles) at the cost-optimal endpoint, while the time-optimal endpoint uses none. This is a direct consequence of the cost and speed parameters adopted in Section 4.2: electric vehicles are an order of magnitude cheaper to operate per kilometer ($c^{\text{elec}} = 0.2$ against $c^{\text{fuel}} = 2$), but travel at 70 km/h against 100 km/h for conventional trucks, so every kilometer shifted onto the electric fleet is both cheaper and slower.

This speed penalty also explains why the cost-optimal solution requires more vehicles overall (8, against 6 at the time-optimal endpoint) rather than simply reassigning the

same six routes to slower vehicles. Because total working time per vehicle is capped at $T^{\max} = 480$ minutes (constraint (4.14)), a slower vehicle covering the same itinerary as a conventional one would risk exceeding this limit; the model compensates by splitting the workload across additional, shorter routes. This is reflected in the route-level statistics: average route length per vehicle decreases from 600.4 km at the time-optimal endpoint to 486.6 km at the cost-optimal endpoint, even though the total distance traveled across the fleet increases slightly, from 3602.7 km to 3893.0 km. The trade-off explored by the Pareto front is therefore not only a trade-off between two abstract objective values, but a concrete trade-off between route efficiency and fleet utilization: minimizing cost is cheaper per kilometer, but only achievable by deploying more vehicles on individually shorter, less consolidated routes. The route maps shown in Figure 5.4 make this reallocation visible at the level of individual vehicles.

One asymmetry in the results is worth noting without over interpreting it: the cost-optimal endpoint was solved to proven optimality in 2908 seconds, while the time-optimal endpoint reached the 3600-second time limit with a 4.95% gap remaining. Since both solves address the same instance and the same model size, this difference is more likely attributable to the specific structure of the search space induced by each objective than to any inherent asymmetry between the two criteria; a definitive explanation would require examining the branch-and-bound behavior directly, which falls outside the scope of this analysis.

Finally, these findings are based on a single base instance, and the magnitude of the trade-off, particularly the exact point at which vehicle allocation shifts, should not be assumed to generalize unchanged to other instance sizes or configurations. The qualitative direction of the effect, however, namely that minimizing cost favors electric vehicles at the expense of fleet size and individual route length, follows directly from the cost and speed parameters of the model itself, and would be expected to hold wherever those parameters apply.

5.6.2 Scalability

This section reports the results of the scalability experiments described in Section 5.4, covering both the growth of computational difficulty with instance size and the variability of that difficulty across independent instances of the same size.

5.6.2.1 Growth with Instance Size

Table 5.7 reports the number of binary and continuous variables, constraints, and nonzero coefficients for each instance size in the cumulative sequence. As expected, all four quantities grow with $|C|$, since the underlying sets of arcs and tasks both expand with the addition of new client nodes.

Table 5.7: Model size statistics for the cumulative instance sequence.

$\|C\|$	Binary vars	Continuous vars	Constraints	Nonzeros
5	6735	4980	35241	120265
8	11055	8220	64584	230545
10	14535	10830	91196	332040
12	18495	13800	124288	459060
15	25335	18930	187651	705015
20	39135	29280	335856	1282690
25	55935	41880	547061	2112315

Figure 5.5 shows the optimality gap of every non-dominated solution generated across the Pareto front of each instance size, illustrating how computational difficulty evolves as $|C|$ increases. At $|C| = 5$ and $|C| = 8$, every solution in the front was solved to proven optimality (0% gap), with run times never exceeding 1209 seconds. At $|C| = 10$, most solutions still reached proven optimality, with only 4 of the 11 points reaching the 3600-second time limit, and a maximum remaining gap of 4.27%. From $|C| = 12$ onward, every solution in the front reached the time limit without closing the gap: at $|C| = 12$, gaps range from 11.35% to 27.66%; at $|C| = 15$, from 9.58% to 18.34%. Notably, the maximum gap observed at $|C| = 15$ is lower than at $|C| = 12$, despite the larger instance size. At $|C| = 20$, only a partial front of 5 solutions was generated, with gaps ranging from 15.59% to 49.74%; given the combination of large remaining gaps and long run times, the ε-constraint sweep was discontinued before a full front could be obtained, and these 5 points are reported as the final result for this instance size. At $|C| = 25$, no feasible or optimal solution was found within the time limit. The complete Pareto front data for every instance size, including each individual point underlying Figure 5.5, is reported in Section B.1.

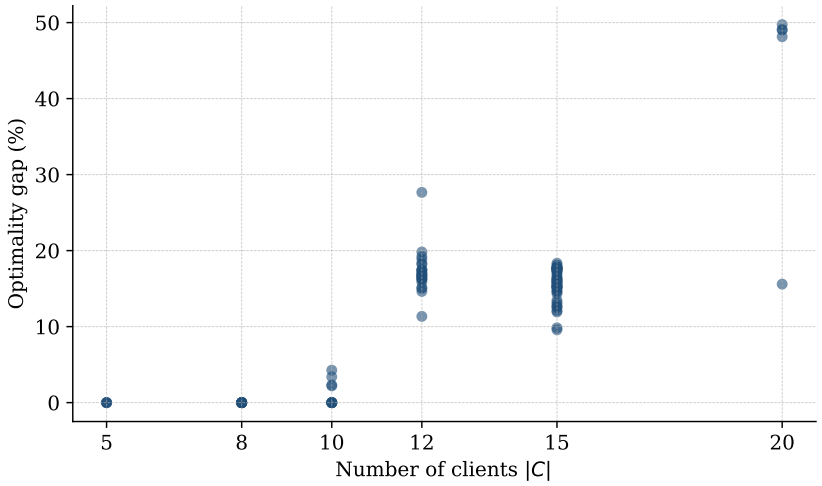


Figure 5.5: Optimality gap of all non-dominated solutions, by number of clients.

Figure 5.6 shows the Pareto fronts of the cumulative-sequence instance at $|C| \in \{5, 10, 15\}$ overlaid on a single set of axes. As instance size increases, the front shifts toward higher time and cost values, reflecting the larger number of tasks to be served, and becomes visibly less smooth: the $|C| = 15$ front shows irregular local segments not present at $|C| = 5$ or $|C| = 10$, consistent with the fact that none of its 44 solutions were solved to proven optimality, unlike the fully optimal $|C| = 5$ front.

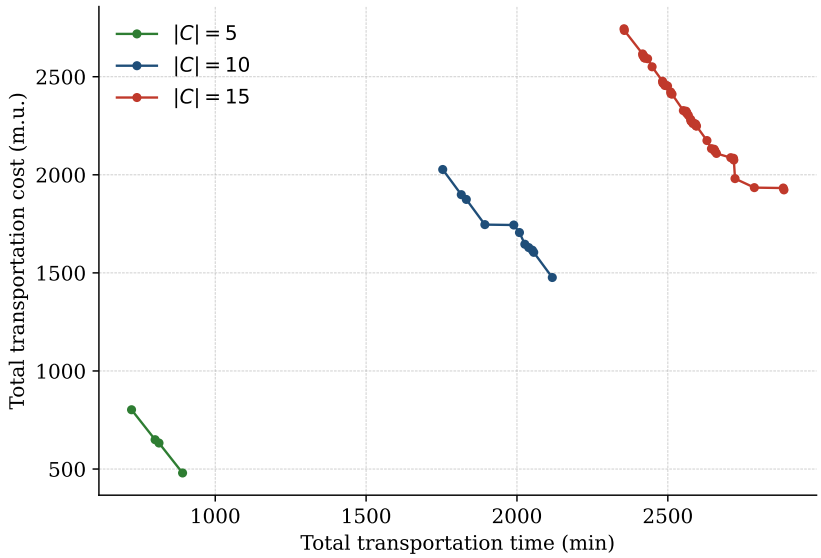


Figure 5.6: Pareto fronts of the cumulative-sequence instance, by number of clients.

The growth in computational difficulty above is not gradual: instances of $|C| = 5$ and $|C| = 8$ are solved to proven optimality within minutes, yet from $|C| = 12$ onward every single solution in the front reaches the 3600-second time limit without closing the gap. This pattern is consistent with the structural growth reported in Table 5.7, where the number of nonzero coefficients grows from 120 265 at $|C| = 5$ to over 2.1 million at $|C| = 25$, but the abruptness of the transition, rather than a smooth degradation, points to a combinatorial threshold typical of NP-hard routing problems: below a certain size, the branch-and-bound search exhausts the feasible region comfortably, while beyond it, the same search is overwhelmed almost immediately, with little middle ground. The $|C| = 10$ instances sit close to this threshold, which is itself informative, as discussed further below.

The results also show that computational difficulty does not grow monotonically with instance size in every case: the maximum gap observed at $|C| = 15$ (18.34%) is lower than at $|C| = 12$ (27.66%). This is consistent with the well-known property of NP-hard problems that instance difficulty depends not only on size but also on the specific structure and data of each instance, a larger instance is not necessarily harder to solve than a smaller one. No general conclusions should be drawn from this single observation.

At the upper end of the sizes tested, the results indicate a practical ceiling for this formulation under the computational setup used. At $|C| = 20$, the ε-constraint sweep was

discontinued after only 5 solutions, with gaps reaching as high as 49.74%, values too large to interpret as a meaningful approximation of the true front. At $|C| = 25$, no feasible or optimal solution was found at all within the time limit. Taken together with the structural growth reported in Table 5.7, this suggests that exact solution of this formulation, as implemented, is practically limited to instances of approximately 15 clients or fewer per planning instance; larger hinterland networks would likely require either a decomposition strategy, a tightened formulation, or a heuristic or metaheuristic approach of the kind reviewed in Section 2.3.2, rather than direct application of the exact MILP model in its current form.

5.6.2.2 Variability Across Instances

Table 5.8 compares the three instances available at each of $|C| \in \{5, 10, 15\}$: the cumulative-sequence instance and the two additional independent instances. Unlike the structural statistics reported in Table 5.7, the number of binary and continuous variables and the number of constraints are identical across all three instances of a given size, since these depend only on $|N|$, $|K|$, and $|A|$. The number of nonzero coefficients varies slightly between instances of the same size, a consequence of the random return indicator δ_i, as discussed in Section 5.6.2.1.

Table 5.8: Variability of solve difficulty across independent instances of the same size.

$\|C\|$	Instance	Points	Gap range (%)	At time limit
5	Cumulative	4	0.00	0 / 4
5	Instance A	12	0.00	0 / 12
5	Instance B	11	0.00	0 / 11
10	Cumulative	11	0.00 – 4.27	4 / 11
10	Instance A	24	0.00 – 13.87	22 / 24
10	Instance B	19	8.53 – 13.18	19 / 19
15	Cumulative	44	9.58 – 18.34	44 / 44
15	Instance A	31	7.64 – 13.20	31 / 31
15	Instance B	27	8.11 – 12.87	27 / 27

The three instances differ markedly in difficulty at $|C| = 10$: the cumulative instance solves most of its front to proven optimality, while Instance A reaches the time limit on all but 2 of its 24 points, and Instance B never reaches proven optimality on any of its 19 points. At $|C| = 15$, by contrast, the three instances behave consistently with one another: none of the 102 solutions across the three fronts reach proven optimality, and the gap ranges largely overlap (7.64–18.34% across all three combined). The complete Pareto front data for every instance summarized in Table 5.8 is reported in Appendix B (Sections B.1 and B.2).

Instance size alone does not determine computational difficulty. At $|C| = 10$, the specific spatial distribution of clients and the particular combination of demands, time

windows, and return indicators can matter as much as $|C|$ itself in determining how hard a given instance is to solve. At $|C| = 15$, by contrast, this distinction disappears: none of the 102 solutions across the three instances reach proven optimality, and their gap ranges overlap substantially. This suggests that, somewhere between 10 and 15 clients, the problem transitions from a regime where instance-specific structure can still make a meaningful difference to one where the combinatorial difficulty of the size class dominates regardless of the particular configuration, consistent with the threshold behavior discussed above.

5.6.3 Sensitivity Analysis

This section reports the results of the sensitivity analysis described in Section 5.5, comparing the base instance (10-client instance A) against each of the four single-factor variants, for both the time-minimizing and cost-minimizing solves.

Table 5.9 reports the objective values, optimality gap, and run time obtained for each variant and each objective, alongside the corresponding base instance values for reference.

Table 5.9: Sensitivity analysis: objective values, gap, and run time by variant.

Variant	Objective	Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
Base	Time-min	2161.61	2487.14	4.95	3600
Base	Cost-min	2828.45	1636.48	0.00	2908
Time Window Tightness	Time-min	2383.66	2766.53	6.95	3600
Time Window Tightness	Cost-min	2994.91	1786.53	4.89	3600
Hazmat Proportion	Time-min	2161.61	2529.16	6.17	3600
Hazmat Proportion	Cost-min	2828.45	1636.48	0.00	1055
Certification Ratio	Time-min	2161.61	2529.16	0.00	1786
Certification Ratio	Cost-min	2772.53	1669.97	0.00	888
Fleet Size	Time-min	2277.95	2308.18	0.00	1471
Fleet Size	Cost-min	2558.18	1925.94	7.76	3600

Table 5.10 reports the fleet composition used in each solution, aggregating certified and non-certified vehicles of the same propulsion type into a single count per type.

Relative to the base instance, the Time Window Tightness variant increases both the gap and run time on the cost-minimizing solve, and is the only variant to increase the number of vehicles used on both objectives (8 and 9, against 6 and 8 for the base instance). The Hazmat Proportion variant leaves the cost-minimizing solution entirely unchanged from the base instance (2828.45 min, 1636.48 m.u., 0% gap), while increasing the gap on the time-minimizing solve from 4.95% to 6.17%. The Certification Ratio variant is solved to proven optimality on both objectives, in substantially less time than the base instance (1786 and 888 seconds, against 3600 and 2908 seconds). The Fleet Size variant is the only one with a different structural model size, since it reduces the number of vehicles

Table 5.10: Sensitivity analysis: fleet composition by variant.

Variant	Objective	Electric	Conventional	Total used
Base	Time-min	0	6	6
Base	Cost-min	5	3	8
Time Window Tightness	Time-min	0	8	8
Time Window Tightness	Cost-min	5	4	9
Hazmat Proportion	Time-min	0	6	6
Hazmat Proportion	Cost-min	5	3	8
Certification Ratio	Time-min	0	6	6
Certification Ratio	Cost-min	5	3	8
Fleet Size	Time-min	1	5	6
Fleet Size	Cost-min	3	4	7

from 15 to 8; it is solved to proven optimality on the time-minimizing objective, but reaches the time limit with a 7.76% gap on the cost-minimizing objective.

The four variants do not affect computational difficulty uniformly: two make the problem harder to solve, one makes it markedly easier, and one has an asymmetric effect that helps one objective while hurting the other. Taken together, they suggest that the effect of tightening a given operational constraint on solver performance depends on what is being constrained and how it interacts with each objective individually, rather than following a single rule such as "more restrictions always mean a harder problem."

The Time Window Tightness variant is the most straightforwardly difficult of the four: it increases the gap on both objectives relative to the base instance (4.95% to 6.95% for time-minimization; 0% to 4.89% for cost-minimization, with the cost-minimizing solve no longer reaching proven optimality at all), and it is the only variant to increase the number of vehicles used on both objectives (8 against 6 for time-minimization; 9 against 8 for cost-minimization). Narrower individual time windows leave less flexibility to sequence multiple clients into a single vehicle's route, so the model compensates by deploying additional vehicles, each covering fewer stops within its own tighter scheduling constraints.

The Hazardous Material Proportion variant has a more selective effect. The cost-minimizing solution is entirely unaffected in value, reaching the same time and cost (2828.45 min, 1636.48 m.u.) as the base instance, despite the proportion of hazardous tasks doubling from 25% to 50%; it does, however, solve considerably faster (1055 against 2908 seconds). This suggests that, for this particular instance, the cost-optimal vehicle-to-task assignment already happened to satisfy the additional certification requirements without modification, though this is likely a property of this specific random draw of hazardous tasks rather than a general result. The time-minimizing solution, by contrast, reaches the same total time as the base instance (2161.61 min) but at a higher cost (2529.16 against 2487.14 m.u.) and a higher gap (6.17% against 4.95%), indicating that the additional hazardous tasks force a different, costlier vehicle assignment even though the total

travel time achievable is unchanged.

The Certification Ratio variant produces the clearest positive result among the four: despite reducing the pool of certified vehicles from 11 to 6, both objectives are solved to proven optimality (0% gap), and substantially faster than the base instance (1786 and 888 seconds, against 3600 and 2908 seconds). Restricting which vehicles may serve which tasks shrinks the feasible region, and a smaller feasible region can leave fewer near-equivalent alternatives for the branch-and-bound search to distinguish between, allowing it to converge more quickly to a provably optimal solution. The fact that this same mechanism appears to operate both when reducing certified vehicles and, tentatively, when increasing instance size, suggests it may be a more general property of this formulation rather than a coincidence specific to either case, though confirming this would require direct inspection of the solver's internal behavior.

The Fleet Size variant shows the most asymmetric effect of the four. Reducing the fleet from 15 to 8 vehicles makes the time-minimizing solve markedly easier, reaching proven optimality (0% gap) in only 1471 seconds, against the base instance's unresolved 4.95% gap at the time limit. The cost-minimizing solve, however, becomes harder: a 7.76% gap remains at the time limit, where the base instance solved to proven optimality. A plausible explanation is that cost-minimization relies more heavily on having a large pool of electric vehicles available to substitute for conventional ones, as discussed in Section 5.6.1; with only 3 electric vehicles available in this variant, against 5 in the base fleet, the model has less room to maneuver toward its preferred cost-minimizing allocation, making the optimal assignment harder to both find and prove. This variant also produces a notable exception to a pattern observed consistently elsewhere: its time-minimizing solution uses 1 electric vehicle, whereas every other configuration examined, including the base instance and all other variants, uses none at all when minimizing time alone. This suggests that vehicle scarcity, not only cost, can be sufficient to justify using a slower electric vehicle: with a smaller conventional fleet, covering all clients within the working-time limit may require deploying the available electric vehicle even when the objective is purely to minimize total travel time, rather than electric vehicles being used exclusively as a cost-driven choice.

5.6.4 Synthesis

The results above address the third and fourth research objectives set out in Section 1.2: framing the problem as a multi-objective optimization task and generating a meaningful Pareto front, and analyzing the results in a way that yields operationally relevant insight rather than a single disconnected number. The Pareto front obtained for the base instance does more than confirm that time and cost cannot be minimized simultaneously; it traces a continuous, interpretable trade-off in which fleet composition itself is the adjustable mechanism, giving a decision-maker a concrete lever, namely how heavily to favor electric vehicles, rather than an abstract weighting between two costs. The scalability and sensitivity analyses extend this from a single instance to a broader operational picture:

they identify not only that difficulty grows with instance size, but where it grows sharply (between 10 and 15 clients), and which operational factors a planner can adjust, and at what computational cost, to keep a given instance tractable.

These findings carry direct implications for how a tool of this kind would be used in practice, within the broader NEXUS project framework that motivates this work (Section 1.1). The scalability results indicate that exact solution of this formulation is reliable only up to approximately 15 clients per planning instance; a real hinterland network of the scale served by the Port of Sines would almost certainly involve more daily transportation tasks than this. Direct application of the model as implemented would therefore require partitioning a full day's tasks into smaller sub-instances, for example by geographic zone or by depot, rather than solving the entire network in a single call, or substituting a heuristic or metaheuristic approach for the largest instances, as discussed in Section 2.3.2. The sensitivity results offer more specific guidance for such a deployment: tightening delivery time windows is the most computationally costly factor examined, increasing both the optimality gap and the number of vehicles required, suggesting that planners should preserve scheduling flexibility wherever client commitments allow it. The certification ratio and fleet size results are more nuanced, and should not be misread as straightforward operational recommendations: a smaller certified pool or a smaller fleet can make the underlying optimization problem easier or harder to solve to a given gap, but this is a statement about computational tractability, not about the quality or cost of the resulting operational plan. A planner should not, for instance, reduce the certified fleet specifically because doing so was observed to solve faster in this analysis; the appropriate certification ratio remains a question of which tasks the operator expects to handle, not of solver convenience.

Taken as a whole, the results support the central claim of this thesis, that explicitly modeling time and cost as separate, competing objectives yields a more informative and more flexible decision-support tool than a single-objective formulation would, while also delineating the boundary of instance sizes and operational configurations within which this particular exact formulation remains practical. A further boundary of the current implementation, identified during the validation of these results, is discussed in Chapter 6, alongside its implications for future extensions of this work.

Chapter 6

Conclusion

This chapter closes the thesis. Section 6.1 summarizes what was accomplished against the research objectives set out in Chapter 1. Section 6.2 discusses the alignment of this work with the United Nations Sustainable Development Goals. Section 6.3 discusses the main limitations of the model and of the computational analysis conducted. Section 6.4 outlines directions in which this work could be extended.

6.1 Summary and Contributions

The main purpose of this thesis was to develop a decision support tool for the operational planning of container truck fleets in port hinterland networks, structured around the five research objectives stated in Section 1.2. This section reviews what was accomplished against each of them.

The first objective, to review the relevant literature and characterize the problem in its operational and academic context, was addressed in Chapter 2. The review covered the classical Vehicle Routing Problem and its multi-depot extension, the main mathematical formulations and solution approaches proposed for this class of problems, and the problem variants closest to the one addressed here, including arc routing, green and multi-objective extensions, optimization under uncertainty, and machine learning-based approaches. This review identified a concrete gap: very few existing studies combine, within a single optimization framework, the full set of operational features that characterize real-world container drayage, namely a multi-depot fleet structure, vehicle heterogeneity, hazardous material restrictions, time windows at every node, driver working-time regulations, and the simultaneous management of full and empty containers.

The second objective, to build a mathematical model capturing this operational reality, was addressed in Chapters 3 and 4. Chapter 3 formally characterized the operational context and the assumptions underlying the model, and Chapter 4 introduced the resulting Mixed-Integer Linear Programming formulation in full: its sets, parameters, decision variables, two objective functions, and the constraints governing routing, timing, hazardous

material handling, and container load flow. The formulation explicitly incorporates every feature identified as a gap in Chapter 2, going beyond simplified routing models that address these features only in isolation.

The third objective, to frame the problem as a multi-objective optimization task and generate the Pareto front between total transportation time and total transportation cost, was addressed through the ε-constraint method described in Section 4.6 and applied in Section 5.6.1. For the base instance examined, this produced a complete front of 24 non-dominated solutions, mapping the full range of available trade-offs rather than collapsing the problem into a single weighted criterion.

The fourth objective, to analyze and interpret the results in a way that is operationally meaningful, was addressed in Section 5.6. Beyond confirming that time and cost cannot be minimized simultaneously, the analysis identified the specific mechanism driving the trade-off, namely the substitution of electric for conventional vehicles as cost is prioritized, and the consequences this has for fleet size and route structure. It further characterized how computational difficulty scales with instance size, identifying a practical solvability threshold of approximately 15 clients under the computational setup used, and isolated the effect of four distinct operational factors, time window tightness, hazardous material proportion, certification ratio, and fleet size, on solver performance, producing findings, such as the easier solvability of a more restricted certification ratio, that would not have been apparent from the scalability analysis alone. Because the cost objective is driven by fuel and energy consumption rather than distance alone, the resulting trade-off also carries an implicit environmental dimension, with cost-minimizing solutions tending to favor lower-emission electric vehicles even though emissions were not modeled explicitly.

The fifth objective, to contribute a well-documented and reusable optimization module within the broader NEXUS project framework, is reflected in the model's deliberate generality: the formulation is not calibrated to a single fixed network, and the sets, parameters, and constraints are structured so that depots, ports, fleet composition, and task characteristics can be reconfigured without altering the underlying model, as demonstrated by the sensitivity analysis itself, in which four substantially different operational configurations were derived from a single base instance with no change to the formulation.

Taken together, these contributions provide a decision-support framework that is both grounded in the operational complexity of real hinterland container distribution and capable of producing interpretable, actionable trade-off information, fulfilling the central aim of this thesis as set out in Section 1.1.

The relevance of this work has also been recognized externally: a preliminary version of these contributions was submitted to and accepted for presentation at the Optimization 2026 conference in Lisbon, organized by APDIO, the Portuguese Operations Research Society. The submitted abstract is included in Appendix C.

6.2 Sustainable Development Goals

The Sustainable Development Goals (SDGs) are a set of 17 global goals established by the United Nations in 2015 as part of the 2030 Agenda for Sustainable Development, adopted by all UN member states (United Nations, 2015). They address the world's most pressing economic, social, and environmental challenges, with a target horizon of 2030. The European Union has formally committed to implementing the SDGs across all policy areas, including transport and logistics.

Two of the 17 SDGs are directly relevant to the work presented in this thesis. SDG 9 (Industry, Innovation and Infrastructure) calls for building resilient and sustainable infrastructure and fostering innovation, of which efficient and resource-aware logistics systems are a key component. SDG 13 (Climate Action) recognizes the transport sector as responsible for approximately 22% of global CO_2 emissions and calls for urgent integration of climate change measures into planning and decision-making. The model developed in this thesis contributes to both goals by proposing a decision-support framework that optimizes hinterland container transport operations while explicitly accounting for the adoption of electric vehicles as a lower-emission alternative to conventional diesel trucks. Table 6.1 summarizes the alignment of this thesis with each goal and their specific targets.

Table 6.1: Alignment of this thesis with the United Nations Sustainable Development Goals.

SDG & Target	**Contribution**	**Indicators & Metrics**
SDG 9 Targets 9.4 and 9.5	Supports integration of electric trucks into heterogeneous freight fleets, promoting resource-efficient hinterland logistics. Proposes a novel MILP formulation advancing the scientific and technological capabilities of logistics planning.	Share of electric vehicles in optimal solutions; fuel cost reduction vs. all-conventional fleet; Pareto solutions favoring electric deployment; novelty of bi-objective MILP formulation in container drayage literature.
SDG 13 Target 13.2	Provides a decision-support tool enabling planners to evaluate time-cost trade-offs, with cost minimization incentivizing lower-emission vehicle allocation.	Distance share traveled by electric vehicles in cost-optimal solution; implicit CO_2 reduction from time-optimal to cost-optimal allocation.

The work presented in this thesis contributes, within the scope of operational logistics planning, to both SDG 9 and SDG 13. While emissions are not modeled as an explicit objective, the cost structure of the model creates a built-in incentive toward electric vehicle deployment, and the Pareto front generated provides decision-makers with the information needed to make climate-aware fleet allocation choices. Future extensions incorporating an explicit emissions objective, as discussed in Section 6.4, would strengthen this alignment

further.

6.3 Limitations

This section discusses the main limitations of the work presented in this thesis, covering both the scope of the mathematical model and the scope of the computational analysis conducted.

The current formulation does not support a vehicle visiting the same port or depot more than once within a single route; each vehicle is restricted to visiting every node at most once. Due to the time available for this work, it was not possible to extend the formulation to fully support this capability, and doing so is identified as a direction for future work in Section 6.4.

As noted in Section 5.2, the model is not calibrated against real operational data from the Port of Sines or any other hinterland network. All computational experiments were conducted on synthetically generated instances, designed to be structurally representative of the network described in Chapter 3 rather than to reproduce it exactly. The qualitative conclusions drawn in this thesis, particularly the trade-off mechanism discussed in Section 5.6.1, follow from the model's cost and speed parameters rather than from any property specific to the synthetic instances used, and should be expected to hold under real data with similar parameter values; the specific magnitudes reported, such as the exact point along the front at which vehicle allocation shifts, should not be assumed to transfer unchanged to a real operational setting.

The sensitivity analysis presented in Section 5.6.3 examines four operational factors starting from a single base instance, the 10-client instance A. This was a deliberate choice, made to isolate the effect of each factor individually under a consistent, manageable computational budget, rather than to characterize how these effects vary across instance sizes or specific instance configurations. The additional instance variability results of Section 5.6.2.2 show that computational difficulty can differ substantially between instances of the same size, which suggests that the magnitude, though likely not the direction, of the sensitivity effects reported in Section 5.6.3 may also vary if a different base instance were used.

The scalability analysis of Section 5.6.2.1 indicates that exact solution of this formulation, using the computational setup described in Section 5.1, is reliable only up to approximately 15 clients per planning instance, beyond which optimality gaps grow rapidly and, at $|C| = 25$, no feasible solution was obtained at all within the time limit. This is a limitation of the exact solution approach adopted, rather than of the underlying formulation, and a real deployment serving a larger daily volume of tasks would require either an instance decomposition strategy or a heuristic or metaheuristic solution method, as discussed in Section 5.6.4.

Finally, several of the solutions reported throughout Chapter 5, including the time-optimal endpoint of the base instance's own Pareto front, were obtained at the 3600-second time limit with a nonzero optimality gap rather than verified to global optimality. Although such solutions remain feasible and are the best available within the computational budget used, individual routing decisions within them are not guaranteed to be optimal, and should be interpreted with this in mind, particularly when used as illustrative examples rather than as proven optimal benchmarks.

6.4 Future Work

This section outlines directions for extending the work presented in this thesis, following directly from the limitations discussed in Section 6.3.

The most immediate extension is to support a vehicle visiting the same port and depot more than once within a single route, a capability not available in the current formulation. This would allow the model to exploit multi-visit routing where it is operationally advantageous, particularly in larger or more port-constrained instances, where the two ports available may otherwise act as a bottleneck.

A second direction is to extend the solution approach beyond exact methods. A heuristic or metaheuristic approach, of the kind reviewed in Section 2.3.2, or a matheuristic combining exact and heuristic components, could extend practical applicability to larger daily task volumes than those considered here. Independently of the solution method used, running the same experiments on a more powerful machine would also be expected to improve both the speed with which solutions are obtained and the quality, in terms of optimality gap, of the solutions found within a given time limit, particularly for the larger instances examined in Section 5.6.2.

A third direction is to calibrate and validate the model against real operational data from the Port of Sines hinterland network, rather than the synthetically generated instances used throughout this thesis. This would allow the qualitative findings of Chapter 5, particularly the trade-off between transportation time and cost, to be tested under realistic demand patterns, time window distributions, and network structure, and would clarify which of the conclusions drawn here generalize beyond the synthetic setting.

A fourth direction is to broaden the sensitivity analysis presented in Section 5.6.3. The current analysis examines four operational factors individually, starting from a single base instance; combined-effects tests, varying more than one factor simultaneously, were considered during the design of this analysis but not carried out, and could reveal interactions between factors that are not visible when each is varied in isolation. Repeating the sensitivity analysis from a different base instance, or across multiple instance sizes, would also help establish how much of the observed sensitivity is attributable to the operational factors themselves, as opposed to the particular instance examined.

A fifth direction concerns the theoretical foundations of the multi-objective formulation itself. A formal analysis of the degree of independence between the two objective functions, total transportation time and total transportation cost, would help establish whether they are truly conflicting across the entire feasible region or only partially so. Such an analysis could draw on the Pareto front data already generated in this work as a starting point, for example by examining the correlation between objective values across non-dominated solutions, and would strengthen the justification for the bi-objective formulation adopted in Section 4.4.

Finally, the model could be extended by incorporating additional objectives, such as carbon emissions, alongside the time and cost objectives already considered, extending the bi-objective formulation of Section 4.4 into a many-objective one. Since the cost objective adopted in this thesis is driven primarily by fuel and energy consumption (Section 4.2), it already serves, to some extent, as an indirect proxy for emissions, and the cost-minimizing solutions discussed in Section 5.6.1 can be loosely interpreted as also favoring lower-emission fleet allocations. An explicit emissions objective would refine this relationship rather than introduce an entirely new consideration, allowing cost and emissions to be traded off independently in cases where the two diverge, for example where electricity generation emissions vary by region or time of day.

A further direction concerns the integration of the optimization model into a fully operational Decision Support System (DSS). In its current form, the model produces a set of Pareto-optimal solutions that must be interpreted and selected externally. A true DSS would encapsulate this optimization core within an interactive graphical interface, allowing the decision-maker to engage directly in the decision process, for example, by visualizing the Pareto front, exploring the trade-offs between solutions, selecting one or more solutions for implementation, and adjusting input parameters in response to changing operational conditions. Such an interface would transform the model from a standalone analytical tool into a practical planning instrument, making the multi-objective optimization genuinely accessible to logistics operators without requiring expertise in mathematical programming.

References

Dirk Steenken, Stefan Voß, and Robert Stahlbock. Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, 26(1):3–49, 2004. doi: 10.1007/s00291-003-0157-z.

APS – Ports of Sines and the Algarve Authority. Hinterland, 2026. URL https://www.apsinesalgarve.pt/port-of-sines/the-port/hinterland/. Accessed: June 2026.

International Organization for Standardization. ISO 668:2020 – series 1 freight containers – classification, dimensions and ratings. International standard, ISO, Geneva, Switzerland, 2020.

George B. Dantzig and John H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. doi: 10.1287/mnsc.6.1.80.

Jairo R. Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiménez, and Nilson Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129, 2015. doi: 10.1016/j.cie.2014.10.029.

Roberto Baldacci, Eleni Hadjiconstantinou, and Aristide 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.

Bruce L. Golden, Thomas L. Magnanti, and Hien Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977. doi: 10.1002/net.3230070203.

George B. Dantzig, Delbert R. Fulkerson, and Selmer M. 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.

Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329, 1960. doi: 10.1145/321043.321046.

Eduardo Lalla-Ruiz and Martijn Mes. Mathematical formulations and improvements for the multi-depot open vehicle routing problem. *Optimization Letters*, 15(1):271–286, 2021. doi: 10.1007/s11590-020-01594-z.

D.G. Mogale, Abhijeet Ghadge, and Sanjay Kumar Jena. Modelling and optimising a multi-depot vehicle routing problem for freight distribution in a retail logistics network. *Computers & Industrial Engineering*, 207:111315, 2025. doi: 10.1016/j.cie.2025.111315.

Luis F. Galindres-Guancha, Eliana M. Toro-Ocampo, and Ramón A. Gallego Rendón. Multi-objective MDVRP solution considering route balance and cost using the ILS metaheuristic. *International Journal of Industrial Engineering Computations*, 9(1):33–46, 2018. doi: 10.5267/j.ijiec.2017.5.002.

P. Surekha and S. Sumathi. Solution to multi-depot vehicle routing problem using genetic algorithms. *World Applied Programming*, 1(3):118–131, 2011.

L. A. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963. doi: 10.1109/TAC.1963.1105511.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

Xiang Song, Dylan Jones, Nasrin Asgari, and Tim Pigden. Multi-objective vehicle routing and loading with time window constraints: a real-life application. *Annals of Operations Research*, 291(1):799–825, 2020. doi: 10.1007/s10479-019-03205-2.

Peter C. Fishburn. Lexicographic orders, utilities, and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974. doi: 10.1287/mnsc.20.11.1442.

A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*. John Wiley, New York, 1961.

Yacov Y. Haimes, Leon S. Lasdon, and David A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971. doi: 10.1109/TSMC.1971.4308298.

Ameni Kraiem, Jean-François Audy, and Amina Lamghari. Mixed integer linear programming model for a multi-depot arc routing problem with different arc types and flexible assignment of end depot. *Transportation Research Procedia*, 82:1109–1119, 2025. doi: 10.1016/j.trpro.2024.12.115.

Özgür Kabadurmuş, Mehmet Serdar Erdoğan, Yiğitcan Özkan, and Mertcan Köseoğlu. A multi-objective solution of green vehicle routing problem. *Logistics & Sustainable Transport*, 10(1):31–44, 2019. doi: 10.2478/jlst-2019-0003.

Yong Wang, Kevin Assogba, Jianxin Fan, Maozeng Xu, Yong Liu, and Haizhong Wang. Multi-depot green vehicle routing problem with shared transportation resource: Integration of time-dependent speed and piecewise penalty cost. *Journal of Cleaner Production*, 232:12–29, 2019. doi: 10.1016/j.jclepro.2019.06.179.

Tuan-Anh Dao and Ngoc-Anh Nguyen Thi. A multi-criteria optimization model for emission-concerned multi-depot vehicle routing problem with heterogeneous fleet. In *2018 International Conference on Applied Smart Systems (ICASS)*, pages 1–6. IEEE, 2018. doi: 10.1109/ICASS.2018.8651943. URL https://ieeexplore.ieee.org/document/8651943.

Xiaofeng Xu, Ziru Lin, Xiang Li, Changjing Shang, and Qiang Shen. Multi-objective robust optimisation model for MDVRPLS in refined oil distribution. *International Journal of Production Research*, 60(22):6772–6792, 2022. doi: 10.1080/00207543.2021.1887534.

Zhongxin Zhou, Minghu Ha, Hao Hu, and Hongguang Ma. Half open multi-depot heterogeneous vehicle routing problem for hazardous materials transportation. *Sustainability*, 13(3):1262, 2021. doi: 10.3390/su13031262.

Xixing Li, Chao Gao, Jing Wang, Hao Tang, Ting Ma, and Feng Yuan. Research on multi-objective green vehicle routing problem with time windows based on the improved non-dominated sorting genetic algorithm III. *Symmetry*, 17(5):734, 2025. doi: 10.3390/sym17050734.

Aigerim Bogyrbayeva, Meraryslan Meraliyev, Taukekhan Mustakhov, and Bissenbay Dauletbayev. Machine learning to solve vehicle routing problems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):4754–4772, 2024. doi: 10.1109/TITS.2023.3334976.

MohammadReza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem. *Advances in Neural Information Processing Systems*, 31:9861–9871, 2018.

Zhenwei Wang, Guoqiang Zhong, and Junyu Dong. Towards constraint-based adaptive hypergraph learning for solving vehicle routing: An end-to-end solution. arXiv preprint arXiv:2503.10421, 2025.

Yifan Zhou, Lingbo Kong, and Hao Wang. A multiobjective edge-based learning algorithm for the vehicle routing problem with time windows. *Information Sciences*, 715:122223, 2025. doi: 10.1016/j.ins.2025.122223.

Öko-Institut. Real-world data analysis of battery electric trucks. Technical report, Öko-Institut e.V., 2024. Available at: https://www.oeko.de/fileadmin/oekodoc/Real-world-data-analysis-of-battery-electric-trucks.pdf.

European Parliament and Council of the European Union. Directive 2008/68/ec on the inland transport of dangerous goods, 2008. URL https://eur-lex.europa.eu/EN/legal-content/summary/inland-transport-of-dangerous-goods.html. Accessed: June 2026.

U.S. Bureau of Transportation Statistics. Commercial truck fuel consumption by average weight. Figure 6-8 in: *Transportation Statistics Annual Report 2024*, 2024. Data from 2021 Vehicle Inventory and Use Survey (VIUS). Available at: https://www.bts.gov/browse-statistical-products-and-data/info-gallery/commercial-truck-fuel-consumption-average-weight.

International Road Union. Driving profitability: Fuel, tolling and cost trends in the eu. Technical report, IRU, 2024. Available at: https://www.iru.org.

Hussein Basma, Arash Saboori, and Felipe Rodríguez. Total cost of ownership for tractor-trailers in Europe: Battery electric versus diesel. Technical report, International Council on Clean Transportation, 2021. Available at: https://theicct.org/sites/default/files/publications/TCO-BETs-Europe-white-paper-v4-nov21.pdf.

United Nations. Transforming our world: The 2030 agenda for sustainable development, 2015. URL https://sdgs.un.org/2030agenda. Resolution adopted by the General Assembly on 25 September 2015.

Appendix A

Instance Data

This appendix reports the data underlying the instances, solutions, and sensitivity analysis variants used throughout Chapter 5. Section A.1 reports the full client node data for the cumulative instance sequence, and Section A.2 the corresponding data for the independent instances; depot and port coordinates are fixed across all instances and are listed in Table 5.1 (Section 5.2), and are not repeated here. For every client in Sections A.1 and A.2, the table columns report the node coordinates (x, y), the processing time p_i, the time window $[a_i, b_i]$, the demand d_i (1 for a 20-foot container, 2 for a 40-foot container), the return indicator δ_i (Yes if the container is returned after service), and the hazardous material requirement h_i (Yes if the task requires hazardous material handling). Section A.3 reports the full vehicle route sequences corresponding to the two endpoint solutions discussed in Section 5.6.1.

A.1 Cumulative Instance Sequence

Table A.1 lists all 25 client nodes used in the cumulative sequence described in Section 5.2. The column *Introduced at* indicates the smallest instance size $|C|$ at which each client first appears; for example, clients introduced at size 5 are present in every instance of the cumulative sequence, while clients introduced at size 25 appear only in the largest instance.

Table A.1: Client node data for the cumulative instance sequence.

Client	**Introduced at**	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	5	$(110, 74)$	31	$[53, 458]$	1	No	No
C_2	5	$(152, 422)$	50	$[122, 479]$	2	Yes	Yes
C_3	5	$(23, 146)$	40	$[23, 395]$	1	No	No
C_4	5	$(28, 213)$	23	$[94, 394]$	1	Yes	No
C_5	5	$(18, 181)$	49	$[125, 378]$	1	Yes	No
C_6	8	$(197, 513)$	12	$[2, 470]$	2	Yes	Yes
C_7	8	$(79, 43)$	17	$[0, 480]$	2	No	No
C_8	8	$(65, 332)$	10	$[8, 465]$	1	No	No
C_9	10	$(202, 147)$	4	$[5, 480]$	1	Yes	Yes
C_{10}	10	$(23, 301)$	31	$[13, 412]$	2	No	No
C_{11}	12	$(123, 441)$	6	$[0, 460]$	1	No	No
C_{12}	12	$(17, 197)$	8	$[3, 478]$	1	No	No
C_{13}	15	$(146, 225)$	13	$[29, 465]$	2	No	No
C_{14}	15	$(171, 433)$	21	$[21, 480]$	2	No	Yes
C_{15}	15	$(65, 442)$	4	$[8, 412]$	1	No	No
C_{16}	20	$(142, 523)$	31	$[17, 413]$	2	No	Yes
C_{17}	20	$(158, 431)$	23	$[65, 380]$	1	Yes	No
C_{18}	20	$(3, 109)$	15	$[71, 476]$	1	Yes	Yes
C_{19}	20	$(56, 23)$	9	$[23, 436]$	2	No	No
C_{20}	20	$(126, 356)$	12	$[19, 399]$	2	No	No
C_{21}	25	$(101, 39)$	9	$[14, 390]$	2	Yes	No
C_{22}	25	$(209, 153)$	23	$[40, 435]$	1	Yes	No
C_{23}	25	$(2, 501)$	17	$[66, 478]$	1	No	Yes
C_{24}	25	$(5, 427)$	43	$[92, 438]$	2	No	No
C_{25}	25	$(16, 143)$	3	$[118, 469]$	2	No	No

A.2 Independent Instances

Tables A.2–A.7 list the client node data for the six independent instances described in Section 5.2, generated separately at sizes $|C| \in \{5, 10, 15\}$. Together with the corresponding instance from the cumulative sequence in Section A.1, these provide three independent instances at each of the three sizes used in the variability analysis of Section 5.4.

Table A.2: Client node data, 5-client instance A.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(202, 498)$	15	$[100, 472]$	2	No	No
C_2	$(128, 270)$	6	$[23, 438]$	2	No	No
C_3	$(25, 413)$	24	$[0, 451]$	1	Yes	Yes
C_4	$(2, 104)$	12	$[13, 479]$	2	Yes	No
C_5	$(41, 147)$	31	$[41, 401]$	1	No	No

Table A.3: Client node data, 5-client instance B.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(193, 255)$	10	$[34, 438]$	2	No	No
C_2	$(27, 19)$	34	$[47, 471]$	1	No	Yes
C_3	$(98, 398)$	13	$[87, 420]$	2	No	No
C_4	$(167, 409)$	19	$[50, 389]$	1	No	No
C_5	$(145, 290)$	6	$[110, 420]$	1	Yes	Yes

Table A.4: Client node data, 10-client instance A.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(201, 74)$	35	$[89, 480]$	1	Yes	No
C_2	$(106, 369)$	50	$[21, 457]$	2	Yes	Yes
C_3	$(24, 275)$	31	$[2, 412]$	1	No	No
C_4	$(37, 157)$	28	$[81, 462]$	2	Yes	No
C_5	$(98, 128)$	19	$[103, 398]$	2	No	No
C_6	$(129, 12)$	9	$[134, 420]$	2	Yes	No
C_7	$(209, 21)$	11	$[12, 412]$	1	No	Yes
C_8	$(11, 539)$	19	$[18, 454]$	2	Yes	No
C_9	$(4, 163)$	4	$[34, 469]$	1	No	Yes
C_{10}	$(87, 309)$	17	$[19, 412]$	1	No	Yes

Table A.5: Client node data, 10-client instance B.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(49, 391)$	35	$[89, 480]$	1	Yes	No
C_2	$(68, 376)$	50	$[21, 457]$	2	Yes	Yes
C_3	$(192, 8)$	31	$[2, 412]$	1	No	No
C_4	$(21, 31)$	28	$[81, 462]$	2	Yes	No
C_5	$(184, 279)$	19	$[103, 398]$	2	No	No
C_6	$(31, 145)$	9	$[134, 420]$	2	Yes	No
C_7	$(56, 73)$	11	$[12, 412]$	1	No	Yes
C_8	$(6, 531)$	19	$[18, 454]$	2	Yes	No
C_9	$(155, 430)$	4	$[34, 469]$	1	No	Yes
C_{10}	$(34, 95)$	17	$[19, 412]$	1	No	Yes

Table A.6: Client node data, 15-client instance A.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(70, 59)$	44	$[38, 479]$	1	No	No
C_2	$(53, 318)$	29	$[52, 469]$	2	Yes	No
C_3	$(182, 163)$	6	$[21, 456]$	1	Yes	No
C_4	$(114, 362)$	14	$[2, 469]$	2	Yes	Yes
C_5	$(51, 153)$	44	$[10, 428]$	2	No	Yes
C_6	$(5, 363)$	7	$[16, 468]$	1	No	Yes
C_7	$(42, 414)$	31	$[109, 440]$	1	No	Yes
C_8	$(206, 185)$	33	$[33, 400]$	1	Yes	No
C_9	$(8, 97)$	21	$[37, 462]$	2	Yes	No
C_{10}	$(42, 372)$	12	$[47, 397]$	2	No	No
C_{11}	$(126, 96)$	19	$[87, 376]$	1	Yes	No
C_{12}	$(29, 294)$	2	$[17, 415]$	2	Yes	No
C_{13}	$(200, 234)$	40	$[108, 448]$	1	Yes	No
C_{14}	$(53, 107)$	13	$[51, 419]$	2	No	Yes
C_{15}	$(159, 91)$	35	$[0, 401]$	2	No	No

Table A.7: Client node data, 15-client instance B.

Client	**Coordinates**	p_i	$[a_i, b_i]$	d_i	δ_i	h_i
C_1	$(169, 366)$	24	$[106, 408]$	1	No	No
C_2	$(132, 396)$	29	$[21, 464]$	2	Yes	Yes
C_3	$(203, 179)$	6	$[98, 423]$	2	Yes	No
C_4	$(53, 249)$	22	$[9, 443]$	2	Yes	No
C_5	$(124, 302)$	9	$[63, 370]$	1	No	No
C_6	$(9, 527)$	5	$[45, 443]$	2	Yes	Yes
C_7	$(47, 269)$	12	$[34, 479]$	1	No	No
C_8	$(104, 537)$	20	$[7, 394]$	1	Yes	No
C_9	$(83, 461)$	6	$[26, 452]$	2	Yes	Yes
C_{10}	$(30, 338)$	3	$[56, 424]$	2	No	No
C_{11}	$(166, 67)$	17	$[47, 468]$	2	No	Yes
C_{12}	$(192, 457)$	17	$[51, 423]$	1	No	No
C_{13}	$(115, 451)$	25	$[122, 404]$	2	Yes	Yes
C_{14}	$(198, 510)$	35	$[111, 440]$	1	No	Yes
C_{15}	$(137, 312)$	25	$[62, 453]$	2	No	No

A.3 Example Solution Routes

This section reports the full route sequences corresponding to the two endpoint solutions shown in Figure 5.4 (Section 5.6.1), for the 10-client instance A instance. Table A.8 lists the routes of the time-optimal solution, and Table A.9 those of the cost-optimal solution.

Table A.8: Vehicle routes, time-optimal solution, 10-client instance A.

Vehicle	Route	Distance (km)
5	D1 → P2 → C1 → C10 → P1 → D3	691.8
7	D1 → P2 → C7 → D2	493.1
8	D3 → P1 → C2 → P2 → C5 → D1	604.1
10	D3 → P1 → C4 → P2 → C9 → C3 → D1	525.7
11	D1 → P2 → C6 → P1 → D3	689.5
12	D3 → P1 → C8 → P2 → D1	598.5

Table A.9: Vehicle routes, cost-optimal solution, 10-client instance A.

Vehicle	Route	Type	Distance (km)
1	D1 → P2 → C4 → P1 → D3	Electric	377.6
2	D1 → P2 → C7 → D2	Electric	493.1
3	D3 → P1 → C2 → P2 → C9 → D1	Electric	440.9
4	D3 → P1 → C10 → C3 → D1	Electric	306.9
7	D3 → P1 → C8 → P2 → D1	Conventional	598.5
10	D1 → P2 → C1 → P1 → D3	Conventional	689.1
11	D1 → P2 → C6 → P1 → D3	Conventional	689.5
15	D1 → P2 → C5 → D2	Electric	297.4

Appendix B

Full Pareto Front Results

This appendix reports the complete Pareto front data underlying the Pareto front, scalability, and variability analyses of Sections 5.6.1 and 5.6.2, for every instance examined in this thesis. Only the two endpoint solutions of the 10-client instance A instance are discussed in the main text (Table 5.6); the complete front for this instance is included here for completeness, alongside every other instance. Section B.1 reports the cumulative-sequence instances, and Section B.2 the independent instances. No solution was found for the 25-client cumulative instance within the time limit, and it is therefore not included. Run time was not recorded for the 20-client cumulative instance, for which the ε-constraint sweep was discontinued before a full front was obtained, as discussed in Section 5.6.2.1.

B.1 Cumulative Instance Sequence

Table B.1: Full Pareto front, 5-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
722.56	802.29	0.00	56
800.71	649.59	0.00	103
813.56	632.66	0.00	91
891.71	479.96	0.00	42

Table B.2: Full Pareto front, 8-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
1378.56	1585.79	0.00	198
1400.94	1579.45	0.00	345
1437.83	1471.25	0.00	566

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
1456.71	1433.09	0.00	676
1479.10	1426.75	0.00	543
1515.99	1318.54	0.00	981
1545.17	1287.39	0.00	1209
1648.45	1218.46	0.00	591
1661.30	1201.54	0.00	1109
1673.38	1188.19	0.00	879
1680.18	1163.38	0.00	861
1739.45	1048.83	0.00	365

Table B.3: Full Pareto front, 10-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
1753.91	2027.08	0.00	2482
1815.48	1898.78	3.39	3600
1832.11	1874.42	4.27	3600
1893.64	1746.07	2.32	3600
1989.45	1744.02	2.20	3600
2008.29	1705.63	0.00	3451
2026.10	1645.99	0.00	3245
2038.94	1629.07	0.00	2987
2051.03	1615.72	0.00	3498
2055.57	1604.71	0.00	3129
2117.10	1476.36	0.00	2908

Table B.4: Full Pareto front, 12-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
1906.73	2204.06	11.35	3600
1968.30	2075.76	14.63	3600
1997.88	2064.58	16.21	3600
2001.46	2006.72	15.13	3600
2034.37	1936.40	17.45	3600
2059.41	1936.23	17.51	3600
2063.03	1878.42	18.87	3600
2067.57	1867.41	16.65	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2099.56	1804.87	16.88	3600
2128.12	1802.86	17.13	3600
2129.10	1739.06	18.31	3600
2224.92	1737.01	16.56	3600
2228.29	1732.50	16.31	3600
2232.02	1704.79	17.43	3600
2256.95	1674.52	16.98	3600
2282.87	1666.52	15.86	3600
2285.20	1657.09	18.23	3600
2292.38	1652.07	19.23	3600
2323.02	1535.16	27.66	3600
2351.58	1533.15	16.29	3600
2353.91	1523.72	15.03	3600
2437.71	1520.71	16.71	3600
2440.61	1519.46	19.82	3600

Table B.5: Full Pareto front, 15-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2354.83	2744.53	9.84	3600
2355.90	2735.76	9.58	3600
2416.41	2616.23	11.93	3600
2417.47	2607.46	12.11	3600
2420.90	2605.17	12.61	3600
2421.97	2596.40	12.52	3600
2423.87	2594.95	12.84	3600
2432.99	2591.83	13.15	3600
2447.89	2550.57	13.46	3600
2482.48	2476.87	14.34	3600
2483.54	2468.10	14.62	3600
2485.44	2466.65	14.64	3600
2489.94	2455.59	15.23	3600
2495.63	2454.76	14.94	3600
2499.10	2452.51	15.17	3600
2509.46	2422.27	15.67	3600
2510.53	2413.50	15.19	3600
2513.96	2411.21	15.29	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2551.51	2327.29	15.62	3600
2560.63	2324.17	15.50	3600
2561.70	2315.40	15.31	3600
2563.60	2313.94	15.93	3600
2568.14	2302.93	15.93	3600
2575.54	2282.91	16.12	3600
2576.60	2274.14	17.32	3600
2578.50	2272.68	16.31	3600
2582.99	2261.62	16.29	3600
2588.69	2260.79	16.74	3600
2592.16	2258.55	15.99	3600
2593.23	2249.78	17.01	3600
2595.08	2248.28	17.72	3600
2629.67	2174.58	16.45	3600
2644.57	2133.32	17.63	3600
2653.69	2130.20	17.37	3600
2654.76	2121.43	17.29	3600
2656.66	2119.98	17.46	3600
2661.20	2108.96	18.10	3600
2708.59	2087.29	17.74	3600
2717.71	2084.18	17.87	3600
2718.78	2075.40	17.65	3600
2722.73	1980.62	17.54	3600
2786.75	1934.59	17.79	3600
2882.57	1932.54	17.80	3600
2884.90	1923.11	18.34	3600

Table B.6: Full Pareto front, 20-client cumulative.

Time (min)	Cost (m.u.)	Gap (%)
3252.98	3838.31	15.59
3593.09	3131.73	49.74
3711.10	2959.69	49.03
3712.99	2905.23	48.14
3723.18	2893.34	49.07

B.2 Independent instances

Table B.7: Full Pareto front, 5-client instance A.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
929.54	1122.12	0.00	24
935.78	1103.44	0.00	35
1035.82	903.98	0.00	48
1043.03	883.91	0.00	34
1049.27	865.24	0.00	156
1149.31	665.78	0.00	378
1294.18	562.32	0.00	231
1296.63	520.23	0.00	202
1298.56	485.39	0.00	186
1310.09	481.49	0.00	103
1318.43	477.16	0.00	58
1410.12	282.02	0.00	38

Table B.8: Full Pareto front, 5-client instance B.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
813.83	962.25	0.00	45
871.95	842.62	0.00	65
934.43	706.14	0.00	102
992.55	586.51	0.00	86
1110.41	551.58	0.00	33
1359.01	529.25	0.00	154
1371.28	519.55	0.00	178
1388.67	474.91	0.00	123
1422.75	415.81	0.00	99
1429.40	399.92	0.00	75
1480.87	296.17	0.00	54

Table B.9: Full Pareto front, 10-client instance A.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2161.61	2487.14	4.95	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2180.80	2486.82	7.81	3600
2277.88	2309.89	11.39	3600
2288.41	2239.81	12.13	3600
2375.46	2206.76	12.63	3600
2405.17	2136.36	11.66	3600
2440.60	2081.88	11.23	3600
2456.90	2077.59	12.78	3600
2502.26	1959.44	13.85	3600
2502.32	1957.73	13.21	3600
2518.56	1955.14	12.98	3600
2542.56	1953.46	13.20	3600
2558.24	1924.23	13.87	3600
2574.48	1921.65	13.21	3600
2659.21	1889.12	13.54	3600
2675.44	1846.90	13.28	3600
2696.05	1821.92	12.59	3600
2710.87	1792.42	12.56	3600
2751.91	1790.13	12.69	3600
2751.97	1788.42	13.33	3600
2772.47	1671.68	7.34	3600
2772.53	1669.97	8.23	3600
2828.38	1638.19	0.00	3179
2828.45	1636.48	0.00	2908

Table B.10: Full Pareto front, 10-client instance B.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2115.85	2356.23	8.53	3600
2214.01	2178.66	13.18	3600
2217.72	2163.85	11.15	3600
2230.70	2114.11	10.55	3600
2238.17	2109.58	11.09	3600
2315.88	1986.29	11.86	3600
2328.87	1936.54	11.13	3600
2332.57	1921.74	10.18	3600
2340.04	1917.21	10.30	3600
2353.03	1867.46	10.52	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2438.21	1739.64	11.34	3600
2454.90	1675.09	10.34	3600
2475.00	1641.71	11.59	3600
2553.06	1497.52	11.23	3600
2576.87	1449.34	11.19	3600
2724.39	1421.98	10.53	3600
2744.49	1388.60	11.87	3600
2748.02	1374.03	11.32	3600
2761.38	1361.16	12.53	3600

Table B.11: Full Pareto front, 15-client instance A.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2200.50	2561.17	7.64	3600
2222.89	2536.83	8.24	3600
2236.45	2487.65	9.04	3600
2280.56	2423.76	9.62	3600
2284.64	2416.99	10.09	3600
2307.31	2342.85	10.42	3600
2320.28	2317.29	10.41	3600
2342.66	2292.95	9.81	3600
2343.25	2269.33	10.12	3600
2356.22	2243.77	8.58	3600
2387.36	2205.44	10.98	3600
2404.41	2173.11	11.19	3600
2429.36	2113.55	13.20	3600
2442.33	2087.99	10.41	3600
2453.45	2070.10	9.59	3600
2457.53	2063.33	11.41	3600
2470.50	2037.78	9.00	3600
2537.49	2035.43	10.23	3600
2548.96	2017.87	11.53	3600
2557.20	1858.37	11.28	3600
2570.17	1832.82	10.69	3600
2637.16	1830.47	10.57	3600
2650.13	1804.91	11.91	3600
2665.80	1798.54	12.32	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
2681.38	1766.06	10.12	3600
2716.55	1692.65	11.32	3600
2732.13	1660.18	11.47	3600
2748.42	1625.90	11.31	3600
2764.00	1593.43	11.09	3600
2777.23	1568.65	10.98	3600
2781.05	1561.01	9.45	3600

Table B.12: Full Pareto front, 15-client instance B.

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
3210.83	3685.05	8.11	3600
3301.70	3487.31	10.59	3600
3332.02	3431.84	12.60	3600
3395.02	3293.25	12.04	3600
3422.89	3234.10	12.50	3600
3449.37	3208.06	11.32	3600
3459.27	3185.00	12.87	3600
3467.29	3173.11	11.89	3600
3516.21	3040.04	12.15	3600
3530.28	3034.52	11.56	3600
3548.08	3033.53	11.98	3600
3558.16	2975.38	12.32	3600
3570.57	2954.85	12.62	3600
3594.53	2926.28	11.47	3600
3614.96	2893.86	11.19	3600
3643.46	2793.20	12.08	3600
3663.88	2760.79	11.36	3600
3678.85	2744.57	11.17	3600
3685.40	2728.54	10.98	3600
3697.81	2708.02	10.79	3600
3705.83	2696.13	11.34	3600
3723.24	2683.57	10.41	3600
3735.65	2663.05	12.22	3600
3742.21	2647.03	10.03	3600
3778.72	2534.47	9.98	3600
3799.15	2502.06	10.74	3600

Time (min)	Cost (m.u.)	Gap (%)	Run time (s)
3833.08	2449.29	9.45	3600

Appendix C

Conference Submission

This appendix reproduces the abstract submitted to Optimization 2026.

Abstract

As the complexity of global supply chains grows, maritime transport, especially container terminals, faces increasing pressure to better integrate sea-side operations with land-side distribution. In this context, planning and dispatching container trucks in hinterland transport is a key challenge, made harder by networks spanning multiple terminals and depots with vehicles and containers dispersed across yards and inland facilities. This work aims to tackle this problem by designing a decision support system based on a Multi-Objective Mixed-Integer Linear Programming to reduce total transportation time and cost. The model determines the best vehicle assignment and task scheduling, accounting for mixed vehicle fleets, driver working time regulations, multiple depots and terminals, time windows for pickups and deliveries, and managing both full and empty containers. Preliminary results show practically interesting trade-offs between these conflicting objectives. Minimizing total transportation time tends to favor the use of conventional fuel-powered trucks, given their greater operational flexibility when compared to electric vehicles. Conversely, when cost and environmental considerations are given more weight, the model increasingly allocates electric vehicles, despite their limitations in terms of range and charging requirements.

Keywords: Global supply chains, Logistics networks, Container terminals, Truck scheduling, Multi-objective optimization

Acknowledgments: This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia IP, under the support UID/50014/2025, and co-financed by Component 5 – Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan, within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026, within project NEXUS, with reference 53.