Robust Optimization of Heavy Goods Electric Vehicle Fleet Planning

CertℰRoute: A Framework for Routing and Charging Scheduling under Time and Energy Consumption Uncertainty

M.B.J. de Goede

Robust Optimization of Heavy Goods Electric Vehicle Fleet Planning

Cert$\mathcal{E}$Route: A Framework for Routing and Charging Scheduling under Time and Energy Consumption Uncertainty

Thesis report

by

M.B.J. de Goede

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on June 27, 2025 at 13:30

Thesis committee:

Chair:	Dr N. Yorke-Smith
Supervisors:	Dr N. Yorke-Smith
	Ir J.L.P. Commandeur
External examiner:	Dr G.R. Chandra Mouli
Place:	Delft
Project Duration:	November, 2024 – June, 2025
Student number:	5073294

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Delft University of Technology

TU Delft Delft University of Technology

Acknowledgements

Standing at the end of my six-year academic journey at TU Delft, I am filled with gratitude for everyone who supported me, directly or indirectly, through this transformative chapter of personal and professional growth. This period has undoubtedly been the most rewarding and fulfilling of my life so far. Thank you for being a part of it.

Hearty thanks go out to my supervising professor, Neil Yorke-Smith, for his continuous support through weekly meetings, insightful feedback, and remarkably prompt responses to my emails.

Additionally, I would like to express my sincere gratitude to my daily supervisor, Joost Commandeur, for offering me the opportunity to carry out this thesis project at Shell as part of the assessed internship program. His deep involvement, constant feedback, and thoughtful guidance, especially in helping me maintain perspective on the bigger picture, were invaluable in keeping me on track.

Kind appreciation is extended to my fellow colleagues at Shell. In particular, I would like to thank Maurits Merks and Joris Veen for their warm welcome and pragmatical advice from day one, and Rob Doel for offering a fresh perspective on my research results.

I am also deeply thankful to my family for their unconditional support throughout my studies and thesis work. A special thanks to my sister Sophia, who introduced me to the focused work environment of the university library, where the majority of this thesis was written.

Lastly, I want to thank my roommates and other friends for being patient listeners, enduring my daily stories and struggles, even when the topics ranged far beyond what could reasonably be considered interesting to someone not directly involved in the project.

Matthijs de Goede
Delft, June 2025

Abstract

The shift towards zero-emission transport has driven rapid adoption of Heavy Goods Electric Vehicles (HGEVs) across Europe. This trend is also evident in the Netherlands, where their numbers have grown exponentially over the past six years. The integration of HGEVs into fleet operations introduces new challenges for fleet planning due to limited battery ranges and significant operational uncertainties in both the time and energy domains.

To address these challenges, this thesis introduces Cert$\mathcal{E}$Route, an Adaptive Robust Optimization (ARO) framework that allows for joint optimization of routing and charging scheduling under time and energy consumption uncertainty. It incorporates both depot and en-route charging while accounting for charger availability and other realistic operational constraints.

Time-related uncertainties in service, waiting, travel, and charging durations are modelled using uncertainty sets, which require minimal assumptions about the underlying probability distributions. Energy consumption is modelled as a function of time-domain uncertainty, with environmental and vehicle-specific energy uncertainty factors accounted for through Monte Carlo Simulation (MCS). The adaptive design of the framework supports a two-stage decision process where routing and charger visits are planned in advance to hedge against worst-case scenarios, while charging amount and timing decisions remain flexible during route execution to avoid overly conservative and costly solutions.

To ensure computational tractability, the framework employs a Column and Constraint Generation (CCG) approximation method enhanced by a novel One-Step Look-ahead Pessimization (OSLP) algorithm, which selectively integrates only provably infeasible scenarios into the optimization problem. Despite its theoretical vulnerability to premature convergence, this algorithm empirically produces highly robust solutions. To improve scalability to larger instances, a multi-scenario Adaptive Large Neighbourhood Search (ALNS) metaheuristic is developed, integrating charging and timing decisions into neighbourhood generation to enable more informed solution exploration.

The framework is evaluated using representative yet synthetic European HGEV planning scenarios. The results highlight the scalability of the proposed solution methods and demonstrate that the resulting plans are highly robust under operational uncertainty. As a result, this work offers a practical foundation for integrated fleet and energy management systems and supports Shell eMobility's strategic goal of enabling sustainable, cost-efficient transport through offering intelligent charging solutions.

Contents

1 Introduction

1.1. Motivation

The transport of people and goods contributed approximately one-fifth of global carbon dioxide emissions in 2020, with road freight transport accounting for a third of this share [1]. The push towards zero-emission mobility has accelerated the adoption of Heavy Goods Electric Vehicles (HGEVs) as replacements for diesel trucks in the European transport sector [2]. This trend is also evident in the Netherlands, where the total number of HGEVs has grown exponentially over the past six years, showing a tenfold increase in annual sales between 2022 and 2023 [3].

The integration of HGEVs into fleet operations introduces new challenges for fleet planning, primarily due to their limited battery range and the presence of operational uncertainties in both the time and energy domains. Energy consumption is influenced by external uncertainty factors such as weather conditions, road quality, and traffic congestion, as well as internal factors like the vehicle's drive train efficiency and driver behavior, all of which significantly affect the vehicle's effective range [4, 5, 6]. Time-related uncertainties further complicate operations. These include varying charging durations stemming from both vehicle- and infrastructure-related uncertainties, fluctuating service times at customer locations caused by unloading delays and waiting periods, and uncertain travel times due to traffic congestion [7, 8].

Due to their signficantly greater mass and associated energy demands, HGEVs are equipped with larger batteries than electric passenger cars or delivery vans, leading to extended charging durations. As a result, charging sessions must be planned more carefully. Currently, fleet operators rely heavily on overnight charging to fully replenish vehicle batteries, ensuring that planned routes can be completed without the need for additional charging during the day. On-the-go charging during route execution is used only incidentally [9]. This strong reliance on overnight charging limits route flexibility and can lead to congestion at depot chargers. Moreover, charging operations are frequently handled separately from route planning, overlooking opportunities for improved operational efficiency through integrated co-optimization.

Shell eMobility provides charging solutions to its customers both in the public and private space. To better service its customers in the provisioning of cost-efficient energy and charging solutions for HGEV fleets, Shell aims to better understand the operational planning behind these fleets. By understanding the customers better it is possible to collaborate more closely, unlocking potential for demand side flexibility, which in turn offers cost reduction potential, supporting the shared goal of zero-emission transport. Additionally, Shell seeks to gain insights into the fleet routing and charging patterns of their customers to make informed decisions about the optimal placement and sizing of new public charging infrastructure, prioritizing locations that align with customer routes to maximize charger utilization. In this broader context, this thesis aims to establish a foundation for integrated fleet and energy planning by addressing the routing and charging scheduling problem from a fleet operations perspective.

1.2. Objectives and Research Questions

Concretely, the primary goal of this thesis is to develop and evaluate an optimization framework for integrated routing and charging scheduling that emphasizes cost-efficiency while ensuring robustness against uncertainties in time and energy consumption. Moreover, the solution methods within the framework must be computationally efficient so that they can be used for day-ahead fleet operation planning. Aligned with these objectives, we formulate the following main research question:

Main Research Question

How can the cost-efficiency of Heavy Goods Electric Vehicle (HGEV) fleet operations in road freight transport be optimized through integrated vehicle routing and charging scheduling, while ensuring robustness under time and energy consumption uncertainty, and maintaining computational tractability?

Research Sub-questions

To systematically address the main research question, the study is structured around four research sub-questions, each targeting a critical aspect of the thesis:

Research Sub-question 1

What is the current state of research on optimization under uncertainty in HGEV fleet planning, and what gaps can be identified?

Research Sub-question 2

How can uncertainties in time and energy consumption be formally represented and incorporated into an optimization model for HGEV charging scheduling and routing?

Research Sub-question 3

Which solution methods can be employed to robustly solve the formulated optimization problem, while ensuring computational tractability?

Research Sub-question 4

How do the proposed solution methods perform when applied to realistic HGEV fleet planning scenarios, in terms of robustness, cost-efficiency, scalability, and adaptability?

1.3. Contributions

In response to our research questions, we present CERT$\mathcal{E}$ROUTE, an optimization framework grounded in Robust Optimization (RO) principles. It incorporates a comprehensive suite of solvers and uncertainty sets that support deterministic, robust, and adaptive robust optimization. Designed to effectively solve the HGEV routing and charging scheduling problem under time and energy consumption uncertainty, its name reflects its core objective: ensuring operational certainty of HGEV routes in the face of uncertainty.

Concretely, our contributions include the following:

1. **Realistic modelling of HGEV fleet operations:** We introduce the Heavy Goods Electric Vehicle Fleet Planning Problem under Time and Energy Consumption Uncertainty (HGEV-FPP-TECU), a new variant of the Electric Vehicle Routing Problem (EVRP) that formalizes the uncertainty in time and energy consumption and explicitly incorporates coordinated charging scheduling at the depot.
2. **Joint consideration of time and energy uncertainties:** To the best of our knowledge, we are the first to jointly address uncertainties in both time and energy consumption through an Adaptive Robust Optimization (ARO) model for an EVRP. To capture the interdependency between the two domains, we model energy consumption as a function of the uncertainty set in the time domain and perform Monte Carlo Simulations (MCS) to estimate its parameters.
3. **Novel pessimization algorithm for robust optimization:** To address the computational complexity of the ARO model, we propose the ONE-STEP LOOK-AHEAD PESSIMIZATION (OSLP) procedure, a novel pessimization algorithm for the Column and Constraint Generation (CCG) framework. This algorithm is designed to quickly identify provably infeasible scenarios, and despite its theoretical vulnerability to premature convergence, it empirically produces highly robust solutions.
4. **Custom ALNS metaheuristic to ensure scalability:** To maintain computational tractability on large-scale HGEV-FPP-TECU instances, we implement a tailored multi-scenario Adaptive Large Neighborhood Search (ALNS) metaheuristic for the Restricted Master Problem (RMP) in CCG. This approach explicitly models charging and timing decisions within its neighbourhood generation mechanisms, thereby facilitating more informed exploration of the solution space. The metaheuristic was found to significantly enhance computational performance over exact methods.
5. **Generation of realistic benchmark instances:** We present a set of realistic HGEV-FPP-TECU benchmark instances that are grounded in real-world data and expert knowledge to capture the operational conditions of European HGEV fleets. These instances serve as a testbed for evaluating the proposed methodology.
6. **Comprehensive empirical evaluation:** We conduct an extensive evaluation of the proposed methodology across the benchmark instances, offering insights into the robustness, cost-efficiency, scalability, and adaptability of the generated solutions.

1.4. Thesis Outline

The remainder of this thesis is outlined as follows:

- Chapter 2 first provides the necessary background by introducing foundational concepts relevant to this research. It provides an overview of the electric vehicle routing problem and common solution strategies. Additionally, it discusses energy consumption modelling and established techniques for handling uncertainty in optimization, with a focus on the robust optimization framework adopted in this work. The chapter then provides a review of related literature and identifies the research gaps addressed by this work.
- Chapter 3 details our methodological framework. It begins with a formal definition of the HGEV-FPP-TECU problem and a justification for the key modelling assumptions. The chapter then presents the proposed adaptive robust optimization model, including its constraints, uncertainty sets, and the proposed energy consumption function. The solution methodology is subsequently described, including the column and constraint generation approach, the novel OSLP pessimization algorithm, and the metaheuristic solution method. The chapter concludes with the synthesis of these components into a suite of solvers that constitute the Cert$\mathcal{E}$Route framework.
- Chapter 4 outlines the experimental design. It describes the process of generating realistic HGEV-FPP-TECU instances, elaborates on the energy consumption modelling, and provides implementation and configuration details necessary for reproducibility.
- Chapter 5 presents the results of the computational experiments. This includes an evaluation of the deterministic solution approach, a sensitivity analysis to assess the impact of uncertain parameters, a comparative analysis between adaptive and static robust optimization, and a robustness assessment. The performance of the proposed metaheuristic approach is also evaluated.
- Chapter 6 concludes the thesis by addressing the research questions, offering practical guidelines for implementing the Cert$\mathcal{E}$Route framework, discussing the limitations of the current work, and suggesting directions for future work.

2

Background and Literature Review

This chapter provides relevant background information and reviews the existing literature, addressing the research sub-question:

> **Research Sub-question 1**
>
> What is the current state of research on optimization under uncertainty in HGEV fleet planning, and what gaps can be identified?

Specifically, Section 2.1, introduces the vehicle routing problem, a theoretical problem that underlies fleet planning optimization. Section 2.2 details how this problem can be adapted to consider electric vehicles. Section 2.3 describes various solution methods for the problem. Section 2.4 describes how the energy consumption of electric vehicles can be modelled. Section 2.5 provides a background on methods for optimization under uncertainty. Section 2.6 then describes how such methods are used in practice to deal with uncertainty in fleet operations. Finally, Section 2.7 gives an overview of closely related work from the literature, identifying the research gaps addressed with this work. Readers already familiar with the electric vehicle routing problem and its solution methods may proceed directly to Section 2.5.

2.1. An Introduction to the The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem in logistics. It involves determining the most efficient set of delivery routes for a fleet of vehicles across multiple locations within a transport network, while satisfying specific constraints such as meeting the demand at each location. Efficiency is typically measured using an objective function that minimizes total traversal distance or cost. The VRP generalizes the Travelling Salesman Problem (TSP), extending it from a single salesperson visiting all locations to a fleet of vehicles collectively serving all locations. Dantzig and Ramser [10] originally introduced the VRP as the Truck Dispatching Problem, in which a homogeneous fleet of gasoline delivery trucks was tasked to serve the gasoline demands of a set of chargers, starting from a central bulk terminal and minimizing the travelled distance. Clarke and Wright [11] generalized further to a linear optimization problem and proposed an iterative heuristic to solve the problem. Since the survey by Christofides [12], the problem is known as the Vehicle Routing Problem or Capacitated Vehicle Routing Problem (CVRP), referring to the limited load capacity of the vehicles.

The VRP is classified as a $\mathcal{NP}$-hard problem [13], meaning no known algorithm can solve all instances to optimality within polynomial time as the problem size grows. Today, the VRP knows countless variations and is one of the most extensively studied problems in operations research [14]. Notable research directions include accounting for heterogeneous vehicle characteristics [15], time constraints [16], the introduction of multiple depots [17], backhauls [18], dynamic routing [19] and rich combinations of the previously mentioned [20]. Comprehensive reviews of more VRP variants can be found in the works by Konstantakopoulos, Gayialis, and Kechagias [14] and Braekers, Ramaekers, and Van Nieuwenhuyse [21].

A recent line of research builds upon the work of Erdoğan and Miller-Hooks [22], concentrating on green vehicles powered by alternative fuels and incorporating the option of refueling at chargers along the route to extend their often limited driving ranges. This thesis focuses on a specific variation that involves electric vehicles, known as the Electric Vehicle Routing Problem (EVRP). In the next section, we outline the most elementary EVRP considered in the literature and discuss several extensions leading up to our Heavy Goods Electric Vehicle Fleet Planning Problem under Time and Energy Consumption Uncertainty (HGEV-FPP-TECU).

2.2. Electrifying the Vehicle Routing Problem

While Battery Electric Vehicles (BEVs) have the potential to reduce transport costs and environmental impact compared to traditional fossil-fuel vehicles, they face significant new challenges in road freight transport such as limited driving range, long charging durations, and a lack of widespread charging infrastructure. These challenges require meticulous planning of charging operations to ensure that BEVs can complete their routes without depleting their battery or taking extensive detours [23, 24, 25]. The Electric Vehicle Routing Problem, introduced by Conrad and Figliozzi [26], extends the CVRP by incorporating battery constraints and allowing electric vehicles to recharge at designated customer locations before proceeding with their routes. A more comprehensive version was later introduced by Erdoğan and Miller-Hooks [22], enabling vehicles to visit separate chargers situated anywhere. This subsequent formulation has become the starting point for numerous later studies [27, 28].

2.2.1. Elementary Formulation

Kucukoglu, Dewil, and Cattrysse [27] conducted an extensive and systematic review of 136 papers from the extensive body of literature surrounding the EVRP. They clearly discuss its various extensions, providing a mathematical formulation that captures the core aspects of the problem. Consolidating from their review, we establish that in its most elementary form, the EVRP can be formulated based on the following fundamental assumptions:

Assumptions

1. The fleet of delivery vehicles contains identical electric vehicles with fixed load and battery capacity.
2. Each vehicle performs a single tour starting and ending at a central depot, or remains at the depot.
3. Each customer to be served has a known demand and is served by exactly one vehicle.
4. Each vehicle can visit a charger for recharging between any two customers.
5. A vehicle's initial State of Charge (SoC) is full.
6. Each charger can be visited by more than one vehicle.
7. A vehicle's battery is always fully charged when visiting a charger (full-charging policy).
8. The battery level of a vehicle must always be between 0 and its battery capacity.
9. The load of a vehicle must always be between 0 and its load capacity.
10. The energy consumption and cost for traversing road connections between the depot, customers, and chargers are known.

Mathematical Definition

We define the Elementary Electric Vehicle Routing Problem (EEVRP) accordingly:

Definition 1 – Elementary EVRP (EEVRP)

Consider a transport network modelled as a complete directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{0\} \cup \mathcal{C} \cup \mathcal{F}' \cup \{N+1\}$ represents the set of vertices, with $\mathcal{C} := \{1, 2, \dots, N\}$ representing the set of N customers i having delivery demand q_i. Let $\mathcal{F}' := \{N+2, \dots, N+1+NF, N+2+NF, \dots, N+1+n_{visits}+NF\}$ represent a set of dummy vertices to allow n_{visits} to each of the NF chargers f in set $\mathcal{F} := \{N+2, \dots, N+1+NF\}$. Let vertices 0 and $N+1$ represent the exit and entrance of the central depot, respectively. For convenience, let $\mathcal{V}' := \mathcal{C} \cup \mathcal{F}'$, $\mathcal{V}'_0 := \mathcal{V}' \cup \{0\}$, and $\mathcal{V}'_{N+1} := \mathcal{V}' \cup \{N+1\}$. Let the set of edges $\mathcal{E} := \{(i,j) \mid i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1},\ i \neq j\}$ contain all possible road connections (i,j), each associated with a known energy consumption h_{ij} and cost c_{ij}. Let $\mathcal{K}$ represent a fleet of homogeneous electric vehicles k with known individual load capacity C and battery capacity Q. Furthermore, let y^k denote the variable battery state of charge of vehicle k, which declines by h_{ij} for every traversed edge $(i,j) \in \mathcal{E}$ and is set to Q according to the full-charging policy when departing from depot 0 or any of the chargers $f \in \mathcal{F}'$, so that y_i^k denotes the battery SoC upon arrival at destination $i \in \mathcal{V}'$. Let u^k denote the variable remaining load quantity of vehicle k. The EEVRP is then defined as constructing a set of routes $\mathcal{R} := \{R_k \mid R_k \subset \mathcal{E}, k \in \mathcal{K}\}$ such that every R_k is a path starting at vertex 0 and ending at vertex $N+1$, every customer $i \in \mathcal{C}$ is part of exactly one route R_k, vehicles k do not run out of battery ($y^k \geq 0$) nor exceed their load capacity ($u^k \leq C$) during R_k, and the total cost associated with $\mathcal{R}$ is minimized.

Mixed Integer Linear Program Formulation

Based on the above mathematical definition, a Mixed Integer Linear Programming (MILP) [29] formulation for the EEVRP can be constructed, following standard practices in VRP literature. An MILP is an optimization problem where both the constraints and the objective function are linear, with some decision variables restricted to integer values. Solving an MILP exactly results in an optimal solution to the problem. The formal definition of an MILP is as follows:

Definition 2 – Mixed Integer Linear Progamming (MILP)

Let a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, vectors $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$, and a subset $I \subseteq \{1, 2, \ldots, n\}$ be given. An MILP seeks to find scalar ζ^*, known as the objective value, such that

$$\zeta^* := \min\{\boldsymbol{c}^\top \boldsymbol{x} \ : \ \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x}_i \in \mathbb{Z},\ \forall i \in I\}.$$

The set $S := \{\boldsymbol{x} \ : \ \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x}_i \in \mathbb{Z},\ \forall i \in I\}$ is called the feasible set.
A solution $\boldsymbol{x}^* \in S$ is considered optimal if $\boldsymbol{c}^\top \boldsymbol{x^*} = \zeta^*$.

A typical MILP modelling approach for VRPs involves introducing a binary decision variable x_{ij}^k for all vertices $i \in \mathcal{V}_0'$, $j \in \mathcal{V}_{N+1}'$, and each vehicle $k \in \mathcal{K}$. Here, $x_{ij}^k = 1$ indicates that edge $(i, j) \in \mathcal{E}$ is included in route R_k, and $x_{ij}^k = 0$ otherwise. In addition, auxiliary continuous decision variables y_i^k and u_i^k are often introduced for each vehicle $k \in \mathcal{K}$ and each location $i \in \mathcal{V}_{N+1}'$. These variables represent the battery SoC and the remaining load quantity at location i upon arrival, respectively. A corresponding MILP formulation for the EEVRP is detailed in Table 2.1 and Model 1 based on the work of Schneider, Stenger, and Goeke [30], Keskin and Çatay [31], Schiffer and Walther [32], and Kucukoglu, Dewil, and Cattrysse [27].

Sets

$0,\ N+1$	Exit and entrance vertices of the central depot
$\mathcal{C}$	Set of customers vertices $\mathcal{C} := \{1, 2, \ldots, N\}$
$\mathcal{F}$	Set of charger vertices $\mathcal{F} : \{N+2, \ldots, N+1+NF\}$
$\mathcal{F}'$	Set of dummy charger vertices $\mathcal{F}' := \{N+2, \ldots, N+1+NF, N+2+NF, \ldots, N+1+n_{visits}+NF\}$ to allow n_{visits} visits to every $f \in \mathcal{F}$
$\mathcal{V}'$	Set of customer and charger vertices $\mathcal{V}' := \mathcal{C} \cup \mathcal{F}'$
$\mathcal{V}_0'$	Set of source location vertices $\mathcal{V}_0' := \mathcal{V}' \cup \{0\}$
$\mathcal{V}_{N+1}'$	Set of destination location vertices $\mathcal{V}_{N+1}' := \mathcal{V}' \cup \{N+1\}$
$\mathcal{V}$	Set of all vertices $\mathcal{V} := \{0\} \cup \mathcal{C} \cup \mathcal{F}' \cup \{N+1\}$
$\mathcal{E}$	Set of edges $\mathcal{E} := \{(i, j) \mid i \in \mathcal{V}_0',\ j \in \mathcal{V}_{N+1}',\ i \neq j\}$
$\mathcal{K}$	Set of homogeneous electric vehicles
$\mathcal{R}$	Set of vehicle routes $\mathcal{R} := \{R_k \mid R_k \subset \mathcal{E}, k \in \mathcal{K}\}$

Decision Variables

x_{ij}^k	Binary decision variable indicating whether edge $(i, j) \in \mathcal{E}$ is included in the route of vehicle $k \in \mathcal{K}$
y_i^k	Continuous decision variable tracking the battery SoC of vehicle $k \in \mathcal{K}$ upon arrival at destination $i \in \mathcal{V}$ [in kWh]

u_i^k	Continuous decision variable tracking the remaining load quantity of vehicle $k \in \mathcal{K}$ upon arrival at destination $i \in \mathcal{V}$ [in units]

Parameters	
q_i	Delivery demand at customer $i \in \mathcal{C}$ [in units]
h_{ij}	Energy consumption over edge $(i,j) \in \mathcal{E}$ [in kWh]
c_{ij}	Traversal cost over edge $(i,j) \in \mathcal{E}$ [in €]
C	Individual vehicle load capacity [in units]
Q	Individual vehicle battery capacity [in kWh]

Table 2.1: Notation overview for the EEVRP

Model 1 – MILP Formulation for the EEVRP

$$c^*_{total} = \min \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{V}'_0} \sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} c_{ij} x_{ij}^k \tag{2.1}$$

subject to (s.t.):

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} x_{ij}^k = 1 \quad \forall i \in \mathcal{C}, \tag{2.2}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} x_{ij}^k \leq 1 \quad \forall i \in \mathcal{F}', \tag{2.3}$$

$$\sum_{j\in\mathcal{V}'_{N+1}} x_{0j}^k \leq 1 \quad \forall k \in \mathcal{K}, \tag{2.4}$$

$$\sum_{i\in\mathcal{V}'_0,\, i\neq j} x_{ij}^k = \sum_{i\in\mathcal{V}'_{N+1},\, i\neq j} x_{ji}^k \quad \forall j \in \mathcal{V}',\, \forall k \in \mathcal{K}, \tag{2.5}$$

$$u_j^k \leq u_i^k - q_i x_{ij}^k + C(1 - x_{ij}^k) \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.6}$$

$$u_0^k \leq C \quad \forall k \in \mathcal{K}, \tag{2.7}$$

$$y_j^k \leq y_i^k - h_{ij} x_{ij}^k + Q(1 - x_{ij}^k) \quad \forall i \in \mathcal{V},\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.8}$$

$$y_j^k \leq Q - h_{ij} x_{ij}^k \quad \forall i \in \mathcal{F}' \cup \{0\},\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.9}$$

$$y_0^k \leq Q \quad \forall k \in \mathcal{K}, \tag{2.10}$$

$$y_i^k \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K}, \tag{2.11}$$

$$u_i^k \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K}, \tag{2.12}$$

$$x_{ij}^k \in \mathbb{B} \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, i \neq j. \tag{2.13}$$

Here, the objective function in Eq. (2.1) aims to minimize the total traversal cost of the routes. Note that in this formulation, recharging operations do not incur additional costs, as the emphasis lies on feasibility. The constraints in Eq. (2.2) ensure that every customer is visited once. The constraints in Eq. (2.3) ensure that each dummy charger can be visited at most once. The constraints Eq. (2.4) make sure that every electric vehicle can be used in at most one route plan. The constraints in Eq. (2.5) are flow conservation constraints ensuring that the total number of incoming edges is equal to the total number of outgoing edges at customer and charger vertices, providing continuity in the routes. The constraints in Eq. (2.6) are the so called Miller-Tucker-Zemlin (MTZ) [33] constraints that, together with the constraints in Eq. (2.7), ensure the load capacity of the vehicles is never exceeded and unconnected subtours that do not start and end at the depot are eliminated. The constraints in Eq. (2.8) to Eq. (2.10) track the battery SoC of the electric vehicles and their battery state after the recharging operations at the chargers following a full charging policy. The constraints in Eq. (2.11) to Eq. (2.13) ensure that the decision variables are restricted to the appropriate domains.

Alternative formulations
The MILP formulation presented in Model 1 is based on the vehicle flow model formulation for VRPs, ensuring that each customer location is visited by exactly one vehicle [34]. Although, this model is very intuitive and wide-spread, there are alternative formulations, such as the commodity flow formulation [35, 36], which constrains the flow of commodities (goods or services) through the network, and the set partitioning formulation [37, 34], which focuses on selecting subsets of feasible routes to cover all customers while minimizing cost, assuming individual route feasibility. These alternatives have stronger MILP relaxations than the vehicle flow model, allowing for more efficient solutions using branch-and-bound methods [38, 36]. However, the set partitioning formulation can lead to an exponential number of feasible routes, requiring complex techniques like column generation and branch-and-price [34]. More details on these solution methods are provided in Subsection 2.3.1.

The sub-tour elimination constraints from Miller, Tucker, and Zemlin [33] in Eq. (2.6) can also be formulated differently. The capacity-cut constraints presented by Toth and Vigo [36] provide an alternative way to simultaneously impose solution connectivity and vehicle capacity requirements. Using branch-and-cut methods, this formulation yields a stronger bound than the MTZ constraints. However, the number of associated constraints grows exponentially rather than quadratically with the number of customers [39].

Constraint Programming
While not as commonly featured in VRP literature, Constraint Programming (CP) is another powerful approach for modelling and solving combinatorial optimization problems. In contrast to MILP, CP does not require constraints and objective functions to be linear, enabling the use of more expressive constraints and specialized algorithms to tackle the corresponding constraint satisfaction subproblems. CP techniques typically prioritize feasibility over optimization, making them particularly well-suited for discrete rather than continuous problems. Similar to MILP, CP faces scalability challenges and is often combined with local search heuristics to find solutions quicker [40]. Notable applications of CP in VRP include the works of Shaw [41] and Backer et al. [42]. The application of CP to solve the EVRP remained largely unexplored, such that only the work by Booth and Beck [43] could be identified.

2.2.2. Time Windows

A popular extension to the EEVRP is the Electrical Vehicle Routing Problem with Time Windows (EVRPTW), as introduced by Schneider, Stenger, and Goeke [30]. This variant is the electrified version of the standard Vehicle Routing Problem with Time Windows (VRPTW), where customers are required to be served within a designated time window. Each customer has a service duration that represents the time required for unloading. At the depot, this corresponds to the time needed to load the vehicle. For simplicity, service durations are generally considered independent of the demand. Furthermore, chargers have specific time windows during which they are available for visits.

Keskin, Laporte, and Çatay [44] introduced the EVRPTW with soft time windows. Time windows are classified as soft when they are flexible and can be exceeded at the cost of a penalty. Such a penalty is often modelled using slack variables that represent the degree of constraint violation. These can then be added to the objective function under minimization to minimize violations. Conversely, time windows are considered hard when they are strictly enforced. That is, vehicles must wait if they arrive earlier than the start of the time window and are prohibited from arriving late [45]. For simplicity, we focus on the hard time window formulation, which can be defined as follows:

Definition 3 – EVRP with Time Windows (EVRPTW)

Consider the EEVRP from Definition 1, with the following added characteristics. In addition to known traversal cost c_{ij} and energy consumption h_{ij}, let t_{ij} denote a known traversal duration associated with every edge $(i, j) \in \mathcal{E}$. Furthermore, let e_i and l_i represent the earliest and latest possible service start times for every vertex $i \in \mathcal{V}$, respectively. Let ν_i denote the corresponding loading/unloading duration for every location $i \in \mathcal{C} \cup \{0\}$. Let g_f be a known and constant recharging rate for every charger $f \in \mathcal{F}'$, so that the full-charging operation has a duration of $g_f(Q - y_f^k)$ for every $k \in \mathcal{K}$. Lastly, let τ_i be the variable service start time for every location $i \in \mathcal{V}$. The EVPRTW is defined as constructing a set of routes $\mathcal{R}$ with the same constraints as in the EEVRP and the added constraint that service start times τ_i should be between e_i and l_i for every location $i \in \mathcal{V}$, taking the traversal, service and charging durations within $\mathcal{R}$ into account.

A corresponding MILP formulation is detailed in Table 2.2 and Model 2 [30, 27].

Decision Variables	
τ_i	Continuous decision variable indicating the service start time at location $i \in \mathcal{V}$ [in minutes]

Parameters	
t_{ij}	Traversal duration associated with edge $(i, j) \in \mathcal{E}$ [in minutes]
e_i	Earliest possible service start time for location $i \in \mathcal{V}$ [in minutes]
l_i	Latest possible service start time for location $i \in \mathcal{V}$ [in minutes]
ν_i	Loading/unloading duration for location $i \in \mathcal{C} \cup \{0\}$ [in minutes]
g_f	Inverse recharging rate at charger $f \in \mathcal{F}'$ [in minutes/kWh]

Table 2.2: Notation overview for the EVRPTW

Model 2 – MILP Formulation for the EVRPTW

$$c^*_{total} = \min \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{V}'_0} \sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} c_{ij}x^k_{ij} \qquad (Eq.\ (2.1))$$

s.t:

$$Eq.\ (2.2) - Eq.\ (2.13),$$

$$\tau_i + (t_{ij} + \nu_i) \sum_{k\in\mathcal{K}} x^k_{ij} \leq \tau_j + l_0(1 - \sum_{k\in\mathcal{K}} x^k_{ij}) \quad \forall i \in \mathcal{C} \cup \{0\},\ \forall j \in \mathcal{V}'_{N+1},\ i \neq j, \tag{2.14}$$

$$\tau_f + t_{fj}x^k_{fj} + g_f(Q - y^k_f) \leq \tau_j + (l_0 + g_f Q)(1 - x^k_{fj}) \quad \forall f \in \mathcal{F}',\ \forall j \in \mathcal{V}'_{N+1},\ \forall k \in \mathcal{K},\ f \neq j, \tag{2.15}$$

$$e_i \leq \tau_i \leq l_i \quad \forall i \in \mathcal{V}, \tag{2.16}$$

$$\tau_i \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V}. \tag{2.17}$$

Here, the constraints in Eq. (2.14) guarantee time feasibility for edges leaving customers and the depot based on the traversal and service durations, while the constraints in Eq. (2.15) do the same for edges leaving chargers, based on the charging duration associated with a full-charging policy. The constraints in Eq. (2.16) enforce the time windows of the locations, whereas the constraints in Eq. (2.17) ensure that the service start time variables are restricted to take real values.

2.2.3. Heterogeneous Fleets

A homogeneous fleet consists of identical vehicles, whereas a heterogeneous or mixed fleet includes different types of vehicles that vary in aspects such as propulsion technology, load capacity, operating cost, and/or environmental impact. In practice, most transport companies that operate HGEVs use mixed fleets comprising both electric and diesel trucks. When focusing on electric fleets, heterogeneity may arise from variations in charging technology, battery capacity, and energy consumption between truck manufacturers and/or models. Choosing the wrong type of vehicle for a specific route can lead to increased traversal costs and may result in capacity constraint violations [46]. On the other hand, the overall transport costs or energy consumption can be optimized by assigning the most suitable vehicle to each route [27]. Hence, an extension to the EEVRP was proposed, relaxing Assumption 1 to accommodate heterogeneous fleets. The works of Jiang et al. [46], Hiermann et al. [47], and Goeke and Schneider [48] build on the EVRPTW, introducing the Mixed Fleet Electric Vehicle Routing Problem with Time Windows (MF-EVPRTW). We consider a variant in which the load capacity, battery capacity, energy consumption, and battery charging rates are specified for each vehicle type, resulting in the following definition:

Definition 4 – Mixed Fleet EVRPTW (MF-EVRPTW)

Consider the EVRPTW from Definition 3, with the homogeneous load capacity C, battery capacity Q, energy consumption h_{ij}, and battery charging rate g_f parameters replaced by the respective heterogeneous parameters Q_k, C_k, h^k_{ij}, and g^k_f for every vehicle $k \in \mathcal{K}$, charger $f \in \mathcal{F}'$ and edge $(i, j) \in \mathcal{E}$. The constraints are adapted accordingly and the definition of the MF-EVRPTW further remains the same as that of the EVRPTW.

A corresponding MILP formulation is detailed in Table 2.3 and Model 3 [47, 27].

Parameters	
Q_k	Individual vehicle battery capacity of vehicle $k \in \mathcal{K}$ [in kWh]
C_k	Individual vehicle load capacity of vehicle $k \in \mathcal{K}$ [in units]
h^k_{ij}	Energy consumption of vehicle $k \in \mathcal{K}$ over edge $(i, j) \in \mathcal{E}$ [in kWh]
g^k_f	Inverse recharging rate for vehicle $k \in \mathcal{K}$ at charger $f \in \mathcal{F}'$ [in minutes/kWh]

Table 2.3: Notation overview for the MF-EVRPTW

Model 3 – MILP Formulation for the MF-EVRPTW

$$c^*_{total} = \min \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{V}'_0} \sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} c_{ij}x^k_{ij} \qquad (Eq.~(2.1))$$

s.t:

$$Eq.~(2.2) - Eq.~(2.5),$$

$$u^k_j \leq u^k_i - q_i x^k_{ij} + C_k(1 - x^k_{ij}) \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.18}$$

$$u^k_0 \leq C_k \quad \forall k \in \mathcal{K}, \tag{2.19}$$

$$y^k_j \leq y^k_i - h^k_{ij}x^k_{ij} + Q_k(1 - x^k_{ij}) \quad \forall i \in \mathcal{V},\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.20}$$

$$y^k_j \leq Q_k - h^k_{ij}x^k_{ij} \quad \forall i \in \mathcal{F}' \cup \{0\},\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j, \tag{2.21}$$

$$y^k_0 \leq Q_k \quad \forall k \in \mathcal{K}, \tag{2.22}$$

$$Eq.~(2.11) - Eq.~(2.14),$$

$$\tau_f + t_{fj}x^k_{fj} + g^k_f(Q_k - y^k_f) \leq \tau_j + (l_0 + g^k_f Q_k)(1 - x^k_{fj}) \quad \forall f \in \mathcal{F}',\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, f \neq j, \tag{2.23}$$

$$Eq.~(2.16) - Eq.~(2.17).$$

Here, the constraints in Eq. (2.6) to Eq. (2.10) are replaced by the constraints in Eq. (2.18) to Eq. (2.21) and those in Eq. (2.15) by those in Eq. (2.22) to account for the heterogeneous vehicle parameters.

2.2.4. Partial Recharging

Thus far, we have assumed that a vehicle is always fully recharged whenever it visits a charger (Assumption 7). However, this is not always necessary to complete the route successfully. Partial recharging can help lower both costs and duration, reducing the likelihood of time window violations [31]. To address this, studies such as those by Bruglieri et al. [49], Keskin and Çatay [31], Schiffer and Walther [32], and Desaulniers et al. [50] relax Assumption 7 by incorporating partial recharging strategies. In essence, this involves introducing a new decision variable to monitor the battery's state of charge upon departing a charger. Formally, we define the Mixed Fleet Electric Vehicle Routing Problem with Time Windows and Partial Recharging (MF-EVRPTW-PR) as follows:

Definition 5 – MF-EVRPTW with Partial Recharging (MF-EVRPTW-PR)

Consider the MF-EVRPTW from Definition 4. Let $y_f^k \leq Y_f^k \leq Q_k$ be the variable SoC on departure from charger $f \in \mathcal{F}' \cup \{0\}$ for every vehicle $k \in \mathcal{K}$. The constraints are adapted accordingly and the definition of the MF-EVRPTW-PR further remains the same as that of the MF-EVRPTW.

A corresponding MILP formulation is detailed in Table 2.4 and Model 4 [31, 27].

Decision Variables	
Y_f^k	Continuous decision variable indicating the SoC of vehicle $k \in \mathcal{K}$ on departure from charger $f \in \mathcal{F}'$ [in kWh]

Table 2.4: Notation overview for the MF-EVRPTW-PR

Model 4 – MILP Formulation for the MF-EVRPTW-PR

$$c^*_{total} = \min \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}'_0} \sum_{j \in \mathcal{V}'_{N+1},\, j \neq i} c_{ij} x_{ij}^k \qquad (Eq.\ (2.1))$$

s.t:

$$Eq.\ (2.2) - Eq.\ (2.5),$$

$$Eq.\ (2.18) - Eq.\ (2.20),$$

$$y_j^k \leq Y_f^k - h_{fj}^k x_{fj}^k + Q_k(1 - x_{fj}^k) \quad \forall f \in \mathcal{F}' \cup \{0\},\ \forall j \in \mathcal{V}'_{N+1},\ \forall k \in \mathcal{K}, \tag{2.24}$$

$$y_f^k \leq Y_f^k \leq Q_k \quad \forall f \in \mathcal{F}' \cup \{0\},\ \forall k \in \mathcal{K}, \tag{2.25}$$

$$Eq.\ (2.11) - Eq.\ (2.14),$$

$$\tau_f + t_{fj} x_{fj}^k + g_f^k (Y_f^k - y_f^k) \leq \tau_j + (l_0 + g_f^k Q_k)(1 - x_{fj}^k) \quad \forall f \in \mathcal{F}',\ \forall j \in \mathcal{V}'_{N+1},\ \forall k \in \mathcal{K},\ f \neq j, \tag{2.26}$$

$$Eq.\ (2.16) - Eq.\ (2.17),$$

$$Y_f^k \in \mathbb{R}_{\geq 0} \quad \forall f \in \mathcal{F}' \cup \{0\},\ \forall k \in \mathcal{K}. \tag{2.27}$$

Here, the constraints in Eq. (2.21) and Eq. (2.22) are replaced by the constraints in Eq. (2.24) and Eq. (2.25), respectively, to model the adjusted battery ranges under partial recharging. The constraints in Eq. (2.23) are replaced by the constraints in Eq. (2.26) to ensure that time window constraints are also adapted for the partial recharging. The constraints in Eq. (2.27) enforce the correct domain for the state of charge.

Alternative formulations

Note that the partial charging formulation from Model 4 assumes a single and constant recharging rate g_f^k for each charger $f \in \mathcal{F}'$ and vehicle $k \in \mathcal{K}$, as in Basso et al. [51]. However, other variations are also possible. For instance, Küçükoğlu, Dewil, and Cattrysse [52] introduce two distinct charging rates per charger, supporting both fast and slow charging. Ceselli et al. [53] present a model where each charger offers a subset of different charging technologies, each with a unique recharging speed and associated cost.

With any constant rate, the battery SoC is assumed to be a linear function of the charging duration. Nevertheless, in reality the charging process is nonlinear, consisting of a constant-current phase with a linear battery level increase followed by a constant-voltage phase with a concave battery level increase as the current decreases [54]. Many studies that address this non-linear behavior rely on piecewise-linear approximations, as introduced by Montoya et al. [55] and later improved upon by Froger et al. [56].

2.3. Solving the Vehicle Routing Problem

The solution methods for VRPs can be divided into three main categories: exact methods, heuristics, and learning-based optimization methods [57]. Exact methods focus on MILP formulations and guarantee an optimal solution by systematically exploring all possible solutions. While they are precise, their computational complexity makes them intractable for large-scale problem instances. Heuristic methods, on the other hand, are problem-specific and leverage the special structure of the problem to take shortcuts, so that they are faster but yield approximate solutions. Lastly, learning-based methods utilize machine learning from input data or reinforcement learning to derive solutions, adapting to problem instances and patterns observed during training. Both heuristic and learning-based methods prioritize computational efficiency over guaranteed optimality, offering solutions that are typically “good enough” for real-world applications [58]. In the following subsections, we delve into foundational and related work on solution methods for VRPs within each of these categories.

2.3.1. Exact Methods

In this section, we first provide an integer programming primer, outlining the fundamental concepts and techniques for solving the VRP exactly. We then discuss how exact methods are typically employed in practice. For more details about solving integer programs exactly, we refer to the textbook by Wolsey [29].

Linear Relaxatation

The complexity of solving the VRP as an MILP arises primarily from the presence of the integer constraints, making the problem $\mathcal{NP}$-hard. One common approach is to relax these integrality constraints, leading to the Linear Program (LP) relaxation of the problem. By doing so, the resulting formulation can be solved in polynomial time using techniques such as the simplex method [59] or interior point methods like the ellipsoid method [60]. However, the solutions obtained from the LP relaxation are often infeasible for the original problem since the integer restrictions are ignored. Despite this, a solution to an LP relaxation is useful as its objective value serves as a lower bound (or upper bound) for the objective value of the original minimization (or maximization) problem [29].

Cutting Planes

A technique known as the cutting plane method can be applied to iteratively refine solutions of an LP relaxation. This technique was introduced by Gomory [61] and uses the MILP constraints violated in the LP relaxation to introduce additional implied LP constraints, known as cutting planes, to systematically eliminate (cut) non-integer solutions from the search space without discarding any feasible MILP solutions. The resulting formulations tend to produce solutions with fewer and fewer fractional values, until all integrality constraints are satisfied.

Branch-and-Bound

Another fundamental technique for solving MILP problems is the branch-and-bound method, as introduced by Land and Doig [62]. This tree-based approach, in which nodes represent subproblems, begins by solving the LP relaxation of the MILP. If the solution does not meet all integrality constraints, the problem is split into two subproblems (branches) by imposing additional constraints to round a selected constraint-violating variable up and down, respectively. This branching process continues until all nodes either satisfy the integrality constraints or are found to be LP-infeasible. Nodes that are LP-infeasible are pruned since adding more constraints can never make them MILP-feasible. Similarly, nodes are pruned if their MILP-feasible solution does not improve the objective function compared to previously found solutions, or if a node's lower bound (in a minimization problem) is worse than that of the best MILP-feasible solution found so far. The branch-and-bound method ensures optimality by systematically exploring all branches until every node in the search tree is pruned.

Branch-And-Cut

The branch-and-bound method can be further strengthened by integrating it with the cutting plane method, resulting in the branch-and-cut method. In this hybrid approach, first introduced by Padberg and Rinaldi [63], cutting planes are dynamically added to subproblems generated during the branching process. These additional constraints enhance the formulation beyond simply bounding specific variables, leading to a more efficient exploration of the solution space.

Column Generation

Column generation is a powerful technique to solve linear programs with a large number of variables and was first described in depth by Dantzig and Wolfe [64]. The method begins by solving a simplified version of the problem, known as the Restricted Master Problem (RMP), which considers only a subset of the problem's variables while the remaining variables are temporarily excluded (implicitly set to zero). Solving the RMP provides an initial solution and an upper bound for the objective value in the case of a minimization problem. To enhance this solution, a subproblem (SP) is formulated and solved to identify new variables (columns) that can improve the objective value. Specifically, the subproblem uses the dual variables of the RMP solution and seeks for the variable with the most negative reduced cost, which is a metric that indicates how much the objective value would decrease if that variable was introduced into the solution of the RMP. If the reduced cost of the identified variable is negative, it is added to the RMP, and the process is repeated until no further variables with negative reduced cost can be identified. This iterative approach allows the algorithm to operate with only a subset of variables at any given time, gradually adding variables as needed. This makes it possible to efficiently solve large-scale linear programs that would otherwise be computationally infeasible.

Branch-And-Price

Combining branch-and-bound with column generation gives rise to the branch-and-price method, a widely used exact approach for solving VRPs. This method was originally introduced by Desrochers, Desrosiers, and Solomon [65] and is rooted in the formulation of a set-covering master problem to ensure that every customer is served, considering all feasible individual vehicle routes. Due to the exponential number of possible routes, column generation is employed within the branch-and-price framework to iteratively identify beneficial routes (columns) that could potentially improve the objective function of the master problem. The overarching set-covering master problem remains consistent across different variations of the VRP. This flexibility makes branch-and-price one of the most effective tools for tackling VRPs in practice. Many state-of-the-art commercial MILP solvers like Gurobi [66] and CPLEX [67] integrate it.

Exact Methods in Practice

Due to their limited scalability, only a small number of works rely entirely on exact methods to solve (E)VRPs. Notable examples include Soysal, Çimen, and Belbağ [68] and Wang et al. [69], both solving a pickup and delivery problem with electric vehicles under energy consumption uncertainty; Kopfer and Vornhusen [70], focusing on heterogeneous vehicles; Tahami, Rabadi, and Haouari [71], using the branch-and-cut method for the CVRP; and Lee [72], employing the branch-and-price algorithm to solve the EVRP with nonlinear charging durations.

Other works combine the branch-and-cut method and the branch-and-price method into the Branch-Price-and-Cut (BCP) method. Desaulniers et al. [50] apply it to the EVRPTW, Ceselli et al. [53] on an EVRP with multiple charging technologies, and Lera-Romero, Bront, and Soulignac [8] on an EVRPTW with time-dependent traversal durations and energy consumption. The common denominator in these works is their focus on new concepts, tested on small, artificial instances instead of realistic ones, as a proof of concept.

Many other works, including those by Hiermann et al. [47] on mixed fleets, Keskin, Laporte, and Çatay [44] on time-dependent waiting durations at rechargers, and Amiri, Zolfagharinia, and Amin [73] on uncertain energy consumption, employ exact solvers exclusively for small problem instances. The obtained solutions are often used to benchmark newly proposed metaheuristics, essential for solving larger instances [28]. The classification of "small" and "large" instances is typically determined by the number of customers. Small instances generally involve 5-15 customers, whereas large instances can include up to a hundred customers, according to the widely used benchmark developed by Schneider, Stenger, and Goeke [30].

2.3.2. Heuristics and Metaheuristics

In contrast to exact methods, inexact methods are generally fast but cannot provide any guarantee on the solution quality. Heuristics can be separated into constructive heuristics, that construct a solution from scratch by iteratively building upon a partial solution, and improvement heuristics, that generate a final solution by iteratively modifying an existing solution in order to obtain a better one [74]. In this section, we provide a few examples of fundamental constructive heuristics and detail the concept of local search, which is an improvement heuristic used by many state-of-the-art solution methods. For a more elaborate discussion on heuristic methods tailored to the VRP, we refer to surveys presented by Cordeau et al. [74] and Laporte, Ropke, and Vidal [75].

Constructive Heuristics

Notable examples of fundamental constructive heuristics for VRPs include the savings heuristic by Clarke and Wright [11], the sweep algorithm by Gillett and Miller [76] and nearest neighbour heuristics described by Applegate et al. [77]. The savings heuristic begins by treating each customer as if they are served by a separate vehicle. It then evaluates the potential cost savings of combining routes between pairs of customers. Starting with the pair that offers the highest savings, the heuristic iteratively merges routes as long as load capacity constraints remain satisfied. In contrast, the sweep algorithm operates by converting customer locations to polar coordinates and arranging them by their polar angles. Customers are grouped by "sweeping" around the depot in a circular manner, assigning them to a vehicle until its capacity is reached. The nearest neighbour heuristic begins at the depot and repeatedly visits the nearest unvisited customer until its capacity is reached. Once a vehicle has reached its load capacity, a new route is started, and the process continues until all customers have been assigned. For the EVRP, these heuristics can be adapted by introducing a second phase in which the nearest charger is inserted into a constructed route whenever the battery constraint is violated. This is done in the Modified Clarke and Wright Savings (MCWS) or two-phase heuristic as introduced by Erdoğan and Miller-Hooks [22].

While these approaches are effective for constructing feasible solutions, achieving a solution with a near-optimal objective value remains challenging. This limitation highlights the importance of improvement heuristics, such as local search techniques, to refine the initial solutions further.

Local Search

Improvement heuristics for VRPs are often based on local search [74]. Local search or neighbourhood search methods define the neighbourhood $N(\boldsymbol{x})$ of a reference solution $\boldsymbol{x}$ as the set of all solutions that are conceptually close to $\boldsymbol{x}$, according to the problem specification. The algorithm then explores a candidate solution $\boldsymbol{x}' \in N(\boldsymbol{x})$ to determine whether it improves the objective value. If an improvement is found, the process is repeated with $\boldsymbol{x}'$ as the new reference solution. This iterative approach continues until no further improvements can be achieved, so that the algorithm converges to a local optimum. The performance of local search methods is highly influenced by how the neighbourhood is defined and which candidate solutions are explored. The exploration can follow different strategies, such as random selection or steepest-descent, which evaluates all neighbours and chooses the best [78]. Local search methods do not guarantee convergence to a global optimum, as they can be trapped in local optima. To address this, local search is often enhanced through metaheuristics [79].

Neighbourhoods

Typical neighbourhoods $N(\boldsymbol{x})$ for a VRP solution $\boldsymbol{x}$ are constructed using operators such as the swap and relocate operators, introduced by Savelsbergh [80]. These operators can be applied within a single vehicle route (intra-route) or between routes (inter-route) within the current solution. Other frequently used operators include the 2-opt (intra-route) operator by Lin [81] and the 2-opt* (inter-route) operator by Potvin and and [82], which was specifically proposed for the VRPTW [83]. A specialized operator for the

EVRP is the stationInRe operator, as introduced by Schneider, Stenger, and Goeke [30]. Specifically, these operators can be described as follows:

- **Swap:** exchanges the positions of two customers within the same route or across different routes.
- **Relocate:** removes a customer from its current position in a route and inserts it into another position, either in the same route or in a different route.
- **2-opt:** removes two edges (i, j) and (k, l) from a route $R = l \rightarrow \ldots \rightarrow i \rightarrow j \rightarrow \ldots \rightarrow k \rightarrow l$, resulting in two paths $P_1 = l \rightarrow \ldots \rightarrow i$ and $P_2 = j \rightarrow \ldots \rightarrow k$. The edges in P_2 are reversed and reattached to P_1 to form a new route $l \rightarrow \ldots \rightarrow i \rightarrow k \rightarrow \ldots \rightarrow j \rightarrow l$. This effectively eliminates crossings.
- **2-opt*:** removes one edge from every route. Let R_1 and R_2 be two routes for which an edge was removed, resulting in parts P_1^1, P_2^1 and P_1^2, P_2^2, respectively. Two new routes R_1' and R_2' are formed by concatenating P_1^1 with P_2^2 and P_1^2 with P_2^1. No reversal is required for inter-route moves as there cannot be crossings.
- **stationInRe:** handles the addition and removal of chargers. It is applicable to all edges (v, w), where v represents a charger. Let w^- denote the predecessor of vertex w. If edge (v, w) is not in the current route, it is inserted alongside with edge (w^-, v), else it is removed and edge (w^-, w) is inserted instead.

Note that each of these operators gives rise to a neighbourhood whose size is at most quadratic $\mathcal{O}(n^2)$ in the size of the problem instance n, which is determined by the total number of customers. Typically the neighbourhood size for local search does not exceed $\mathcal{O}(n^3)$ [84].

Metaheuristics

A metaheuristic is a higher-level framework that can be used to combine basic heuristics into a general algorithm that is problem-independent so that it can be applied to a wide set of problems [85]. The strength of the metaheuristic search mechanism lies in its ability to balance two complementary strategies: exploration (diversification) and exploitation (intensification). Exploration focuses on investigating new search areas, while exploitation concentrates on searching within the vicinity of the best-known solutions [79]. In this section, we will give an overview of metaheuristic methods that are often used in VRP literature. For an extensive overview, we refer to surveys by Glover [85] and Awad and Elshaer [86].

Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) is a metaheuristic proposed by Mladenović and Hansen [87], that extends the local search heuristic. VNS is the most commonly used method for solving VRPs in practice [14, 86]. The key idea of VNS is to leverage multiple neighbourhood structures to escape local optima and explore a broader solution space. VNS starts by constructing an initial solution $\boldsymbol{x}$ using a construction heuristic such as those described in Subsection 2.3.2, proceeding as follows:

1. **Shaking:** In each iteration, a solution $\boldsymbol{x}'$ is randomly selected from the so-called shaking neighbourhood $N_k(\boldsymbol{x})$ of the current solution $\boldsymbol{x}$, where k indicates the current neighbourhood structure. The purpose of this step is to diversify the search and move to unexplored regions of the solution space.
2. **Local Search:** A local search with neighbourhood $N(\boldsymbol{x}')$ is then performed on $\boldsymbol{x}'$, yielding solution $\boldsymbol{x}''$. This step ensures intensification by refining the candidate solution to a local optimum.
3. **Update:** If the objective value of $\boldsymbol{x}''$ is better than that of $\boldsymbol{x}$, or some other acceptance criterion is met, $\boldsymbol{x}''$ is set as the new reference solution. If not, $\boldsymbol{x}''$ is discarded and the algorithm picks the next shaking neighbourhood $N_{k+1}(\boldsymbol{x})$ in line, looping through the neighbourhoods.

If no improvement is found after evaluating all neighbourhoods, the search terminates. The shaking neighbourhoods often have a nested structure with $N_k(\boldsymbol{x}) \subset N_{k+1}(\boldsymbol{x})$, ensuring exploitation where possible, and exploration of broader regions of the solution space when no improvements can be found locally. This makes VNS a powerful search method that converges towards high-quality solutions. For more information regarding VNS, we refer to Hansen et al. [88].

Large Neighbourhood Search

Large Neighbourhood Search (LNS) is a metaheuristic introduced by Shaw [89], that is similar to VNS and other local search heuristics but works with larger neighbourhoods that are implicitly defined by a two-stage heuristic process. By "larger", we refer to neighbourhoods of size $\mathcal{O}(n^k)$, with $k > 3$ and n

denoting the size of the problem instance [84]. Starting from an initial solution $\boldsymbol{x}$, the algorithm first employs a destroy heuristic to partially destruct $\boldsymbol{x}$, yielding a partial solution $\boldsymbol{x}'$. A repair heuristic is then applied to $\boldsymbol{x}'$ to reconstruct a feasible solution $\boldsymbol{x}''$. For the EVRP, a removal operator would typically remove a subset of customers and/or chargers from the routes, while a repair operator would reintroduce them, potentially into different routes.

The degree of destruction should be chosen carefully. If only a small part of the solution is destroyed, the heuristic may struggle to explore the search space. Conversely, if a large part of the solution is destroyed, the heuristic almost degrades into repeated re-optimization. Typically, the subset of removed elements is chosen stochastically, ensuring that each iteration focuses on a different part of the solution. Alternative removal strategies include targeting customers with high traversal costs in the current route, customers that are geographically close to one another, or customers with similar demands. For the repair operator, customers can be reinserted in the route with smallest cost increase (greedy) or regret, which is the difference in cost of inserting it in the best and one but best route. After performing a destroy-and-repair cycle, the newly generated solution $\boldsymbol{x}''$ can either be accepted as the new reference solution or be discarded based on some acceptance criterion. The acceptance criterion is typically based on the objective value but can also include stochastic elements. For more information about LNS, we refer to Pisinger and Ropke [84].

Adaptive Large Neighbourhood Search

Adaptive Large Neighbourhood Search (ALNS) is also recognized as a leading algorithm for solving VRPs in practice [86, 14]. It was developed specifically for VRPs by Ropke and Pisinger [90] and extends LNS by incorporating multiple destroy and repair heuristics into a pool rather than relying on a single heuristic pair. Each heuristic in the pool is assigned a weight that determines the likelihood of its selection. These weights are dynamically adjusted during the search process based on the empirical success of each heuristic, measured by the quality of solutions they produce. This dynamic adjustment allows the algorithm to adapt to the specific problem instance and the current state of the search. The adaptive nature of ALNS enables it to explore diverse regions of the solution space, reducing the likelihood of becoming trapped in local optima. It also simplifies calibration efforts, as the algorithm automatically tunes the influence of each heuristic. One notable downside of ALNS is that it often favors complex repair methods to achieve high-quality solutions. While this can result in superior solution quality, it typically comes at the cost of increased computational time. We refer to Pisinger and Ropke [84] for a more detailed discussion on ALNS.

Tabu Search

Tabu Search (TS) was introduced by Glover [91] as an alternative metaheuristic designed to avoid getting stuck in a local optimum during search. Gendreau, Hertz, and Laporte [92] were the first to apply it to solve VRPs. The key concept is maintaining a list of previously seen solutions that are deemed tabu (forbidden) and may not be revisited, thus allowing solutions with worse objective values to be visited for exploration and preventing the search process from cycling. Even short tabu lists can be highly effective in escaping local optima. An initial solution $\boldsymbol{x}$ is obtained using a construction heuristic. In each iteration, the best non-tabu solution $\boldsymbol{x}' \in N(\boldsymbol{x})$ is selected from the neighbourhood. Once the move from $\boldsymbol{x}$ to $\boldsymbol{x}'$ is made, $\boldsymbol{x}$ is added to the tabu list. Solutions can also be removed from the tabu list with each iteration. The algorithm proceeds until no further improvements can be made or another stopping criterion is met. Important design parameters include the content of the tabu list and the tabu tenure, which determines how long elements reside in the tabu list. For VRPs, the content of the tabu list can be certain routes, entire schedules or certain moves such as the two edges removed by a 2-opt operator. The tabu tenure can be fixed by a specified number of iterations, vary based on the quality of the current solution or vary based on the progress of the search. For example, the tenure could be set to gradually increase over the iterations to focus on exploration in the beginning and focus on intensification in later stages of the search. For more information on TS, we refer to Gendreau and Potvin [93].

Simulated Annealing

Similarly to TS, Simulated Annealing (SA) is a metaheuristic that allows degradations of the objective value in order to escape local minima. This probabilistic method was introduced by Kirkpatrick, Gelatt Jr, and Vecchi [94] and is inspired by the annealing process in metallurgy. It begins with an initial solution $\boldsymbol{x}$ obtained through a construction heuristic and explores neighbouring solutions $\boldsymbol{x}' \in N(\boldsymbol{x})$. If $\boldsymbol{x}'$ improves the objective function, it is always accepted. If $\boldsymbol{x}'$ is worse, it may still be accepted with a probability defined by $\exp(-\Delta(\boldsymbol{x}, \boldsymbol{x}')/T)$, where $\Delta(\boldsymbol{x}, \boldsymbol{x}')$ is the difference in their objective values, and T is the "temperature" parameter. Higher temperatures make it more likely to accept worse solutions, enabling broad exploration early in the process. Over time, T decreases according to a cooling schedule (e.g., $T = 0.99 \cdot T$ per iteration), reducing the likelihood of accepting worse solutions and allowing the algorithm to converge towards an optimal or near-optimal solution. Osman [95] was the first to apply SA to solve VRPs. For more details regarding SA, we refer to Delahaye, Chaimatanan, and Mongeau [96].

Evolutionary Algorithms
Evolutionary Algorithms (EAs) are population-based stochastic search algorithms that mimic the processes of natural evolution. Inspired by Darwin's theory of natural evolution [97], these algorithms model the adaptation of species to their environment over successive generations. In this metaphor, optimization is viewed as the creation of highly-fit, well-adapted individuals through a process of reproduction, genetic inheritance, and survival of the fittest [98]. In this study, we limit our discussion to genetic algorithms and ant colony optimization, as those are the most prevalent EAs used in VRP literature. For a more comprehensive discussion on evolutionary algorithms, we refer to Bartz-Beielstein et al. [98] and Yu and Gen [99].

Genetic Algorithms
Genetic Algorithms (GAs) were introduced by Holland [100] and first applied to VRPs by Van Breedam [101]. Rather than starting with a single solution that iteratively evolves, GAs work with a population of individuals (genotype), each representing different candidate solutions (phenotype), that evolves over time. During each iteration, the fitness of every individual in the population, which is determined by a function tied to the problem's objective, is evaluated. Individuals with higher fitness are more likely to influence the composition of the next generation. The new generation is formed based on three key mechanisms:

1. **Selection:** determines which individuals from the current generation will contribute genetic material to the next based on their fitness value.
2. **Cross-over:** combines selected parents to produce offspring, incorporating traits from both parents.
3. **Mutation:** randomly modifies offspring with a low probability to maintain diversity and ensure exploration of the search space.

The integration of natural selection, which focuses on exploitation, and genetic inheritance, which promotes exploration, enables GAs to effectively tackle complex optimization problems [98]. For a more detailed discussion on genetic algorithms, refer to the textbook by Goldberg [102].

Ant Colony Optimization
Ant Colony Optimization (ACO) was introduced by Dorigo, Maniezzo, and Colorni [103] to solve TSPs and later applied to VRPs by Bullnheimer, Hartl, and Strauss [104]. The method draws inspiration from foraging behavior within ant colonies, where individual ants find the shortest paths between their nest and food sources by following pheromone trails left by others. The algorithm starts by deploying a set (population) of artificial ants, which represent vehicles, from random customers in the network. These ants explore paths across the customer graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ randomly, selecting the next unvisited customer with equal probability until all customers are visited by every ant. The ants deposit pheromones $\Delta\tau_{ij}$ on the edges $(i, j) \in \mathcal{E}$ they traverse, making those paths more likely to be chosen in subsequent iterations. In later iterations, the selection of edges by ants is determined by two factors: how effective the traversal of that edge was in previous iterations (stored in pheromone trails τ_{ij}) and the inherent promise of that edge (based on local heuristics like distance d_{ij}, energy consumption h_{ij}, or traversal duration t_{ij}). Over time, pheromones on less optimal routes evaporate, reducing their influence, while shorter and better routes accumulate more pheromone and thus become increasingly favored. Note that this process yields solutions for the TSP. To adapt it for the EVRP, capacity and battery constraints can be incorporated by having ants return to the depot when load capacity is met [104] and by inserting the nearest charger when the remaining battery energy is insufficient for the next customer [105]. For more information on ant colony optimization, we refer to Dorigo and Stützle [106].

Memetic Algorithms
Memetic algorithms are a class of population-based metaheuristics that leverage the exploratory strength of evolutionary methods alongside the intensification capabilities of neighbourhood-based local search heuristics [107]. One notable example is Hybrid Genetic Search (HGS), introduced by Vidal et al. [108] to

address the Multi-Depot Periodic Vehicle Routing Problem (MDPVRP), which involves multiple depots and multi-day planning. This approach was later extended into a unified framework capable of solving a wide range of multi-attribute VRPs [109].

In HGS, offspring are created through genetic crossover and subsequently refined using local search with swap and relocate operators. The offspring are then categorized into feasible and infeasible populations. Infeasible solutions undergo a repair process with a certain probability, during which the penalty for constraint violations in the cost function is increased tenfold and another round of local search is performed to restore feasibility. Successfully repaired solutions are moved to the feasible population. A greedy elimination strategy is applied whenever the population exceeds a predefined limit, keeping the fittest individuals in the population.

Metaheuristics in Practice

All the aforementioned metaheuristics have successfully been applied to solve larger instances of EVRPs. For a detailed overview per metaheuristic, we refer to the review by Kucukoglu, Dewil, and Cattrysse [27].

Among all methods, ALNS stands out as the most frequently employed in works addressing the EVRP. Demir, Bektaş, and Laporte [110] were among the first to apply ALNS to the green vehicle routing problem introduced by Erdoğan and Miller-Hooks [22]. Subsequent studies have extended its use to various EVRP variants: Goeke and Schneider [48] addressed the MF-EVRPTW, Keskin and Çatay [31] focused on the EVRPTW-PR, Keskin, Laporte, and Çatay [44] incorporated time-dependent waiting durations at charging stations, Keskin, Çatay, and Laporte [111] considered time uncertainty, and Pelletier, Jabali, and Laporte [4] and Wang, Zheng, and Zhou [112] used it to solve the EVRPTW-PR under energy consumption uncertainty.

The combination of VNS with TS was introduced by Schneider, Stenger, and Goeke [30] for the EVRPTW and later adopted by Bruglieri et al. [49] for the EVRPTW-PR. Jeong et al. [113] also applied this hybrid approach to a variant involving energy consumption uncertainty. SA has been used as the primary heuristic by Felipe et al. [114], who integrated multiple local search strategies to solve the EVRPTW-PR.

Evolutionary and memetic algorithms are gaining traction in this domain too. For instance, Zhang et al. [105] and Mao et al. [115] utilized an ACO approach, while Amiri, Zolfagharinia, and Amin [73] combined ALNS with a multi-objective GA to address energy consumption uncertainty. Additionally, Xiao et al. [116] proposed a memetic algorithm that integrates GA with local search under TS to handle mixed backhaul scenarios, whereas Dong, Wang, and Zhang [117] introduced one with Adaptive Local Search (ALS) to solve a variant of the VRP with dynamic demands. Hiermann et al. [118] applied HGS to solve the MF-EVRPTW with a mixed fleet of conventional and electric vehicles, while Vidal et al. [119] used it to address a variant with time-dependent service and traversal durations.

2.3.3. Learning-based Methods

Learning-based methods for solving the VRP come in many flavours. Since we do not focus on learning-based methods in this work, we limit our discussion to an overview of some useful applications. We refer to the surveys by Li et al. [57], Bai et al. [120], and Bogyrbayeva et al. [121] for more comprehensive reviews of learning-based optimization methods for the VRP.

Problem Decomposition

In practical settings, a widely adopted approach to manage the complexity of large-scale VRP instances is the Cluster First, Route Second (CFRS) strategy [122]. This paradigm begins by assigning customers to specific vehicles through clustering and then constructs individual vehicle routes using either (meta)heuristics or exact optimization techniques [120]. This introduces the challenging task of making optimal clusters, which is in itself an $\mathcal{NP}$-hard problem. However, unsupervised learning methods, such as K-Means [123] and density-based clustering [124], can be used to cluster the customers efficiently, like in the work by Erdoğan and Miller-Hooks [22], where a density-based spatial clustering method is combined with the MCWS heuristic.

End-To-End Learning

End-to-end-learning approaches aim to directly learn a mapping from problem instances to routing solutions, without relying on intermediate optimization steps. Hence, once trained, they can quickly generate solutions for unseen instances during inference. Models for end-to-end learning often take the shape of advanced deep learning models including Graph Neural Networks (GNNs) [125] to extract network features and Recurrent Neural Networks (RNNs) [126] to deal with sequential information. Supervised end-to-end-learning approaches are typically only applied to simpler routing problems that resemble a TSP, because they depend on large datasets of high-quality VRP solutions, which are often unavailable [57, 121].

To overcome this limitation, Reinforcement Learning (RL) [127] has emerged as a promising end-to-end-learning alternative. In this approach, an agent interacts with the environment, represented by the current routing solution, by selecting actions, such as assigning a customer to a specific vehicle route. The agent then receives a reward, which reflects the impact of that action on the objective function. This feedback loop enables the agent to refine its decision-making strategy over time based on the outcomes of its actions in various environment states. Unlike supervised learning methods, RL methods do not rely on precomputed solutions, making them particularly well-suited for tackling complex and dynamic VRPs [57, 121]. A notable application of RL to solve complex EVRPs is the study by Basso et al. [128], in which a safe RL approach is used to address uncertainties in energy consumption.

Improving Heuristics

Instead of replacing them, machine learning methods can also be used to improve heuristic solution methods for the VRP. These often involve a large number of hyper-parameters, such as the destroy rate in ALNS, the acceptance criteria in SA or the population size in GAs. The challenge of finding good parameters to ensure high quality solutions is a problem on its own, known as the Parameter Setting Problem (PSP) [120]. Data-driven regression models can be used to link problem-specific features to optimal parameter values. An example is the work by Calvet et al. [129] that introduces statistical learning to fine-tune an Iterated Local Search (ILS) heuristic [130] for the multi-depot VRP.

Beyond parameter tuning, machine learning can be made to play a role in guiding the search process itself, learning which neighbourhood functions to select during local, variable, and large neighbourhood search procedures. An example is the work by Lopes Silva et al. [131], who utilized a multi-agent RL framework to train agents for LS neighbourhood selection in solving the VRP with time windows.

More recently, Nadi et al. [132] explored the use of Bayesian optimization [133] to learn spatial and temporal preferences of human fleet planners from partially observed tours. These learned preferences were then integrated into the ALNS objective function, enabling the metaheuristic to generate routes that better reflect human decision-making.

2.4. Modelling the Energy Consumption of Electric Vehicles

The modelling of energy consumption for electric vehicles is a critical aspect of the EVRP, as it directly influences the feasibility of the planned routes in practice. Thus far, we have abstracted energy consumption into a deterministic parameter h_{ij}^k that is assumed to be known for every edge $(i, j) \in \mathcal{E}$ and vehicle $k \in \mathcal{K}$. In reality, this highly complex value is often not readily available as it depends on many factors including traversal distance, traffic and road conditions, vehicle characteristics, load level, and environmental conditions [27]. Works such as those by Erdoğan and Miller-Hooks [22] and Schneider, Stenger, and Goeke [30] consider the energy consumption to simply be a linear function of the the travelled distance. Although such a linear formulation eases the formulation of a MILP, non-linear approaches modelling the longitudinal dynamics of the vehicles can often obtain more realistic results [27].

2.4.1. Longitudinal Vehicle Dynamics Model

Many non-linear modelling techniques stem from the research by Barth, Younglove, and Scora [134], modelling the longitudinal dynamics of vehicles to estimate engine power demand. Such models, as first applied to the VRP setting by Demir, Bektaş, and Laporte [135], calculate the mechanical power required to overcome rolling resistance, aerodynamic drag, and gravitational forces, while accounting for engine power losses. Examples include the works of Pelletier, Jabali, and Laporte [4], Goeke and Schneider [48], Basso et al. [51], and Amiri, Zolfagharinia, and Amin [73].

What sets HGEVs apart from combustion engine Heavy Goods Vehicles (HGVs) is their ability to recuperate energy. During deceleration or while driving downhill, the traction force can become negative, allowing energy to be transferred back to the battery. In this process, known as regenerative breaking, the electric motor functions as a generator, converting the vehicle's kinetic energy back into electrical energy that can be reused to power the vehicle.

Since regenerative braking can extend the vehicle's range, it is often excluded to provide more conservative range estimates, as in Pelletier, Jabali, and Laporte [4]. However, to obtain more realistic estimates, the model by Asamer et al. [6] includes recuperation. Table 2.5 and Model 5 illustrate the longitudinal vehicle dynamics modelling approach including recuperation to model the total energy consumption of a vehicle as a function of the total traversal duration [6, 134].

Parameters	
g	Gravitational constant [in m/s^2]
c_r	Rolling friction coefficient [in N/kN]
ρ_a	Air density [in kg/m^3]
v_{min}	Minimum velocity of the vehicle required for regeneration [in m/s]
c_d	Aerodynamic drag coefficient
A_f	Frontal area of the vehicle [in m^2]
η_M	Energy efficiency of transmission, motor and power conversion [in %]
η_G	Energy efficiency of transmission, generator and in-vehicle charger [in %]
P_0	Power demand of auxiliary components (e.g: airco, infotainment) [in W]
T	Route duration [in s]
t	Timestep within range $0 \dots T$ [in s]
Variables	
$m(t)$	Total vehicle mass at timestep t [in kg]
$\theta(t)$	Gradient angle of the road at timestep t [in degrees]
$v(t)$	Velocity of the vehicle at timestep t [in m/s]
$a(t)$	Acceleration of the vehicle at timestep t [in m/s^2]
$F_r(t)$	Rolling resistance at timestep t [in N]
$F_a(t)$	Aerodynamic drag at timestep t [in N]
$F_g(t)$	Gravitational force at timestep t [in N]
$F_t(t)$	Traction force at timestep t [in N]
$P_M(t)$	Mechanical power at timestep t [in W]
$P_{out}(t)$	Total power demand at timestep t [in W]
$P_{in}(t)$	Recuperation energy at timestep t [in W]
$P(t)$	Total powerat timestep t [in W]
$E(T)$	Total vehicle energy consumption for route duration T [in kWh]

Table 2.5: Notation overview for the Longitudinal Vehicle Dynamics Model for Energy Consumption

Model 5 – Longitudinal Vehicle Dynamics Formulation for Energy Consumption

Rolling resistance:

$$F_r(t) = c_r \cdot m(t) \cdot \cos\theta(t). \tag{2.28}$$

Aerodynamic drag:

$$F_a(t) = \frac{1}{2} \cdot \rho_a \cdot A_f \cdot c_d \cdot v(t)^2. \tag{2.29}$$

Gravitational force:

$$F_g(t) = m(t) \cdot g \cdot \sin\theta(t). \tag{2.30}$$

Traction force:

$$F_t(t) = m(t) \cdot a(t) + (F_r(t) + F_a(t) + F_g(t)). \tag{2.31}$$

Mechanical power:

$$P_M(t) = F_t(t) \cdot v(t). \tag{2.32}$$

Total power demand:

$$P_{out}(t) = \frac{P_M(t)}{\eta_M} + P_0. \tag{2.33}$$

Recuperation of energy:

$$P_{in}(t) = \begin{cases} 0, & \text{if } v(t) \leq v_{min}. \\ F_t(t) \cdot v(t) \cdot \eta_G + P_0, & \text{otherwise.} \end{cases} \tag{2.34}$$

Total power:

$$P(t) = \begin{cases} P_{out}(t), & \text{if } F_t(t) \geq 0. \\ P_{in}(t), & \text{if } F_t(t) < 0. \end{cases} \tag{2.35}$$

Total energy consumption:

$$E(T) = \frac{1}{3.6 \cdot 10^6} \cdot \int_0^T P(t)\, dt. \tag{2.36}$$

Here, Eq. (2.28) to Eq. (2.30) model the forces acting on the vehicle, which vary with speed $(v(t), a(t))$, the curb and cargo weight of the vehicle $(m(t))$, and the road, environmental, and tire conditions $(\theta(t), \rho_a, c_r)$. The traction force from Eq. (2.31) is the force required to achieve the desired acceleration $a(t)$, according to Newton's Second Law. Equations Eq. (2.32) and Eq. (2.33) translate this to the total power $P_{out}(t)$ that the electric motor needs to deliver, given the current velocity $v(t)$, while accounting for the energy efficiency η_M of the transmission, motor, and power conversion, as well as additional energy consumption P_0 by auxiliary components such as air conditioning and infotainment systems. Eq. (2.34) describes the recuperation of energy. During recuperation, parameter η_G accounts for the efficiency of the transmission, generator, and in-vehicle charger. Note that the total power $P(t)$ in Eq. (2.35) can be negative due to energy recuperation. To determine the total energy consumption from the vehicle's battery in kWh, Eq. (2.36) integrates the total power $P(t)$ over the route duration T.

2.4.2. Parameter Analysis

The parameters in Table 2.1 are assumed to be known and fixed for every tour. The gravitational constant g is known to be 9.81 $\mathrm{m/s^2}$. The values for the rolling friction coefficient c_r for heavy duty (category C3) tyres found within the literature lie within the range of 5.01 to 6.80 N/kN. This value is also regulated by the European Union to be at most 6.5 N/kN since 2020. In the baseline heavy duty tractor-trailer combination model developed by Oscar Delgado and Muncrief [136], a value of 5.5 N/kN is taken as the baseline. The air density ρ_a depends on the temperature and ranges from 1.34 to 1.16 $\mathrm{kg/m^3}$ assuming temperatures ranging from −10 to 30 °C [137]. The minimum velocity v_{min} required to recuperate energy while breaking or descending is estimated to be around 15 km/h [6, 138]. The aerodynamic drag coefficient c_d and the vehicle's frontal area A_f vary based on the manufacturer and model. These two factors are frequently combined into a single measurement known as the air drag area $c_d \cdot A_f$, expressed in $\mathrm{m^2}$. In 2021, the average air drag area across all European truck manufacturers was approximately 5.63 m^2 [139].

However, several truck brands have introduced new cabins with more aerodynamic designs since then. Considering the energy efficiency as a constant is a simplification of reality as it varies depending on the vehicle and environmental conditions such as temperature. Notably, Fetene et al. [140] found a 34% decrease in the overall energy efficiency of BEVs during winter compared to summer. Asamer et al. [6] analyzed values ranging from 0.63 to 0.90 for η_M and from 0.64 to 0.82 for η_G in their sensitivity analysis, setting baseline values at 0.9 and 0.8, respectively. Moreover, their sensitivity analysis concludes that the energy consumption is most sensitive to η_M out of all constants. Basso et al. [51] models the overall energy efficiency as a second-degree polynomial in the velocity and estimates the values using regression. The power demand of auxiliary components such as air conditioning is highly dependent on driver behavior. Pelletier, Jabali, and Laporte [4] report a baseline of 1.58 and an upper bound of 4.43 kW for medium-duty electric vehicles. Oscar Delgado and Muncrief [136] report slightly higher values for heavy duty tractor-trailer combinations ranging from 4.55 to 6.65 kW, and establish a baseline of 5.5 kW.

2.4.3. Variable Simulation

In Model 5, the variables under simulation are the total vehicle mass $m(t)$, gradient angle of the road $\theta(t)$, velocity $v(t)$ and acceleration $a(t)$. To obtain energy consumption values h_{ij}^k for every edge $(i,j) \in \mathcal{E}$ and vehicle $k \in \mathcal{K}$, one could explicitly model the underlying road network of every edge $(i,j) \in \mathcal{E}$ as a separate graph $\mathcal{E}_r = (\mathcal{V}_r, \mathcal{E}_r)$ of road segments in which the traversal of the shortest path $S_{ij} \subseteq \mathcal{E}_r$ from i to j can be simulated for vehicle k to obtain the values of the variables for $t = 0 \ldots T$, setting T to be the time t_{ij} required to traverse $(i,j) \in \mathcal{E}$, according to the simulation [51].

Vehicle Mass

The curb mass of heavy duty tractor-trailer combinations typically lies in the range of 14,000 to 15,700 kg, with Oscar Delgado and Muncrief [136] setting baselines of 7,000 kg for the tractor and 7,400 kg for the trailer. On European roads, the vehicle's total mass including the cargo, also referred to as Gross Combination Mass (GCM), is restricted to be 40,000 kg, which leaves a maximum payload of 25,600 kg. However, in practice the payload is typically around 19,300 kg [136].

In Model 5, note that the total mass m influences all forces but aerodynamic drag, making mass a crucial factor in energy consumption. During a trip, the total mass m of a vehicle changes over time as it makes stops to partially unload its cargo. However, for a traversed road connection $(i,j) \in \mathcal{E}$, the mass remains constant. Therefore, instead of being a function of timestep t, it can also be expressed as a linear function of the currently loaded cargo quantity u, using

$$m(u) \coloneqq m_c^k + m_u u, \tag{2.37}$$

where m_c^k is the curb mass of vehicle $k \in \mathcal{K}$ and m_u is the mass of one unit of cargo [48]. Note that in Section 2.2, we defined u_j^k to denote the amount of cargo in the vehicle upon arrival at location j. Hence, the total vehicle mass of vehicle $k \in \mathcal{K}$ when traversing edge $(q,r) \in S_{ij}$ is equal to $m(u_j^k)$.

To maintain linearity in the optimization problem, Wang et al. [69] and Goeke and Schneider [48] precompute edge-based energy consumption values using simulations for different mass values and vehicles. They then model the energy consumption h_{ij}^k as an affine function of the vehicle's mass $m(u_j^k)$ defined by Eq. (2.37), for every edge $(i,j) \in \mathcal{E}$ and vehicle $k \in \mathcal{K}$.

Gradient Angle

Although the Netherlands is characterized by predominantly flat terrain, such that gradient angles seem insignificant, Donkers, Yang, and Viktorović [5] considered the effect of gradient angles on energy consumption of passenger BEVs in the Netherlands and found that a slight slope of 1° could already double the energy consumption when driving at velocities around 30 km/h, while adding 30 to 35% to the energy consumption when driving at higher velocities at the motorway. Although similar studies tailored to HGEVs could not be identified, the impact of gradient angles on energy consumption is expected to be even more significant for HGEVs due to the larger mass term in the gravitational force. To take the gradient of traversed road segments $(q,r) \in \mathcal{E}_r$ into account, one can leverage elevation data from APIs such as HERE [141] or open data projects such as Shuttle Radar Topography Mission (SRTM) [142] and European Digital Elevation Model (EU-DEM) [143].

Velocity and Acceleration

The velocity and the acceleration are closely related as the acceleration at timestep t directly determines the velocity at timestep $t + 1$. Note that in Model 5, the velocity term in Eq. (2.29) for aerodynamic drag is the only higher-order term in the entire model, making it the most influential factor for the energy consumption at high velocities. Goeke and Schneider [48], Amiri, Zolfagharinia, and Amin [73], and Pelletier, Jabali, and Laporte [4] assume a constant velocity of $90, 60$ and $45\,\text{km/h}$, respectively over each edge $(i, j) \in \mathcal{E}$, omitting acceleration and disregarding the characteristics of the underlying road network $\mathcal{E}_r$.

Basso et al. [51] found that estimating total energy solely based on average velocity accounts for only about 76% of the total energy estimated when the acceleration and deceleration contributing to that average velocity are also considered. Hence, they consider the traversal of underlying road segments $(q, r) \in \mathcal{E}_r$, which are each subdivided into three phases to account for acceleration and deceleration. During the acceleration phase, the vehicle speeds up from an initial velocity to a cruising velocity with a constant acceleration of at most $2\,\text{m/s}^2$. In the cruising phase, acceleration is zero, and the velocity remains constant. Finally, in the braking phase, the velocity decreases with constant deceleration. The energy consumption per phase is calculated based on its average velocity and constant acceleration. The energy consumption for edge $(q, r) \in \mathcal{E}_r$ is the sum of that of its three phases and the total energy consumption per edge $(i, j) \in \mathcal{E}$ is determined by summing over S_{ij}.

Some works consider traffic and other road conditions to craft accurate duty cycles, which give a second-to-second speed profile using data or detailed simulations over the road segments (q, r) in the road graph $\mathcal{E}_r$. Basso et al. [51] validate their three phase model by using the Simulation of Urban MObility (SUMO) simulator developed by Lopez et al. [144] to model various traffic conditions. In SUMO the traffic is continuously simulated whilst a vehicle follows a predetermined route. In this case, SUMO continuously sends a target speed to a proprietary vehicle model, which simulates the driver and vehicle attempting to follow the duty cycle, accelerating and braking to reach the target velocity. Traffic lights, turns, surrounding vehicles, and other traffic situations are all simulated by SUMO to generate an accurate speed profile. Although accurate and therefore great for validation purposes, the application of SUMO requires a detailed mapping of the entire operational area of the vehicles, rendering it infeasible for country-wide operations.

Wang et al. [69] instead pair the AMBER vehicle simulator [145] with the HERE [141] routing API, which accounts for traffic conditions and returns the driving path to follow as well as the duty cycle. Keyantuo et al. [146] do not employ their own longitudinal vehicle model and instead rely entirely on the energy calculations performed by the HERE API internally based on its integrated vehicle dynamics model. Similarly, Woo et al. [147] input the routes returned by HERE to another API named WideSense [148] to calculate high-precision estimates of the traversal durations and energy consumptions for every edge $(i, j) \in \mathcal{E}$ based on a proprietary model. The drawback of using such APIs is that they have limited free quotas and often require paid subscriptions to handle the number of requests needed to populate a reasonably sized traversal duration or energy consumption matrix.

2.5. Incorporating Uncertainty into Optimization Problems

In classical EVRPs modelled as MILPs, parameters such as traversal durations, service durations, and energy consumption are assumed to be deterministic. However, in many real-life scenarios, these parameters are uncertain. When this uncertainty is ignored, the resulting solutions can often violate real-world constraints, leading to vehicles arriving late or running out of battery. Hence, various methods have been developed to account for such uncertainties within optimization models. We discuss the two primary approaches to optimization under uncertainty: stochastic optimization, and robust optimization [149, 150, 151, 152].

2.5.1. Stochastic Optimization

Stochastic Optimization (SO) refers to optimization techniques where some parameters are uncertain (stochastic) and modelled as random variables with known probability distributions. These techniques aim to make decisions that perform well on average, often incorporating probabilistic guarantees [149].

Recall the definition of an MILP from Definition 2. We now consider an extension where uncertainty affects a part of the constraint system. Let the constraint matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and the right-hand side vector $\boldsymbol{b} \in \mathbb{R}^{m}$ be known and fixed (deterministic). Additionally, let $\boldsymbol{L}(\boldsymbol{z}) \in \mathbb{R}^{m_1 \times n}$ and $\boldsymbol{l}(\boldsymbol{z}) \in \mathbb{R}^{m_1}$ form a second set of constraint parameters, whose elements are given by affine functions of the random variable vector $\boldsymbol{z} \in \mathbb{R}^{l}$, which is assumed to have a known probability distribution and probability measure $\mathbb{P}$, resulting in

$$\boldsymbol{L}(\boldsymbol{z})\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}). \tag{2.38}$$

A naive method to address the uncertainty is to replace the random vector $\boldsymbol{z}$ with its expected value $\mathbb{E}[\boldsymbol{z}]$, and solve the resulting deterministic problem. However, this approach provides no formal guarantee on the feasibility of the solution under realizations of $\boldsymbol{z} \neq \mathbb{E}[\boldsymbol{z}]$ that deviate from their expectation. To explicitly account for such realizations, one may instead adopt a chance-constrained approach, where Eq. (2.38) is enforced to hold with a high probability of least $(1 - \epsilon)$ for some tolerance level $\epsilon \in (0, 1)$. This leads to a Stochastic Linear Program (SLP) with probabilistic constraints, formulated as follows [153]:

Model 6 – SLP with Probabilistic Constraints

$$\zeta^* = \min_{\boldsymbol{x}} \quad \boldsymbol{c}^\top \boldsymbol{x} \tag{2.39}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \tag{2.40}$$

$$\mathbb{P}(\boldsymbol{L}(\boldsymbol{z})\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z})) \geq 1 - \epsilon. \tag{2.41}$$

The chance constraint modelled by Eq. (2.41) is generally non-linear in $\boldsymbol{x}$ and can be computationally challenging to evaluate or optimize over, especially when the distribution of $\boldsymbol{z}$ is complex. Moreover, in practice, the exact probability distribution may not be known a priori, so it has to be estimated from finite data samples using methods such as Monte Carlo simulation, which introduces additional modelling uncertainty.

Stochastic Optimization with Recourse

An alternative way of dealing with stochasticity is to divide the decision-making process into two stages, a technique known as stochastic optimization with recourse. In this setup, first-stage decisions are made before the uncertainty is revealed, whereas second-stage decisions or recourse actions are made after the uncertainty is realized, allowing adjustments to the initial decisions to maintain feasibility under the true conditions. The goal is to minimize the expected total cost, which includes both the first-stage cost and the expected cost of recourse actions given the first-stage decisions.

Starting from the MILP definition from Definition 2, the corresponding SLP can be modelled as follows. Let $\boldsymbol{z} \in \mathbb{R}^{l}$ be a random vector representing uncertainty, defined on a probability space with a known distribution. Next to the decision vector $\boldsymbol{x} \in \mathbb{R}^{n}$ for first-stage decisions, let the vector $\boldsymbol{y} \in \mathbb{R}^{n_2}$ comprise second-stage decision variables that depend on the realization of $\boldsymbol{z}$. Let the cost coefficients for the recourse actions be given by vector $\boldsymbol{d} \in \mathbb{R}^{n_2}$ and let $\boldsymbol{F}(\boldsymbol{z}) \in \mathbb{R}^{m_2 \times n_2}$ and $\boldsymbol{f}(\boldsymbol{z}) \in \mathbb{R}^{m_2}$ be the constraint matrix and right-hand vector imposing constraints on $\boldsymbol{y}$. Similarly, let matrices $\boldsymbol{G}(\boldsymbol{z}) \in \mathbb{R}^{m_3 \times n}$ and $\boldsymbol{H}(\boldsymbol{z}) \in \mathbb{R}^{m_3 \times n_2}$ and right-hand vector $\boldsymbol{j}(\boldsymbol{z}) \in \mathbb{R}^{m_3}$ define coupling constraints between the first- and second-stage variables. The parameters of these matrices and vectors are again affine functions of the uncertainty realization $\boldsymbol{z}$.

$\Omega(\boldsymbol{x}, \boldsymbol{z})$ can then be defined as the set of all feasible second-stage decisions $\boldsymbol{y}$ for given first-stage decisions $\boldsymbol{x}$ and uncertainty realization $\boldsymbol{z}$:

Definition 6 – Second-Stage Feasible Region

$$\Omega(\boldsymbol{x}, \boldsymbol{z}) := \{\boldsymbol{y} : \boldsymbol{F}(\boldsymbol{z})\boldsymbol{y} \leq \boldsymbol{f}(\boldsymbol{z}),\ \boldsymbol{G}(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{H}(\boldsymbol{z})\boldsymbol{y} \leq \boldsymbol{j}(\boldsymbol{z}),\ \boldsymbol{y} \geq \boldsymbol{0}\}.$$

Furthermore, a recourse function $Q(\boldsymbol{x}, \boldsymbol{z})$ can be defined to give the optimal value of the second-stage problem given the first-stage decisions $\boldsymbol{x}$ and uncertainty realization $\boldsymbol{z}$:

Definition 7 – Recourse Function

$$Q(\boldsymbol{x}, \boldsymbol{z}) := \min_{\boldsymbol{y} \in \Omega(\boldsymbol{x}, \boldsymbol{z})} \{\boldsymbol{d}^\top \boldsymbol{y}\}.$$

Now, the resulting SLP with recourse is given by:

Model 7 – SLP with Recourse

$$\zeta^* = \min_{\boldsymbol{x}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \mathbb{E}_{\boldsymbol{z}}[Q(\boldsymbol{x}, \boldsymbol{z})] \tag{2.42}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}. \tag{2.43}$$

In practice, the expectation $\mathbb{E}_{\boldsymbol{z}}[Q(\boldsymbol{x}, \boldsymbol{z})]$ is typically not readily available in closed form. However, if a finite set of scenarios is given $\{\boldsymbol{z}^s\}_{s=1}^S$, where each scenario $\boldsymbol{z}^s$ occurs with known probability p_s, the expectation amounts to

$$\mathbb{E}_{\boldsymbol{z}}[Q(\boldsymbol{x}, \boldsymbol{z})] = \sum_{s=1}^{S} p_s Q(\boldsymbol{x}, \boldsymbol{z}^s). \tag{2.44}$$

Substituting this into Model 7 results in a linear optimization problem, that can be solved using the techniques discussed in Section 2.3. If the exact probabilities are unknown but a dataset of observed uncertainty realizations is available, MCS can be used to estimate the expected value. For further reading on stochastic optimization techniques, we refer to the textbooks by Birge and Louveaux [149], Kall, Mayer, et al. [153], and Heyman and Sobel [150].

2.5.2. Robust Optimization

Unlike stochastic optimization, Robust Optimization (RO) does not rely on known probability distributions for the uncertain parameters. Instead, uncertainty is represented through uncertainty sets, which contain all plausible realizations of uncertainty. Uncertainty sets are often constrained by an uncertainty budget that controls how much uncertainty is allowed. Robust optimization techniques aim to identify solutions that perform well under the worst-case scenario within these sets, rather than to optimize for the expected outcome, thereby providing stronger guarantees on feasibility than SO techniques [151].

Let $\boldsymbol{z} \in \mathcal{U} \subset \mathbb{R}^l$ be an l-dimensional vector containing uncertain variables $(z_1, \ldots, z_l)$, so that every uncertainty realization $\boldsymbol{z}$ is part of uncertainty set $\mathcal{U}$. Borrowing Eq. (2.38) from the stochastic formulation to represent constraints on $\boldsymbol{x}$ under the uncertainty realization $\boldsymbol{z} \in \mathcal{U}$, one can formulate the robust optimization problem as follows:

Model 8 – Robust Optimization

$$\zeta^* = \min_{\boldsymbol{x}} \max_{\boldsymbol{z} \in \mathcal{U}} \quad \boldsymbol{c}^\top \boldsymbol{x} \tag{2.45}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \tag{2.46}$$

$$\boldsymbol{L}(\boldsymbol{z})\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}) \quad \forall \boldsymbol{z} \in \mathcal{U}. \tag{2.47}$$

Since the uncertainty vector $\boldsymbol{z} \in \mathcal{U}$ can take infinitely many real-valued realizations, Eq. (2.47) generally represents an infinite family of constraints. To make the problem tractable, the uncertainty set $\mathcal{U}$ is typically chosen to be a compact and convex subset of $\mathbb{R}^l$, meaning it is both closed and bounded. For formal definitions of these concepts, we refer to the textbook by Bertsimas, Brown, and Caramanis [151].

According to Carathéodory's theorem, any point $\boldsymbol{z} \in \mathcal{U}$ can be expressed as a convex combination of at most $l + 1$ extreme points of $\mathcal{U}$, provided that $\mathcal{U}$ is compact and convex. Furthermore, since the constraint function (left-hand side) of Eq. (2.47) is linear in $\boldsymbol{z}$, it is also convex. A fundamental result in convex analysis states that a convex function over a compact convex set attains its maximum at an extreme point of that set. Therefore, to ensure that the constraint in Eq. (2.47) holds for all $\boldsymbol{z} \in \mathcal{U}$, it is sufficient to verify that it holds for all extreme points of $\mathcal{U}$ [154, 155]. This reduces the infinite constraint set to a finite one, indexed by the set of extreme points ext($\mathcal{U}$):

$$\boldsymbol{L}(\boldsymbol{z})\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}) \quad \forall \boldsymbol{z} \in ext(\mathcal{U}). \tag{2.48}$$

Thus, substituting Eq. (2.48) for Eq. (2.47) results in a robust model with finitely many constraints.

Adaptive Robust Optimization

In Model 8, all decisions are made simultaneously, accounting for every possible uncertainty realization. This results in a single solution that is heavily biased towards the worst-case scenario and therefore very conservative [151]. Like stochastic optimization with recourse, the robust optimization paradigm also has an adaptive variant, named Adaptive Robust Optimization (ARO), in which the decision process is split in two stages to make the solutions less conservative.

Using the definition and notation for the recourse function from Definition 7, one can formulate the adaptive robust optimization model as follows:

Model 9 – Adaptive Robust Optimization

$$\zeta^* = \min_{\boldsymbol{x}} \quad (\boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{z} \in \mathcal{U}} Q(\boldsymbol{x}, \boldsymbol{z})) \tag{2.49}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}. \tag{2.50}$$

$\equiv$

$$\min_{\boldsymbol{x},\, \eta} \quad \eta \tag{2.51}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \tag{2.52}$$

$$\eta \geq \boldsymbol{c}^\top \boldsymbol{x} + Q(\boldsymbol{x}, \boldsymbol{z}), \quad \forall \boldsymbol{z} \in \mathcal{U}, \tag{2.53}$$

$$\eta \in \mathbb{R}_{\geq 0}. \tag{2.54}$$

Here. the equivalent second variant is based on the assumption that the maximum of an infeasible problem with $\Omega(\boldsymbol{x}, \boldsymbol{z}) = \emptyset$ is defined as $-\infty$.

Essentially, this just replaces the expectation in Model 7 with a maximization over the uncertainty set $\mathcal{U}$ to account for the worst-case scenario, and incorporates the constraints defined in Eq. (2.47) into the coupling constraints from Definition 7. Under the assumption of a compact, convex set, $ext(\mathcal{U})$ can again be substituted for $\mathcal{U}$.

2.5.3. Uncertainty Sets

Usually, uncertainty sets for RO are modelled as polyhedra, which are closed and compact sets that can be described by finitely many linear equations, so that membership can be modelled using an MILP. Next, we review several commonly used uncertainty sets in robust optimization, as presented in the textbook by Bertsimas, Brown, and Caramanis [151]:

Box Uncertainty Set

The most trivial uncertainty set is the box uncertainty set. Let z_i^{nom} denote the nominal or mean values of z_i for $i = 1 \ldots l$, which we will henceforth denote as $i \in [l]$. Similarly, let z_i^{dev} be the symmetric maximal deviation from z_i^{nom}, so that all variables z_i reside within intervals $[z_i^{nom} - z_i^{dev}, z_i^{nom} + z_i^{dev}]$. This gives rise to the box uncertainty set $\mathcal{U}_{box}$:

Definition 8 – Box Uncertainty Set

$$\mathcal{U}_{box} \coloneqq \{\boldsymbol{z} \in \mathbb{R}^l \;:\; \forall i \in [l]\,,\; z_i^{nom} - z_i^{dev} \le z_i \le z_i^{nom} + z_i^{dev}\}.$$

Budgeted Uncertainty Set

The box uncertainty set assumes that all l variables z_i can attain their worst-case deviation $|z_i - z_i^{nom}|$ at the same time, giving rise to very pessimistic scenarios that are very unlikely in reality. Hence, a parameter Γ, denoting the maximum number of simultaneous worst-case deviations, is often introduced to bound the total uncertainty within a realization $\boldsymbol{z}$. This gives rise to the budgeted uncertainty set $\mathcal{U}_{budget}$:

Definition 9 – Budgeted Uncertainty Set

$$\mathcal{U}_{budget} \coloneqq \{\boldsymbol{z} \in \mathbb{R}^l \;:\; \forall i \in [\,l\,]\,,\; z_i^{nom} - z_i^{dev} \le z_i \le z_i^{nom} + z_i^{dev}\,,\; \sum_{i=1}^{l} \frac{|z_i - z_i^{nom}|}{z_i^{dev}} \le \Gamma\}.$$

Note that setting $\Gamma \ge l$ yields the box uncertainty set and that shrinking Γ can be thought of as cutting corners of the box uncertainty set. Determining an appropriate Γ value without any information on the covariance of the uncertain variables is generally difficult. However, if a dataset is available, one can estimate Γ by adjusting it to create the smallest-volume set that comprises all data points.

Ellipsoidal Uncertainty Set

Alternatively, constructing an uncertainty set can be thought of as constructing a confidence interval like in statistics. An uncertainty set can be defined as an ellipsoid with scaling matrix $\boldsymbol{P}$, center $\boldsymbol{p}$, and radius r:

Definition 10 – Ellipsoidal Uncertainty Set

$$\mathcal{U}_{ellipsoid} \coloneqq \{\boldsymbol{z} \in \mathbb{R}^l \;:\; \|\mathbf{Pz} - \mathbf{p}\|_2 \le r\}.$$

A nice property of the ellipsoidal uncertainty set is that it can easily be fitted to given uncertainty realizations $\boldsymbol{z}^s$, assuming these originate from a multivariate Gaussian distribution. The LP described in Model 10 minimizes r to determine the minimal confidence level of the multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ estimated from S given data points $\{\boldsymbol{z}^s\}_{s=1}^{S}$, with sample mean $\hat{\boldsymbol{\mu}} \in \mathbb{R}^l$ and covariance matrix $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{l \times l}$ containing all S scenarios or data points [156]:

Model 10 – LP Fitting an Ellipsoidal Uncertainty Set Based on Sampled Data

$$\min_{r} \; r$$

s.t:

$$\|(\hat{\boldsymbol{\Sigma}}^{1/2})^{-1}(\boldsymbol{z}^s - \hat{\boldsymbol{\mu}})\|_2 \le r, \quad \forall s \in [S].$$

Unlike the box and budgeted uncertainty sets, the ellipsoidal uncertainty set is non-polyhedral, which means that it cannot be described by finitely many linear equations, such that membership cannot be enforced using an MILP. Instead a Second-Order Cone Programming (SOCP) has to be solved [151].

Data-Driven Uncertainty Sets

While the aforementioned uncertainty sets are conceptually straightforward and allow for reduced conservatism by adjusting parameters such as Γ and r, doing so typically requires domain-specific expertise. Furthermore, these methods rely on the assumption that uncertainty dimensions are independently and symmetrically distributed, which limits their effectiveness in capturing correlations among uncertain variables. When data is available, uncertainty sets for robust optimization can be directly estimated from it, as demonstrated in recent studies [157, 158].

Early works focused on the creation of polyhedral uncertainty sets, such as the study by Shang, Huang, and You [157] employing Kernel-based Support Vector Clustering (K-SVC) and the work by Ning and You [159] combining Principal Component Analysis (PCA) with Kernel Density Estimation (KDE). Although computationally efficient, these methods lacked adaptivity to complex uncertainty distributions. Later, Goerigk and Kurtz [160] employed a Deep Neural Network (DNN) to construct highly compact but non-convex uncertainty sets, demanding high computational resources.

Recently, Li, Yorke-Smith, and Keviczky [158] introduced an alternative approach for constructing uncertainty sets by combining several machine learning techniques. The method begins with Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [124] to filter out low-probability outliers. Next, Gaussian Mixture Models (GMM) [161] are used to form a predefined number of clusters. Within each cluster, PCA is applied to de-correlate features. This results in box-shaped uncertainty sets in the transformed PCA space. The overall uncertainty set is then defined as the union of these individual subsets, which forms a non-convex but computationally tractable structure that can be represented using a limited number of linear constraints. This approach effectively captures complex uncertainty distributions while preserving computational efficiency. A similar methodology can also be used to construct ellipsoidal uncertainty sets that better align with Gaussian-shaped clusters, although this again requires solving an SOCP.

2.5.4. Solution Methods for Adaptive Robust Optimization

Even if the uncertainty set $\mathcal{U}$ is compact and convex, such that it can be substituted with the set of its extreme points, $ext(\mathcal{U})$, the number of scenarios that need to be considered to arrive at a robust feasible solution can grow large when the dimensionality l is high [156]. This is particularly troublesome for ARO, where the number of variables and constraints scales with the number of considered scenarios. Hence, numerous techniques have been developed to approximately solve adaptive robust optimization problems. We distinguish between inner-approximation methods and outer-approximation methods.

Inner Approximation Methods

The optimal second-stage decision can be represented as a function $\boldsymbol{y}(\boldsymbol{z})$, known as a decision rule [156]. A notable example of such an approach is the work by Ben-Tal et al. [162], modelling $\boldsymbol{y}(\boldsymbol{z})$ as an affine function of $\boldsymbol{z}$, whose parameters are learned through co-optimizing them with $\boldsymbol{x}$. However, Jeong et al. [113] reject this approach for the EVRPTW-PR, as the learning of the affine parameters increases the already large number of decision variables, such that the problem easily becomes intractable. Another downside of such an inner approximation technique is that it might not cover all optimal decision rules, such that only a subset of all feasible $\boldsymbol{x}$ is considered.

Outer Approximation Methods

On the other hand, outer approximation methods operate on a superset of the feasible $\boldsymbol{x}$. The most prevalent examples are the Benders decomposition with cutting planes technique, as introduced by Bertsimas et al. [154], and the Column and Constraint Generation (CCG) technique by Zeng and Zhao [155]. Both methods are very similar and decompose the problem into a Restricted Master Problem (RMP) and subproblem (SP). The RMP solves the main optimization problem over a finite subset of scenarios $\bar{\mathcal{U}} \subseteq \mathcal{U}$. The SP is also known as the pessimization or adversary's problem. It aims to find a new scenario q defined by uncertainty realization $\boldsymbol{z}^q \in \mathcal{U} \setminus \bar{\mathcal{U}}$ that would violate the constraints given the current routing solution $\boldsymbol{x}$ for any of the possible second-stage solutions $\boldsymbol{y}^q$ or existing second-stage solution $\boldsymbol{y}^s$ with $\boldsymbol{z}^s \in \bar{\mathcal{U}}$, depending on the formulation. An identified scenario q is added to the RMP after which the procedure is repeated until the SP fails to find a new scenario. This indicates that the RMP provides an optimal first-stage solution $\boldsymbol{x}$ that is robust to all scenarios $\boldsymbol{z} \in \mathcal{U}$.

These methods typically yield robust solutions within a few RMP iterations, avoiding the need to exhaustively enumerate all possible scenarios in the potentially infinite uncertainty set, offering substantial computational advantages [156]. Note that CCG is an extension of the column generation method described in Section 2.3.1 that adds both variables (columns) as well as constraints (rows) to the problem. The major difference between the Benders and CCG methods lies in the type of constraints added to the RMP. In CCG, a copy of the adaptive variables $\boldsymbol{y}$ and the corresponding constraints from the original model are added to the master problem. Conversely, in the Benders decomposition approach, the new constraints define cutting planes that eliminate infeasible solutions.

2.6. Uncertainties in Electric Vehicle Fleet Planning

Having examined how uncertainties can be modelled in optimization problems, and the methods available to address them, we now shift our focus to related works from the literature to explore the types of uncertainties encountered as well as the chosen solution methods.

Stochastic VRPs

Uncertainty in conventional VRPs was initially addressed through the Stochastic Vehicle Routing Problem (SVRP). Most formulations of this problem focus on uncertainty in customer locations, demand quantities, or traversal and service durations [163, 164]. A foundational contribution by Laporte, Louveaux, and Mercure [165] introduced a VRPTW model incorporating stochastic traversal and service durations through chance constraints and a recourse strategy to repair solutions. This approach has since become a standard in the field, inspiring a wide range of subsequent research. Since we do not focus on stochastic optimization in our work, we refer to surveys by Gendreau, Jabali, and Rei [163] and Oyola, Arntzen, and Woodruff [164] for a comprehensive overview of recent developments in stochastic VRPs.

Robust VRPs

More recently, robust optimization techniques have also been applied to conventional VRPs under uncertainty. Notably, Sungur, Ordónez, and Dessouky [166] and Gounaris, Wiesemann, and Floudas [167] proposed branch-and-cut algorithms for solving robust CVRPs with uncertain customer demands. Expanding on this, Hu et al. [168] addressed the combination of demand and traversal duration uncertainty using a branch-and-price-and-cut approach for the VRPTW. Similarly, Hoogeboom et al. [169] developed a robust VRPTW model that accounts for uncertainty in both traversal and service durations, using a branch-and-cut scheme to solve the problem exactly. Adaptive robust optimization approaches using Benders decompositions have been employed by Jaillet, Qi, and Sim [170] and Lei, Lin, and Miao [171] to handle uncertainty in traversal durations and customer demands, respectively.

EVRPs under Customer Uncertainty

One of the earliest contributions involving uncertain EVRPs is by Schiffer and Walther [172], who addressed uncertainties in customer information such as spatial distribution, demand, and service time windows, through a scenario-based robust optimization approach. Their solution methodology combined ALNS with dynamic programming. Zhang, Chen, and Zhang [173] explored a setting with stochastic customer demands, while omitting time window considerations. They proposed a hybrid metaheuristic that integrates VNS with an evolutionary algorithm.

EVRPs under Energy Consumption Uncertainty without Timing Considerations

Pelletier, Jabali, and Laporte [4] considered the energy consumption of EVs as the source of uncertainty in the EVRP and solved the problem by integrating a robust optimization model under various uncertainty sets with a two-phase metaheuristic involving LNS to sketch solutions and LS to refine them. Basso, Kulcsár, and Sanchez-Diaz [7] predicted the time-dependent energy consumption by applying a probabilistic Bayesian machine learning technique and integrated chance-constrained partial recharging into a model that was solved exactly using CPLEX [67]. Although traversal durations and service durations are included to model precedence constraints, time window constraints are not incorporated. Basso et al. [128] applied safe reinforcement learning to deal with stochastic customer requests and stochastic energy consumption. Kim, Park, and Jeong [174] proposed a robust optimization model to address uncertainty in energy consumption by employing edge segmentation. In this approach, a separate uncertainty budget is used for each edge in the transport network, which is divided into multiple segments competing for the overall uncertainty budget, effectively reducing the model's conservatism. The problem was solved using a GA. Amiri, Zolfagharinia, and Amin [73] tackled the EVRP with HGEVs under uncertain energy consumption. Energy consumption was modelled using a box uncertainty set that included both weight-dependent and weight-independent variables that were applied uniformly across all edges. To manage the trade-off between energy efficiency and route robustness, they employed multi-objective optimization techniques, combining a non-dominated sorting GA with ALNS.

EVRPs under Energy Consumption Uncertainty with Timing Considerations
While the previously mentioned works on energy consumption uncertainty did not consider the temporal dimension, some other studies did incorporate it, albeit by treating timing-related parameters as deterministic. Wang et al. [69] and Soysal, Çimen, and Belbağ [68] both solved a Pickup and Delivery Problem (PDP) with electric vehicles and time windows under energy consumption uncertainty. Wang et al. [69] estimated an ellipsoidal uncertainty set using a probabilistic method for modelling the energy consumption of HGEVs and solved the corresponding SOCP using Gurobi [66]. Soysal, Çimen, and Belbağ [68] used artificial data and chance constraints to formulate a model that was solved exactly using CPLEX. Basso et al. [51] utilized a probabilistic Bayesian machine learning approach to estimate time-dependent energy consumption per edge for the EVRP. While traversal durations were also modelled as time-dependent, they were treated deterministically based on estimated values. Time windows were omitted in the problem formulation. Wang, Zheng, and Zhou [112] employed a combination of ALNS and LS to solve the EVRPTW-PR under budgeted energy consumption uncertainty. Jeong et al. [113] addressed energy consumption uncertainty in the EVRPTW-PR using adaptive robust optimization. They employed a budgeted uncertainty set and tackled the problem using CCG, employing a hybrid VNS-TS metaheuristic in combination with a corridor-based generalized cost function, as introduced by Schiffer and Walther [172], to solve the RMP. The Alternating Direction (AD) algorithm proposed by Konno [175] was used to solve the associated pessimization problem. In a recent study, Damianakis et al. [176] focused on smart charging control strategies and adopted a robust optimization framework with a budgeted uncertainty set to capture operational uncertainties from the perspective of an Energy Management System (EMS) connected to the chargers. Their model incorporates uncertainties in arrival state-of-charge, energy demand, and the arrival and departure times of the electric vehicles.

EVRPs under Time Variability and Uncertainty
Time-dependent waiting durations at recharging stations for the EVRPTW-PR were addressed by Keskin, Laporte, and Çatay [44], who segmented the day into multiple time intervals with varying queue lengths. They applied a queuing model to estimate waiting durations specific to each interval. The routing decisions were tackled using ALNS, whereas charging and scheduling decisions were refined through an MILP model solved with CPLEX. Lera-Romero, Bront, and Soulignac [8] incorporated time-dependent traversal durations and energy consumption into the EVRPTW-PR, treating the parameters for each of the intervals as deterministic and employing an exact branch-cut-and-price algorithm to solve the problem. Keskin, Çatay, and Laporte [111] considered stochastic waiting durations at charging stations, applying ALNS to solve the EVRPTW-PR. They then relied on Monte Carlo Simulation (MCS) to estimate the reliability and the expected cost of the solutions.

2.7. Overview of Closely Related Work

Building on our extensive literature review, we now present an overview of the most related works, highlighting the research gaps our work addresses. Table 2.6 presents a summary of the related literature and our contributions, as stated in Section 1.3.

Addressing Heavy-Duty Electric Vehicle Operations under Uncertainty
While the EVRP has been extensively studied, research specifically targeting heavy-duty electric vehicle operations under uncertainty remains scarce. Notable exceptions include the works of Basso et al. [7, 51], which focus on electric buses, and those by Pelletier, Jabali, and Laporte [4], Wang et al. [69] and Amiri, Zolfagharinia, and Amin [73] which target HGEVs in a North American setting, showing differences in vehicle specifications (e.g., frontal area, mass) and regional factors (e.g., road and traffic conditions) compared to a European setting. No works with realistic case studies involving a European setting with HGEVs could be identified.

Basso et al. [7, 51] employ proprietary, high-fidelity vehicle simulation models validated on Swedish bus operations, which differ significantly from typical HGEV operations, being more urban and stop-intensive. Their problem formulation lacks practical considerations for delivery problems such as transport costs, customer demands, and time windows. While their first work relies on deterministic energy consumption forecasts, their second work utilizes a stochastic optimization approach with chance constraints, which can lead to route plans that may not be feasible under worst-case scenarios.

Wang et al. [69] use realistic vehicle parameters and simulate energy consumption via duty cycles to define an ellipsoidal uncertainty set. However, their model requires solving a second-order cone program, is limited to small fleets (up to 10 vehicles), and is tested on a static static network, where only the demand changed between the instances. Pelletier, Jabali, and Laporte [4] incorporate detailed energy modelling but assume constant velocities, neglecting acceleration and recuperation effects. They also omit charging station visits and time windows. Amiri, Zolfagharinia, and Amin [73] present a comprehensive optimization model but simplify energy consumption using only two uncertain variables applied uniformly across the network, assuming constant velocity.

Employing Scalable Solution Methods for Robust Optimization
The recent work by Wang, Zheng, and Zhou [112] shares methodological similarities with ours, employing a scalable ALNS metaheuristic to solve a robust formulation based on a budgeted uncertainty set for the energy consumption. However, their model assumes unlimited charger availability, overlooks partial recharging, and their evaluation is based on the synthetic examples introduced by Schneider, Stenger, and Goeke [30], which lack any real-world grounding.

Although adaptive robust optimization has proven effective across a broad spectrum of optimization problems [177], including vehicle routing problems [170, 171], its application to EVRPs under uncertainty remains largely unexplored. To date, the only study applying this approach in the EVRP context is by Jeong et al. [113]. This work employs a budgeted uncertainty set and solves the resulting RMP using a VNS-TS metaheuristic, in contrast to the ALNS algorithm used in many other studies. Their pessimization method employs a relaxed version of the problem, which means it does not strictly rule out feasible scenarios from being added to the RMP. This allows for potential enhancements that could limit the number of scenarios introduced. Moreover, their evaluation is again based on modified artificial instances from Schneider, Stenger, and Goeke [30], where expected energy consumption is uniformly assigned across all arcs and deviations are arbitrarily defined, omitting realistic energy modelling. Lastly, a comparison with a static robust optimization technique is missing in their work.

Incorporating the Combination of Time and Energy Uncertainties
To the best of our knowledge, no existing research has jointly addressed the uncertainties in both energy consumption and timing within the EVRP using a robust optimization framework. The study by Lera-Romero, Bront, and Soulignac [8] incorporates time-varying energy and timing characteristics to account for traffic congestion effects. However, it does so by dividing the planning horizon into a limited number of intervals with fixed, known parameters, capturing time variability rather than true uncertainty. Additionally, their energy consumption model excludes recuperation effects, and the instances used are adapted from those proposed by Desaulniers et al. [50], which lack real-world grounding. Basso et al. [51] apply a stochastic optimization approach that includes time-dependent velocity predictions for each edge, influencing the stochastic energy consumption function. Yet, these predictions are precomputed and treated as deterministic inputs in the optimization model. Keskin, Çatay, and Laporte [111] do consider true time uncertainty through stochastic waiting durations at charging stations, but rely on Monte Carlo simulation, which does not provide formal robustness guarantees.

Enforcing Realistic Constraints on Charging Sessions
Furthermore, there a no works that account for charger availability beyond fixed time windows. Most EVRP models even overlook the initial charging session at the depot, assuming that vehicles start their routes with fully charged batteries. Although the recent work by Saklaoui et al. [178] explicitly models the initial charging phase and limits the total power drawn by simultaneously charging vehicles, focusing on the integration with the depot's energy management system, it does not incorporate any form of uncertainty and omits energy consumption modelling. A similar limitation applies to the study by Keskin and Çatay [31], which presents a detailed EVRPTW-PR model and utilizes a scalable ALNS-based solution approach, yet also neglects any uncertainty.

Closing the Gaps

As stated in Section 1.3, our work addresses the identified research gaps by proposing an ARO framework that accounts for both time and energy consumption uncertainty. We model energy consumption as a function of time-domain uncertainty, with parameter estimation performed via MCS. To improve the computational efficiency of the CCG approximation method for ARO, we introduce a refined pessimization strategy that selectively incorporates only provably feasible scenarios into the master problem.

To ensure scalability to larger problem instances, we develop a multi-scenario ALNS metaheuristic. Unlike previous corridor-based methods that rely on complex cost functions and implicit second-stage modelling, our approach explicitly integrates second-stage decisions and employs intelligent operators to guide the search. This shifts the complexity from the objective function to the search process, enabling more informed solution exploration.

Furthermore, we incorporate realistic operational constraints often overlooked in prior studies, including charger availability, partial recharging, and initial depot charging. Our evaluation is grounded in a realistic though synthetic European context, using realistic HGEV specifications and operational parameters, contrasting with prior works that often rely on fully synthetic or North American datasets. Table 2.6 compares our approach with prior work.

Reference	**Time**	**Energy**	**Depot**	**Source of**	**Modelling**	**Solution**
	Windows	**Modelling**	**Charging**	**Uncertainty**	**Approach**	**Approach**
Soysal, Çimen, and Belbağ [68]				Energy	SO MILP	Exact
Kim, Park, and Jeong [174]				Energy	RO MILP	GA
Basso et al. [51]		✓		-	MILP	Exact
Basso, Kulcsár, and Sanchez-Diaz [7]		✓		Energy	SO MILP	Exact
Wang et al. [69]	✓	✓		Energy	RO SOCP	Exact
Pelletier, Jabali, and Laporte [4]		✓		Energy	RO MILP	LNS-LS
Amiri, Zolfagharinia, and Amin [73]	✓	✓		Energy	RO MILP	ALNS-GA
Wang, Zheng, and Zhou [112]	✓			Energy	RO MILP	ALNS-LS
Lera-Romero, Bront, and Soulignac [8]	✓	✓		-	MILP	Exact
Keskin, Çatay, and Laporte [111]	✓			Time	MILP	ALNS + MCS
Hoogeboom et al. [169]	✓			Time	RO MILP	Exact
Hu et al. [168]	✓			Demand + Time	RO MILP	Exact
Saklaoui et al. [178]	✓		✓	-	MILP	GA
Jeong et al. [113]	✓			Energy	ARO MILP	VNS-TS (RMP) + AD (SP)
This work	✓	✓	✓	Time + Energy	(A)RO MILP	ALNS (RMP) + Exact (SP) + MCS

Table 2.6: Summary of literature review

3

Methodology

This chapter describes the methodological components underlying CERT$\mathcal{E}$ROUTE, addressesing the following two research sub-questions:

Research Sub-question 2

How can uncertainties in time and energy consumption be formally represented and incorporated into an optimization model for HGEV charging scheduling and routing?

Specifically, Section 3.1 describes the HGEV charging scheduling and routing problem under time and energy consumption uncertainty that we aim to address with this work. Section 3.2 details an adaptive robust optimization model for this problem that accounts for the uncertainties in the time and energy domain by using uncertainty sets in the time domain and establishing a mapping to the energy domain.

Research Sub-question 3

Which solution methods can be employed to robustly solve the formulated optimization problem, while ensuring computational tractability?

Section 3.3 introduces a robust solution method for the optimization problem based on the column and constraint generation framework, including a novel pessimization method that allows for an efficient exploration of the search space. Section 3.4 details a metaheuristic approach that can embedded into this procedure to ensure scalability to larger instances. Lastly, Section 3.5 outlines how the previously introduced methodological components are synthesized into a suite of solvers that constitute CERT$\mathcal{E}$ROUTE.

3.1. Problem Description

In this section, we introduce the Heavy Goods Electric Vehicle Fleet Planning Problem under Time and Energy Consumption Uncertainty (HGEV-FPP-TECU). We begin by outlining the scope of the problem, providing a formal definition of an HGEV and detailing the key assumptions that underpin our problem formulation. Subsequently, we offer a formal problem definition and introduce the associated notation. Lastly, we show that the problem is $\mathcal{NP}$-complete, establishing its computational complexity.

Formal definition of an HGEV

We refrain from using the widespread but ambiguous term Heavy-Duty Electric Vehicle (HDEV), which can also be associated with electric buses, for which operating conditions differ significantly. Instead, we define the term HGEV in alignment with the EU's definition of a Heavy Goods Vehicle (HGV) [179]:

Definition 11 – Heavy Goods Electric Vehicle (HGEV)

Consider a vehicle suitable for the transport of freight/goods over land, often referred to as lorry or (semi-)truck. We define such a vehicle to be a Heavy Goods Electric Vehicle (HGEV) if both of the following conditions are met:

> 1. The vehicle's total mass including the cargo, also referred to as Gross Combination Mass (GCM), exceeds 3,500 kg.
> 2. The vehicle is equipped with a fully battery-electric drive.

This excludes combustion engine HGVs as well as battery electric passenger cars, buses, and delivery vans from the scope of our research, although a similar methodology might be applicable to those vehicle types.

Assumptions
We agreed upon the following set of assumptions for the HGEV-FPP-TECU with Shell:

1. A fixed-size heterogeneous fleet of HGEVs is given, each defined by physical properties, load capacity, battery capacity, maximum charging rate, per-kilometre cost, and hourly operating cost.
2. Fleet operations are planned one day in advance and comprise a single day.
3. Driving time regulations do not have to be accounted for explicitly.
4. All vehicles are dispatched from a single depot.
5. A fixed set of customers is given, each with a known location and delivery demand.
6. Each customer is served by exactly one vehicle.
7. Delivery demands are expressed in terms of mass.
8. A fixed set of private chargers is located at the depot.
9. A fixed set of public or semi-public chargers is located outside the depot.
10. Each vehicle either performs a single tour starting and ending at the depot or remains at the depot.
11. Each vehicle's load must not exceed its load capacity.
12. Each vehicle's battery state of charge (SoC) must remain within its usable range.
13. Each vehicle starts its tour with the SoC at the lower bound of its usable range.
14. Each tour begins with recharging at a depot charger.
15. Each location is reachable from every other location via a directed road connection.
16. Each location has a fixed service time window during which loading/unloading/charging may begin.
17. The service end time at the depot specifies the latest time by which vehicles must return to the depot.
18. Vehicles arriving before a location's service window may wait; late vehicles cannot be serviced.
19. Vehicles may visit a charger between any two locations.
20. Each charger may be visited by zero, one, or multiple vehicles, with a per-vehicle bound on the number of visits.
21. Charging sessions at a charger may not overlap.
22. Each vehicle may be partially recharged at a charger.
23. Each road connection has a known, fixed traversal distance.
24. Each road connection has an uncertain but bounded traversal duration, identical for all vehicles.
25. Energy consumption is uncertain but bounded and depends on the vehicle, road connection, and load level.
26. Service durations for loading and unloading are uncertain but bounded and independent of demand.
27. Multiple vehicles may be loaded simultaneously at the depot.
28. Waiting durations before loading, unloading, or recharging are uncertain but bounded.
29. Charging rates are uncertain but bounded, depend on the vehicle–charger pair, and are independent of SoC.
30. The effect of battery degradation is insignificant.
31. Energy costs per kWh are known and fixed per charger.
32. Distance costs are known and fixed per driven kilometre.
33. Operating costs are counted per minute from the end of the initial depot charging session until the final arrival at the depot.
34. The objective is to minimize total operational cost.

Justification

Most of the assumptions are adapted from classical EVRP literature, including the studies by Schneider, Stenger, and Goeke [30], Keskin and Çatay [31], Schiffer and Walther [32], and Hiermann et al. [47]. However, these works generally assume that vehicles leave the depot with fully charged batteries. In contrast, our approach explicitly models the charging process at the depot, recognizing its importance for co-optimization with energy management. This modelling choice aligns with more recent works such as those by Klein and Schiffer [180] and Saklaoui et al. [178], which focus more on charging schedule optimization. Additionally, we assume that charging sessions on the same charger cannot overlap, diverging from conventional assumptions that chargers are perpetually available.

The assumption that the charging rate is independent of the state of charge simplifies reality, where charging rates typically decline beyond an SoC of 80% [54]. However, this simplification is less problematic for HGEVs than for passenger cars and delivery vans due to their larger battery capacities. By incorporating uncertain charging rates, we can implicitly reflect the slower rates at a higher SoC, setting the lower bound of the uncertainty range accordingly. Since our framework focuses on day-to-day route planning, we do not account for long-term battery degradation. However, this effect can be incorporated over time by applying established degradation models (see Ruiz, Damianakis, and Mouli [181] for a recent review) to adjust battery capacity parameters and refine the uncertainty bounds of the charging rates before applying the optimization model.

To narrow the scope of the problem, driving regulations have been excluded. We realize that this is a limitation of our work. Nonetheless, total route durations can easily be constrained to comply with these regulations. We assume that the mandatory 45-minute break after 4.5 hours of driving [182], which is allowed to be split into two shorter intervals, can always be aligned with unloading or charging operations. Lastly, while our study focuses on a single-depot scenario, we believe that the framework can be readily extended to accommodate multiple depots [17] or backhaul operations [18].

Mathematical Definition

We define the HGEV-FPP-TECU as in Definition 12, building upon the EVRP formulations and notation introduced in Section 2.2. An overview of the introduced notation is given in Table 3.1.

Definition 12 – HGEV-FPP-TECU

Network:
Consider a transport network modelled as a complete directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{0\} \cup \mathcal{C} \cup \mathcal{F}' \cup \{N+1\}$ represents the set of vertices (locations), with $\mathcal{C} := \{1, 2, \ldots, N\}$ representing the set of customers i having known and fixed delivery demand q_i. Let $\mathcal{F}' := \{N+2, \ldots, N+1+NF, N+2+NF, \ldots, N+1+n_{visits}K+NF\}$ represent a set of dummy vertices to allow $n_{visits}K$ visits to each of the NF chargers f in set $\mathcal{F} := \{N+2, \ldots, N+1+NF\}$. Let vertices 0 and $N+1$ represent the exit and entrance of the central depot, respectively. Let $\mathcal{F}'' \subset \mathcal{F}'$ represent the set of chargers located at the depot. For convenience, let $\mathcal{V}' := \mathcal{C} \cup \mathcal{F}'$, $\mathcal{V}'_0 := \mathcal{V}' \cup \{0\}$, and $\mathcal{V}'_{N+1} := \mathcal{V}' \cup \{N+1\}$. Let the function $Parent(o)$ map every dummy vertex $o \in \mathcal{F}'$ to its corresponding base charger $f \in \mathcal{F}$. Let the set of edges $\mathcal{E} := \{(i,j) \mid i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1},\ i \neq j\}$ contain all possible road connections (i,j), each associated with a known and fixed traversal distance d_{ij}.

Vehicles:
Let $\mathcal{K}$ represent a fleet of K heterogeneous HGEVs k with known and fixed individual load capacity C_k, battery capacity Q_k and empty curb mass m_c^k. Let r_k be a decision variable tracking whether vehicle k is deployed or not and let decision variables x_{ij}^k indicate whether an edge $(i,j) \in \mathcal{E}$ is traversed by vehicle k or not. Let u_j^k be the decision variable tracking the load quantity of vehicle k upon arrival at destination $j \in \mathcal{V}'_{N+1}$.

Uncertainty:
Let $\mathcal{S}$ be a given set of scenarios s, each giving rise to different realizations of the uncertain

parameters in the time and energy domains.

Time:
Let t^{s}_{ij} denote the traversal duration associated with every edge $(i, j) \in \mathcal{E}$ under scenario $s \in \mathcal{S}$. Furthermore, let e_i and l_i represent the earliest and latest possible service start times for every location $i \in \mathcal{V}$, respectively. Let w^{s}_{j} denote the waiting or queuing duration before service start associated with every location $j \in \mathcal{V}$ and ν^{s}_{i} the loading/unloading duration for every location $i \in \mathcal{C} \cup \{0\}$. Let τ^{s}_{i} be a decision variable that tracks the actual service start time for every location $i \in \mathcal{V}$. Let g^{ks}_{f} be the inverse charging rate that determines the charging duration per kWh associated with the combination of every charger $f \in \mathcal{F}'$ and vehicle $k \in \mathcal{K}$. Let τ^{s}_{k} be a decision variable representing the start time of vehicle k's route, corresponding to the moment it is loaded at the depot. Let δ^{s}_{k} be a decision variable denoting the time it departs from its initial charging session at one of the depot chargers $f \in \mathcal{F}''$, and let decision variable σ^{s}_{k} indicate the time it returns to the depot, marking the end of its route.

Energy:
Let decision variables y^{ks}_{i} and Y^{ks}_{f} jointly track the battery SoC of vehicle $k \in \mathcal{K}$, measuring it upon arrival at destination $i \in \mathcal{V}'$ and upon departure from charger $f \in \mathcal{F}'$, respectively, so that the amount of recharged energy at charger f is given by $Y^{ks}_{f} - y^{ks}_{f}$. Let h^{ks}_{ij} be the energy consumption for every traversed edge $(i, j) \in \mathcal{E}$ in scenario $s \in \mathcal{S}$, by which the battery SoC of vehicle k is decreased upon traversal.

Costs:
Let c^{k}_{km} denote the distance cost per driven kilometre for vehicle $k \in \mathcal{K}$, c^{f}_{kWh} the energy cost per kWh at charger $f \in \mathcal{F}'$ and let c_{min} be the time cost per minute that is charged for the duration $\sigma_k - \delta_k$ that requires a driver for vehicle k.

Objective:
The HGEV-FPP-TECU is defined as constructing a set of routes $\mathcal{R} \coloneqq \{R_k \mid R_k \subset \mathcal{E}, k \in \mathcal{K}\}$ such that every R_k is a path starting at vertex 0 and ending at vertex $N+1$ and every customer $i \in \mathcal{C}$ is part of exactly one route R_k. The sum of the customer demands q_i assigned to a vehicle k should respect the load capacities C_k. Furthermore, $\mathcal{R}$ should be robust, meaning that for all scenarios $s \in \mathcal{S}$, timing and charging decisions can be made such that the vehicles do not run out of battery, time windows are respected, and charging sessions do not overlap. Lastly, the worst-case total operational cost c_{total} over all scenarios $s \in \mathcal{S}$, given by summing the distance, energy and time costs associated with $\mathcal{R}$, should be minimized.

Sets	
0, $N+1$	Exit and entrance vertices of the central depot
$\mathcal{C}$	Set of customers vertices $\mathcal{C} \coloneqq \{1, 2, \dots, N\}$
$\mathcal{F}$	Set of charger vertices $\mathcal{F} : \{N+2, \dots, N+1+NF\}$
$\mathcal{F}'$	Set of dummy charger vertices $\mathcal{F}' \coloneqq \{N+2, \dots, N+1+NF, N+2+NF, \dots, N+1+n_{visits}+NF\}$ to allow $n_{visits}K$ visits to every $f \in \mathcal{F}$
$\mathcal{F}''$	Set of depot charger vertices $\mathcal{F}'' \subset \mathcal{F}'$
$\mathcal{V}'$	Set of customer and charger vertices $\mathcal{V}' \coloneqq \mathcal{C} \cup \mathcal{F}'$
$\mathcal{V}'_0$	Set of source location vertices $\mathcal{V}'_0 \coloneqq \mathcal{V}' \cup \{0\}$
$\mathcal{V}'_{N+1}$	set of destination location vertices $\mathcal{V}'_{N+1} \coloneqq \mathcal{V}' \cup \{N+1\}$
$\mathcal{V}$	Set of all vertices $\mathcal{V} \coloneqq \{0\} \cup \mathcal{C} \cup \mathcal{F}' \cup \{N+1\}$

$\mathcal{E}$	Set of edges $\mathcal{E} \coloneqq \{(i,j) \mid i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1},\ i \neq j\}$
$\mathcal{K}$	Set of K heterogeneous HGEVs
$\mathcal{R}$	Set of vehicle routes $\mathcal{R} \coloneqq \{R_k \mid R_k \subset \mathcal{E}, k \in \mathcal{K}\}$
$\mathcal{S}$	Set of scenarios representing the uncertainty in the energy and time domains

Functions

$Parent(o)$	Function $\mathcal{F}' \rightarrow \mathcal{F}$ mapping dummy charger $o \in \mathcal{F}'$ to the corresponding base charger $f \in \mathcal{F}$

Decision Variables

r_k	Binary decision variable indicating whether vehicle $k \in \mathcal{K}$ is deployed
x_{ij}^k	Binary decision variable indicating whether edge $(i,j) \in \mathcal{E}$ is included in the route R_k of vehicle $k \in \mathcal{K}$
u_i^k	Continuous decision variable tracking the remaining load quantity of vehicle $k \in \mathcal{K}$ upon arrival at destination $i \in \mathcal{V}$ [in kg]
y_i^{ks}	Continuous decision variable tracking the battery SoC of vehicle $k \in \mathcal{K}$ upon arrival at destination $i \in \mathcal{V}$ under scenario $s \in \mathcal{S}$ [in kWh]
Y_f^{ks}	Continuous decision variable indicating the SoC of vehicle $k \in \mathcal{K}$ on departure from charger $f \in \mathcal{F}'$ under scenario $s \in \mathcal{S}$ [in kWh]
τ_i^s	Continuous decision variable indicating the service start time at location $i \in \mathcal{V}$ under scenario $s \in \mathcal{S}$ [in minutes]
τ_k^s	Continuous decision variable indicating the route start time of vehicle $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$ [in minutes]
δ_k^s	Continuous decision variable indicating the departure time from the initial depot charger under scenario $s \in \mathcal{S}$ [in minutes]
σ_i^s	Continuous decision variable indicating the service end time at location $i \in \mathcal{V}$ under scenario $s \in \mathcal{S}$ [in minutes]
σ_k^s	Continuous decision variable indicating the route end time of vehicle $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$ [in minutes]

Uncertain Parameters

h_{ij}^{ks}	Energy consumption of vehicle $k \in \mathcal{K}$ over edge $(i,j) \in \mathcal{E}$ under scenario $s \in \mathcal{S}$ [in kWh]
t_{ij}^s	Traversal duration associated with edge $(i,j) \in \mathcal{E}$ under scenario $s \in \mathcal{S}$ [in minutes]
ν_i^s	Loading/unloading duration for location $i \in \mathcal{C} \cup \{0\}$ under scenario $s \in \mathcal{S}$ [in minutes]
g_f^{ks}	Inverse charging rate for vehicle $k \in \mathcal{K}$ at charger $f \in \mathcal{F}'$ under scenario $s \in \mathcal{S}$ [in minutes/kWh]
w_i^s	Waiting/queuing duration for location $i \in \mathcal{V}$ under scenario $s \in \mathcal{S}$ [in minutes]

Parameters

q_i	Delivery demand at customer $i \in \mathcal{C}$ [in kg]
d_{ij}	Traversal distance associated with edge $(i,j) \in \mathcal{E}$ [in km]

Q_k	Battery capacity of vehicle $k \in \mathcal{K}$ [in kWh]
C_k	Load capacity of vehicle $k \in \mathcal{K}$ [in kg]
m_c^k	Empty curb mass of vehicle $k \in \mathcal{K}$ [in kg]
e_i	Earliest possible service start time for location $i \in \mathcal{V}$ [in minutes]
l_i	Latest possible service start time for location $i \in \mathcal{V}$ [in minutes]
n_{visits}	Maximum number of visits to the same charger per vehicle [in minutes]
c_{km}^k	Distance cost associated with vehicle $k \in \mathcal{K}$ [in €/km]
c_{kWh}^f	Energy cost associated with charger $f \in \mathcal{F}'$ [in €/kWh]
c_{min}^k	Time cost associated with vehicle $k \in \mathcal{K}$ [in €/minute]
n_{visits}	Number of allowed visits per charger $f \in \mathcal{F}$ per vehicle $k \in \mathcal{K}$

Table 3.1: Notation overview for the HGEV-FPP-TECU

$\mathcal{NP}$-completeness of the HGEV-FPP-TECU

The HGEV-FPP-TECU is an $\mathcal{NP}$-complete problem. This can be readily demonstrated using reduction by restriction. For the sake of brevity, we do not provide the full proof, but the main idea is as follows: starting from an instance of the conventional VRP, which is known to be $\mathcal{NP}$-complete [13], one can construct a corresponding instance of the HGEV-FPP-TECU by assigning energy and timing parameters such that the additional constraints become trivially satisfied. This includes setting the energy consumption and traversal duration to zero over all edges. Such a transformation can be performed in polynomial time of the input size. As a result, solving the HGEV-FPP-TECU allows one to solve the VRP, demonstrating that the former is at least as hard as the latter. Together with the fact that the HGEV-FPP-TECU belongs to the complexity class $\mathcal{NP}$, as its solutions can be verified in polynomial time with respect to the input size, this proves its $\mathcal{NP}$-completeness.

3.2. Formulating a Robust Optimization Model

Due to the lack of available data to estimate the probability distributions necessary for stochastic optimization, and our aim to provide strong guarantees on the feasibility of the generated routes, we adopt a robust optimization approach to solve the HGEV-FPP-TECU. In the following subsections, we define uncertainty sets in the time domain (Subsection 3.2.1), introduce a novel mapping to the energy domain (Subsection 3.2.2), and formulate a linear robust optimization model, consistent with established practices in the EVRP literature (Subsection 3.2.3). The choice for an adaptive formulation of this model is justified in Subsection 3.2.4.

3.2.1. Formulation of the Uncertainty Sets

With a slight abuse of notation, we redefine the scenario set $\mathcal{S}$ as an index set $\mathcal{S}(\mathcal{U})$ over an underlying uncertainty set $\mathcal{U}$. As a starting point for $\mathcal{U}$, we adopt the box uncertainty set $\mathcal{U}_{box}$ from Definition 8, which assumes that each uncertain parameter is independently bounded and that all uncertain parameters may simultaneously attain their worst-case values. The box uncertainty set is suitable when there is insufficient data available to construct more sophisticated uncertainty sets, such as data-driven or ellipsoidal sets. Although conservative, it ensures robustness against all possible realizations within the specified bounds.

However, in practice, it is highly unlikely that all road connections are congested at once or that every customer simultaneously experiences extended service durations. To better reflect realistic conditions and reduce conservatism, we also introduce and experiment with a locally budgeted uncertainty set $\mathcal{U}_{local}$, that assigns a separate uncertainty budget to each type of uncertain parameter, following the approach proposed by Goerigk and Lendl [183]. The separate budgets allow for more flexibility compared to the regular budgeted uncertainty set $\mathcal{U}_{budget}$ from Definition 9.

A key modelling decision is to limit both uncertainty sets to the time domain. Incorporating uncertain energy consumption directly into the uncertainty set would introduce dependencies between the edge-specific traversal time t_{ij}^s and energy consumption h_{ij}^{ks}, which are difficult to enforce and conflict with the independence assumption inherent in the uncertainty sets. Instead, we establish a mapping from the time domain to the energy domain by expressing h_{ij}^{ks} as a function of traversal duration t_{ij}^{s} and vehicle mass $m_c^k + u_j^k$, as detailed in Subsection 3.2.2.

Box Uncertainty Set

In the box uncertainty set, each uncertain parameter is assumed to lie within a symmetric interval centered at its nominal value. This introduces the following additional parameters for the HGEV-FPP-TECU:

Parameter	Description	Indices
t_{ij}^{nom}	Nominal traversal duration	$i \in \mathcal{V}_0'$, $j \in \mathcal{V}_{N+1}'$
t_{ij}^{dev}	Maximum deviation in traversal duration	$i \in \mathcal{V}_0'$, $j \in \mathcal{V}_{N+1}'$
ν_i^{nom}	Nominal service duration	$i \in \mathcal{C} \cup \{0\}$
ν_i^{dev}	Maximum deviation in service duration	$i \in \mathcal{C} \cup \{0\}$
$g_f^{k(\text{nom})}$	Nominal inverse charging rate	$f \in \mathcal{F}'$, $k \in \mathcal{K}$
$g_f^{k(\text{dev})}$	Maximum deviation in inverse charging rate	$f \in \mathcal{F}'$, $k \in \mathcal{K}$
w_j^{nom}	Nominal waiting duration	$j \in \mathcal{V}$
w_j^{dev}	Maximum deviation in waiting duration	$j \in \mathcal{V}$

Table 3.2: Parameters used to define the uncertainty set

Using these parameters, we redefine the box uncertainty set $\mathcal{U}_{box}$ as follows:

Definition 13 – Box Uncertainty Set

Let uncertainty set $\mathcal{U}_{box} \subset \mathbb{R}^l$ be indexed by $s \in \mathcal{S}(\mathcal{U}_{box})$ and defined over the uncertainty vector

$$\boldsymbol{z}^s = \begin{bmatrix} \boldsymbol{t}^s \\ \boldsymbol{\nu}^s \\ \boldsymbol{g}^s \\ \boldsymbol{w}^s \end{bmatrix}, \quad \text{with dimension } l = l_1 + l_2 + l_3 + l_4,$$

where

- $\boldsymbol{t}^s \in \mathbb{R}^{l_1}$: traversal durations $t_{ij}^s \in [t_{ij}^{\text{nom}} \pm t_{ij}^{\text{dev}}]$, for $i \in \mathcal{V}_0'$, $j \in \mathcal{V}_{N+1}'$,
- $\boldsymbol{\nu}^s \in \mathbb{R}^{l_2}$: service durations $\nu_i^s \in [\nu_i^{\text{nom}} \pm \nu_i^{\text{dev}}]$, for $i \in \mathcal{C} \cup \{0\}$,
- $\boldsymbol{g}^s \in \mathbb{R}^{l_3}$: inverse charging rates $g_f^{ks} \in [g_f^{k(\text{nom})} \pm g_f^{k(\text{dev})}]$, for $f \in \mathcal{F}'$, $k \in \mathcal{K}$,
- $\boldsymbol{w}^s \in \mathbb{R}^{l_4}$: waiting durations $w_j^s \in [w_j^{\text{nom}} \pm w_j^{\text{dev}}]$, for $j \in \mathcal{V}$.

Let z_i^{nom} and z_i^{dev} denote the nominal value and maximum deviation of the i-th component of $\boldsymbol{z}^s$, respectively. The following uncertainty set now correspond to Definition 8:

$$\mathcal{U}_{box} := \left\{ \boldsymbol{z}^s \in \mathbb{R}^l \;:\; \forall i \in [l],\ z_i \in [z_i^{\text{nom}} \pm z_i^{\text{dev}}] \right\}.$$

We define the nominal-scenario to be described by the vector

$$\boldsymbol{z}^{nom} = \begin{bmatrix} z_1^{nom} \\ z_2^{nom} \\ \dots \\ z_l^{nom} \end{bmatrix}.$$

A nice property of $\mathcal{U}_{box}$ is that it also yields a unique worst-case scenario. This is because, for all its uncertain parameters, higher values correspond to more challenging real-world conditions. Hence, the worst-case scenario can be defined by the vector in which all uncertain parameters simultaneously attain their maximum values, given by

$$\boldsymbol{z}^{worst} = \begin{bmatrix} z_1^{nom} + z_1^{dev} \\ z_2^{nom} + z_2^{dev} \\ \cdots \\ z_l^{nom} + z_l^{dev} \end{bmatrix}.$$

Locally Budgeted Uncertainty Set
We introduce the following additional parameters to the HGEV-FPP-TECU to budget the uncertainty associated with each of the uncertain parameter types:

Parameter	Description
Γ_t	Uncertainty budget for traversal durations
Γ_ν	Uncertainty budget for service durations
Γ_g	Uncertainty budget for inverse charging rates
Γ_w	Uncertainty budget for waiting durations

Table 3.3: Parameters used to define the locally budgeted uncertainty set

Focusing on scenarios where uncertainty leads to more adverse real-world conditions, we account only for deviations above the nominal values in the uncertainty budget. Using the definition for the box uncertainty set, we define the locally budgeted uncertainty set $\mathcal{U}_{local}$ as follows:

Definition 14 – Locally Budgeted Uncertainty Set

Consider the partition

$$\mathcal{P} = \{(\mathcal{I}_\nu, \Gamma_\nu), (\mathcal{I}_t, \Gamma_t), (\mathcal{I}_g, \Gamma_g), (\mathcal{I}_w, \Gamma_w)\},$$

where each $\mathcal{I}_\bullet \subseteq [l]$ denotes the index set corresponding to one of the components of $\boldsymbol{z}^s \in \mathcal{U}_{box}$.

Then the locally budgeted uncertainty set $\mathcal{U}_{local} \subseteq \mathcal{U}_{box}$ is defined as

$$\mathcal{U}_{local} := \left\{ \boldsymbol{z}^s \in \mathcal{U}_{box} \ : \ \forall (\mathcal{I}_j, \Gamma_j) \in \mathcal{P}, \ \sum_{i \in \mathcal{I}_j} \frac{z_i^s - z_i^{\text{nom}}}{z_i^{\text{dev}}} \leq \Gamma_j \right\}.$$

3.2.2. Modelling Energy Consumption

Recall from Section 2.4 that accurately modelling the energy consumption of HGEVs is inherently complex due to its dependence on various dynamic and interrelated factors. Following the approaches of Wang et al. [69] and Goeke and Schneider [48], we model the energy consumption as being linear in the distance and dependent on the vehicle's total mass.

The energy consumption also depends on the effects of velocity and acceleration, as per the longitudinal vehicle dynamics model described in Model 5. However, we do not incorporate the longitudinal dynamics model directly into our optimization problem. It includes a quadratic velocity term, which, being inversely related to the uncertain traversal duration t_{ij}^s, would render the model non-linear.

To preserve linearity, while implicitly incorporating the effect of velocity, we redefine the uncertain energy consumption h_{ij}^{ks} for each vehicle $k \in \mathcal{K}$ on edge $(i, j) \in \mathcal{E}$ in scenario $s \in \mathcal{S}(\mathcal{U})$ as a function of the traversal distance d_{ij}, total vehicle mass $m(u_j^k) := u_j^k + m_c^k$ and uncertain traversal duration t_{ij}^s:

Definition 15 – Energy Consumption Function

$$h(d_{ij}, t_{ij}^s, m(u_j^k)) := \left(\beta_{ij}^k + \alpha_{ij}^k m(u_j^k) + \gamma_{ij}^k \frac{t_{ij}^s}{d_{ij}} + \kappa_{ij}^k m(u_j^k) \frac{t_{ij}^s}{d_{ij}} \right) \psi_{ij}^k d_{ij}, \tag{3.1}$$

where

- $\frac{t_{ij}^s}{d_{ij}}$ denotes the inverse velocity [in min/km],
- $\beta_{ij}^k \in \mathbb{R}$ denotes the intercept term,
- $\alpha_{ij}^k \in \mathbb{R}$ denotes the mass coefficient,
- $\gamma_{ij}^k \in \mathbb{R}$ denotes the inverse velocity coefficient,
- $\kappa_{ij}^k \in \mathbb{R}$ denotes the interaction coefficient,
- ψ_{ij}^k denotes the robustness multiplier.

Note that the interaction term contains a product between the decision variable u_j^k and the uncertain parameter t_{ij}^s, which may initially suggest a non-linearity. However, as detailed later, we adopt a two-phase pessimization approach: during the minimization phase, decision variables are adjusted while uncertain parameters are held fixed, and during the maximization phase, uncertain parameters vary while decision variables remain constant. Since only one set of variables changes at a time, the interaction term preserves linearity within each phase. The preserved linearity allows for rapid pessimization through solving linear programs, for which highly efficient solution methods are included in contemporary off-the-shelf solvers.

The coefficients β_{ij}^k, α_{ij}^k, $\gamma_{ij}^k, \kappa_{ij}^k$, and ψ_{ij}^k are assumed to be pre-computed and to represent the nominal scenario with respect to time-independent uncertainties in the energy domain. We compute these coefficients using a combination of simulation and function approximation techniques, as further discussed in Subsection 4.2.1. The robustness multiplier ψ_{ij}^k can be used to achieve different levels of robustness against energy consumption uncertainty, with $\psi_{ij}^k = 1$ corresponding to nominal-value optimization. Consequently, the remaining sources of variability in the energy domain are limited to changing mass levels and traversal duration uncertainty in the time domain.

3.2.3. Formulation of a Robust Mixed Integer Linear Program

Next, we formulate a linear optimization model for the HGEV-FPP-TECU, following standard practices in EVRP literature. In theory, this model can be solved to optimality using off-the-shelf solvers such as Gurobi [66] and CPLEX [67]. However, in practice, more advanced methods are typically required to maintain the tractability of the problem. In the formulation presented in Model 11, we anticipate the adaptive robust optimization approach via column and constraint generation by distinguishing between adaptive and non-adaptive variables, introducing scenario-specific copies of the non-adaptive variables, and formulating constraints across all scenarios within the uncertainty set. This modelling choice is further motivated in the next subsection.

To promote clarity, the non-adaptive or first-stage decision variables are indicated in purple, the adaptive or second-stage decision variables in lightblue, and the uncertain parameters in red. We employ a constant M, also known as the big-M, that is conditionally subtracted or added to equations to make the associated constraints trivially satisfied, allowing for conditional constraints to be modelled.

Model 11 – ARO MILP Formulation for the HGEV-FPP-TECU

Objective:
Minimize the total worst-case operational cost

$$c_{total}^* = \min \quad (\sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{V}_0'} \sum_{j\in\mathcal{V}_{N+1}'} x_{ij}^k d_{ij} c_{km}^k + \max_{s\in\mathcal{S}(\mathcal{U})} \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}'} (Y_f^{ks} - y_f^{ks}) c_{kWh}^f + \sum_{k\in\mathcal{K}} (\sigma_k^s - \delta_k^s) c_{min}^k), \tag{3.2}$$

s.t:

Route and Load Level Constraints:
Every customer is visited exactly once:

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{V}_{N+1}',\, j\neq i} x_{ij}^k = 1 \quad \forall i \in \mathcal{C}. \tag{3.3}$$

Dummy chargers are visited at most once:

$$\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{V}'_{N+1},\, j\neq f} x^k_{fj} \leq 1 \quad \forall f \in \mathcal{F}'. \tag{3.4}$$

Vehicles leave the depot at most once:

$$\sum_{j\in\mathcal{V}'_{N+1}} x^k_{0j} \leq 1 \quad \forall k \in \mathcal{K}. \tag{3.5}$$

Flow is conserved at intermediate stops:

$$\sum_{i\in\mathcal{V}'_0,\, i\neq j} x^k_{ij} = \sum_{i\in\mathcal{V}'_{N+1},\, i\neq j} x^k_{ji} \quad \forall j \in \mathcal{V}',\, \forall k \in \mathcal{K}. \tag{3.6}$$

Load levels decrease upon departure from locations with demand:

$$u^k_j \leq u^k_i - q_i x^k_{ij} + C_k(1 - x^k_{ij}) \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, i \neq j. \tag{3.7}$$

The maximum number of visits to the same charger per vehicle is respected:

$$\sum_{i\in\mathcal{V}'_0}\ \sum_{o\in\mathcal{F}',\, Parent(o)=f} x^k_{io} \leq n_{visits} \quad \forall f \in \mathcal{F},\, \forall k \in \mathcal{K}. \tag{3.8}$$

Vehicle deployment is tracked:

$$r_k \geq x^k_{ij} \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1}. \tag{3.9}$$

Initial load levels correspond to the summed location demand over the route:

$$u^k_0 = \sum_{i\in\mathcal{V}'_0}\sum_{j\in\mathcal{V}'_{N+1},\, j\neq i} x^k_{ij} q_j. \tag{3.10}$$

The summed demand of subsequent locations do not exceed a vehicle's capacity:

$$q_i + q_j \leq C_k - (1 - x^k_{ij})M. \tag{3.11}$$

Self loops are prevented:

$$x^k_{ij} = 0 \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, Parent(i) = Parent(j). \tag{3.12}$$

Every deployed vehicle starts with an initial charging session at the depot:

$$\sum_{f\in\mathcal{F}''} x^k_{0f} \geq r_k \quad \forall k \in \mathcal{K}. \tag{3.13}$$

Timing and Charging Constraints:
Battery initial SoC is 0:

$$y^{ks}_0 = 0 \quad \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.14}$$

Battery SoC when visiting a charger is non-decreasing:

$$y^{ks}_f \leq Y^{ks}_f \quad \forall f \in \mathcal{F}',\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.15}$$

Departure time at customers is based on the service duration:

$$\sigma^s_i \geq \tau^s_i + \nu^s_i \quad \forall i \in \mathcal{C},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.16}$$

Departure time at chargers is based on the charging duration:

$$\sigma^s_f \geq \tau^s_f + g^{ks}_f(Y^{ks}_f - y^{ks}_f) \quad \forall f \in \mathcal{F}',\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.17}$$

Charging sessions at the same charger do not overlap:

$$\sigma_f^s \leq \tau_o^s, \quad \forall f,\, o \in \mathcal{F}',\, \forall s \in \mathcal{S}(\mathcal{U}),\, o > f,\, Parent(f) = Parent(o) \tag{3.18}$$

Battery SoC decays over traversed edges:

$$y_j^{ks} \leq y_i^{ks} - h(d_{ij}, t_{ij}^s, m(u_j^k))x_{ij}^k + M(1 - x_{ij}^k) \quad \forall i \in \mathcal{C} \cup \{0\},\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}),\, i \neq j. \tag{3.19}$$

Battery is recharged at visited chargers:

$$y_j^{ks} \leq Y_f^{ks} - h(d_{ij}, t_{ij}^s, m(u_j^k))x_{fj}^k + M(1 - x_{fj}^k) \quad \forall f \in \mathcal{F}',\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}),\, f \neq j. \tag{3.20}$$

Arrival time at locations is based on the travel and waiting duration:

$$\tau_j^s \geq \sigma_i^s + t_{ij}^s + w_j^s - (1 - \sum_{k \in \mathcal{K}} x_{ij}^k)M \quad \forall i, j \in \mathcal{V}',\, \forall s \in \mathcal{S}(\mathcal{U}),\, i \neq j. \tag{3.21}$$

Route start times are tracked:

$$\tau_k^s \leq \tau_j^s - w_j^s - t_{ij}^s - \nu_0^s + (1 - x_{0j}^k)M \quad \forall j \in \mathcal{F}'',\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.22}$$

Route end times are tracked:

$$\sigma_k^s \geq \sigma_i^s + t_{i(N+1)}^s - (1 - x_{i(N+1)}^k)M \quad \forall i \in \mathcal{V}',\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.23}$$

Driver start times correspond to departure from the depot charger:

$$\delta_k^s \leq \sigma_f^s + (1 - x_{0f}^k)M \quad \forall k \in \mathcal{K},\, \forall f \in \mathcal{F}'',\, \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.24}$$

The driver start times for undeployed vehicles is fixed to zero:

$$d_k^s \leq r^k M, \quad \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.25}$$

The SoC upon arrival corresponding unvisited locations is fixed to zero:

$$y_j^{ks} \leq (\sum_{i \in \mathcal{V}'_0} x_{ij}^k)M, \quad \forall k \in \mathcal{K},\, \forall j \in \mathcal{V}'_{N+1},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.26}$$

Edges whose traversal always leads to a time violation are avoided:

$$e_i + \nu_i^s + t_{ij}^s + w_j^s \leq l_j + (1 - x_{ij}^k)M \quad \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall k \in \mathcal{K},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.27}$$

Customers whose visit leads to a late arrival at the depot should not be added:

$$e_i + \nu_i^s + t_{ij}^s + w_j^s + \nu_j^s + t_{j(N+1)}^s \leq l_{N+1} + (1 - x_{ij}^k)M \quad \forall i \in \mathcal{V}'_0, \forall j \in \mathcal{V}'_{N+1}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.28}$$

Edges that cannot be traversed with the current battery capacity are avoided:

$$Q_k - h(d_{ij}, t_{ij}^s, m(u_j^k)) \geq (x_{ij}^k - 1)M \quad \forall k \in \mathcal{K},\, \forall i \in \mathcal{V}'_0,\, \forall j \in \mathcal{V}'_{N+1},\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.29}$$

Bounding Constraints:

Vehicles do not arrive at locations before their earliest service start time:

$$e_i \leq \tau_i^s \quad \forall i \in \mathcal{V}',\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.30}$$

Vehicles do not arrive at locations after their latest service start time:

$$\tau_i^s \leq l_i \quad \forall i \in \mathcal{V}',\, \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.31}$$

Vehicles do not leave the depot before its earliest service start time:

$$e_0 \leq \tau_k^s \quad \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.32}$$

Vehicles do not arrive at the depot after its latest service start time:

$$\sigma_k^s \leq l_{N+1} \quad \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.33}$$

Battery SoC remains in the usable range:

$$y_i^{ks} \leq Q_k \quad \forall i \in \mathcal{V},\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.34}$$

$$Y_f^{ks} \leq Q_k \quad \forall f \in \mathcal{F}',\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.35}$$

Load capacity is not exceeded:

$$u_i^k \leq C_k \quad \forall i \in \mathcal{V}',\ \forall k \in \mathcal{K}. \tag{3.36}$$

Domain Constraints for the Decision Variables:

$$x_{ij}^k \in \mathbb{B} \quad \forall i \in \mathcal{V}_0',\ \forall j \in \mathcal{V}_{N+1}',\ \forall k \in \mathcal{K},\ i \neq j, \tag{3.37}$$

$$r_k \in \mathbb{B} \quad \forall k \in \mathcal{K}, \tag{3.38}$$

$$u_i^k \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V}',\ \forall k \in \mathcal{K}, \tag{3.39}$$

$$\tau_i^s,\ \sigma_i^s \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V}',\ \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.40}$$

$$\tau_k^s,\ \sigma_k^s,\ \delta_k^s \in \mathbb{R}_{\geq 0} \quad \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.41}$$

$$Y_f^{ks} \in \mathbb{R}_{\geq 0} \quad \forall f \in \mathcal{F}',\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.42}$$

$$y_i^{ks} \in \mathbb{R}_{\geq 0} \quad \forall i \in \mathcal{V},\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.43}$$

For simplicity, we fix $M = 10^5$ across all constraints, though tighter, constraint-specific values could improve numerical stability and strengthen the model formulation. The arrival time variables τ_i^s and departure time variables σ_i^s are modelled without an index over the vehicles as every customer or charger dummy $i \in \mathcal{V}'$ can be visited by at most one vehicle. While this reduces the number of decision variables, it requires introducing vehicle-specific variables τ_k^s and σ_k^s to capture the start and end times of each route, as all deployed vehicles depart from and return to the depot, so that the depot does not have unique arrival and departure times. The SoC upon departure Y_f^{ks} is only modelled for charger dummies $f \in \mathcal{F}'$, as there are no constraints involving the SoC upon departure from other locations.

The constraints in Eq. (3.10) to Eq. (3.13) and Eq. (3.26) to Eq. (3.29) are redundant but strengthen the model formulation, restricting the search space further so that solutions can be found quicker. These constraints are adapted from the work by Schneider, Stenger, and Goeke [30]. The constraints in Eq. (3.25) are non-redundant and prevent the objective value in Eq. (3.2) from being artificially lowered by assigning large positive values to the otherwise unconstrained variables δ_k^s outside the designated routes R_k. For y_f^{ks} this is already prevented by the non-decreasing constraints from Eq. (3.15). The non-overlapping constraints in Eq. (3.18) also function as symmetry-breaking constraints, as they enforce that charger dummies $f \in \mathcal{F}'$ with lower indices are always assigned to earlier charging sessions than those with higher indices $o > f$.

Technically, the equality constraints, such as in Eq. (3.3), are non-linear. However, since they can be expressed as a pair of $\leq$ and $\geq$ constraints, we treat them as linear constraints in our representation. The constraints are presented in a non-standard form for clarity and readability. However, they can be converted to standard form by rewriting each one as a $\leq$ constraint [29]. This allows us to represent the MILP into a compact matrix form.

3.2.4. Reducing Conservatism with Adaptive Robust Optimization

Recall from Subsection 2.5.2 that the key idea of Adaptive Robust Optimization (ARO) is to divide the decision process in multiple stages to decrease the inherent conservatism of static or single-stage RO models. In a static robust formulation of the problem, routing, load level, timing and charging decisions are all made simultaneously, accounting for every possible uncertainty realization. This results in a single solution that is heavily biased towards the worst-case scenario and therefore very conservative [151].

The route and the associated loads for each vehicle should be decided upon before the trip, immunizing it against all possible uncertainty realizations. We include the choice for charger visits along the route in this first-stage as well. If not, the problem would turn into a dynamic VRP, where vehicles can be redirected after the uncertainty realization. This variant is known to be even more computationally intractable than a normal VRP [19]. More importantly, optimizing routes solely based on customer sequences overlooks the locations of the chargers, which can result in less efficient trips in practice when chargers far from the anticipated path have to be visited to ensure energy feasibility. Planning the visits to chargers in advance allows for co-optimizing the allocation of charging resources at the depot and paves the way to charger reservations at Shell service stations. Lastly, drivers generally favor the predictability of a fixed route over one that constantly changes. Therefore, we consolidate the routing and load level variables $(x_{ij}^k,\, r_k,\, u_j^k)$ into a first-stage decision vector $\boldsymbol{x}$, which is determined prior to the realization of uncertainty.

In contrast, we do not strictly need a single charging and timing solution that works for every uncertainty realization. It suffices to know that for a chosen load-feasible route, some feasible charging and timing solution exists for any realization of the uncertainty. Once the uncertainty is revealed during the execution of the trip, charging and timing decisions can be made accordingly. This adaptability allows for improved cost, timing and energy efficiency during the execution of the trip. Hence, we consider the timing decision variables $(\tau_i, \tau_k,\, \sigma_i, \sigma_k,\, \delta_k)$ and the charging decision variables $(y_j^k,\, Y_f^k)$ as adaptive. By replicating these decision variables and their corresponding constraints for each scenario $s \in \mathcal{S}(\mathcal{U})$ in Model 11, it is able to account for all possible scenarios during the first decision stage. When represented in vector form, the scenario-specific variables form a set of candidate decision vectors, denoted by $\{\boldsymbol{y}^s\}$. In the second decision stage, the final second-stage decision vector, $\boldsymbol{y}$ can be selected from this set.

Compact Matrix Form

To streamline notation, we introduce a compact matrix representation for Model 11 in Model 12, analogous the representations used in Section 2.5. We refer to Appendix A for the exact definition of the notation.

Model 12 – HGEV-FPP-TECU in Compact Matrix Form

$$c^*_{total} = \min_{\boldsymbol{x},\, \{\boldsymbol{y}^s\}} \quad (\boldsymbol{c}^\top \boldsymbol{x} \; + \; \max_{\boldsymbol{s} \in \mathcal{S}(\mathcal{U})} \boldsymbol{d}^\top \boldsymbol{y}^s) \tag{3.44}$$

s.t.

$$\boldsymbol{A}(\boldsymbol{x}) \leq \boldsymbol{b}, \tag{3.45}$$

$$\boldsymbol{F}(\boldsymbol{z}^s)\boldsymbol{y}^s \leq \boldsymbol{f}(\boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.46}$$

$$\boldsymbol{G}(\boldsymbol{z}^s)\boldsymbol{x} + \boldsymbol{H}(\boldsymbol{z}^s)\boldsymbol{y}^s \leq \boldsymbol{j}(\boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.47}$$

$$\boldsymbol{L}(\boldsymbol{z}^s)\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.48}$$

$\equiv$

$$c^*_{total} = \min_{\boldsymbol{x},\, \eta,\, \{\boldsymbol{y}^s\}} \; \eta \tag{3.49}$$

s.t.

$$\boldsymbol{A}(\boldsymbol{x}) \leq \boldsymbol{b}, \tag{3.50}$$

$$\boldsymbol{L}(\boldsymbol{z}^s)\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\mathcal{U}). \tag{3.51}$$

$$\eta \geq \boldsymbol{c}^\top \boldsymbol{x} \; + \; \boldsymbol{d}^\top \boldsymbol{y}^s \quad \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.52}$$

$$\boldsymbol{y}^s \in \Omega(\boldsymbol{x}, \boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\mathcal{U}), \tag{3.53}$$

$$\eta \in \mathbb{R}_{\geq 0}. \tag{3.54}$$

The second formulation follows from the definition of the second-stage feasible region $\Omega(\boldsymbol{x}, \boldsymbol{z}^s)$ from Definition 6. This region includes all feasible second-stage decisions $\boldsymbol{y}$ for a given combination of first-stage decisions $\boldsymbol{x}$ and scenario $\boldsymbol{z}^s$.

3.3. Exploring Scenarios via Column and Constraint Generation

When using the box uncertainty set $\mathcal{U}_{box}$, a single worst-case scenario can directly be identified and optimized for. However, for more sophisticated uncertainty sets, such as the locally budgeted uncertainty set $\mathcal{U}_{local}$, no single worst-case scenario exists. Instead, multiple scenarios must likely be evaluated to ensure feasibility across the entire set. Since $\mathcal{U}_{local}$ is compact and convex [183], robust feasibility can be achieved by optimizing over all of its extreme points, denoted by $ext(\mathcal{U}_{local})$ [154, 155]. Nevertheless, the number of corresponding scenarios

$$\sum_{(\mathcal{I}_i,\, \Gamma_i) \in \mathcal{P}} \sum_{k=0}^{\lfloor \Gamma_i \rfloor} \binom{|\mathcal{I}_i|}{k} \tag{3.55}$$

can become prohibitively large for high dimensional uncertainty sets, which naturally come with larger HGEV-FPP-TECU instances.

To limit the number of scenarios evaluated during optimization, we utilize the Column and Constraint Generation (CCG) method for ARO, as introduced in Subsection 2.5.2. This outer approximation technique starts with solving the problem under the nominal scenario and incrementally adds new scenarios to the problem through a pessimization process, aiming to ensure robustness across the entire uncertainty set without exhaustively enumerating all (extreme) scenarios.

A key advantage of the CCG approach is that the corresponding Restricted Master Problem (RMP) can be formulated as Model 12, operating on a finite subset of scenarios $\bar{\mathcal{U}} \subseteq \mathcal{U}$, indexed by $\mathcal{S}(\bar{\mathcal{U}})$, over which the original problem constraints and variables are copied (see Model 13). This allows the use of a wide range of existing solution techniques developed for EVRPs, in contrast to the Benders decomposition method, which requires the use of exact methods to generate cutting planes. Following the approach of Jeong et al. [113], we therefore opt to use CCG. However, we employ an alternative and novel pessimization procedure to identify constraint-violating scenarios that are added to the RMP.

Model 13 – Restricted Master Problem (RMP)

$$c_{total} = \min_{\boldsymbol{x},\, \eta,\, \{\boldsymbol{y}^s\}} \eta \tag{3.56}$$

s.t.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \tag{3.57}$$

$$\boldsymbol{L}(\boldsymbol{z}^s)\boldsymbol{x} \leq \boldsymbol{l}(\boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\bar{\mathcal{U}}), \tag{3.58}$$

$$\eta \geq \boldsymbol{c}^\top \boldsymbol{x} \, + \, \boldsymbol{d}^\top \boldsymbol{y}^s \quad \forall s \in \mathcal{S}(\bar{\mathcal{U}}), \tag{3.59}$$

$$\boldsymbol{y}^s \in \Omega(\boldsymbol{x}, \boldsymbol{z}^s) \quad \forall s \in \mathcal{S}(\bar{\mathcal{U}}), \tag{3.60}$$

$$\eta \in \mathbb{R}_{\geq 0}. \tag{3.61}$$

Fig. 3.1 describes our full CCG framework with a flow chart. Algorithm 1 gives the top-level pseudocode. The next subsections detail the pessimization approach. Subsection 3.3.1 outlines two established pessimization approaches and highlights their limitations, which motivate the introduction of the novel OSLP pessimization algorithm, which is discussed in Subsection 3.3.2.

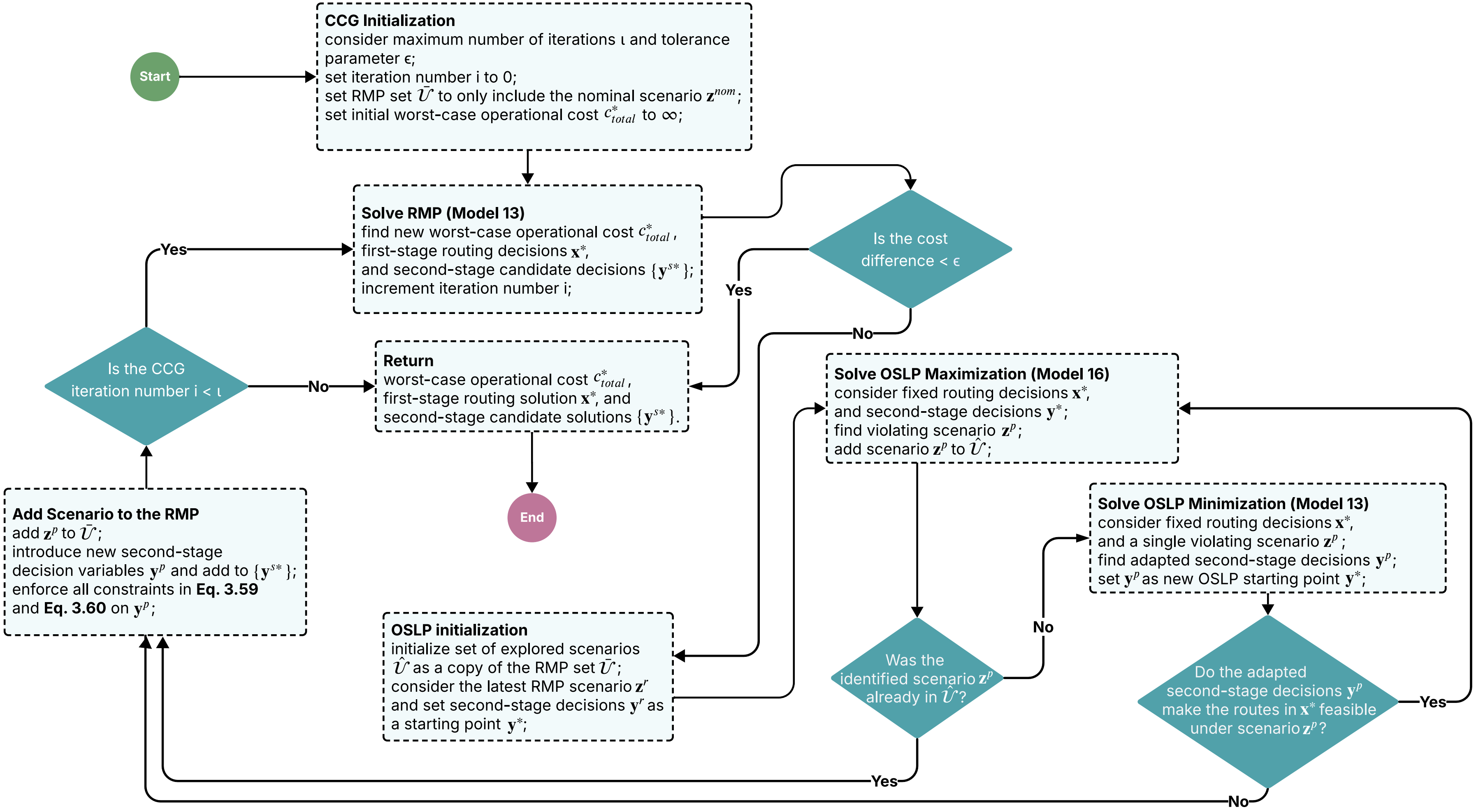

Start
CCG Initialization
consider maximum number of iterations ι and tolerance parameter ϵ;
set iteration number i to 0;
set RMP set $\bar{\mathcal{U}}$ to only include the nominal scenario $\mathbf{z}^{nom}$;
set initial worst-case operational cost c^*_{total} to ∞;
Solve RMP (Model 13)
find new worst-case operational cost c^*_{total},
first-stage routing decisions $\mathbf{x}^*$,
and second-stage candidate decisions $\{\mathbf{y}^{s*}\}$;
increment iteration number i;
Is the cost difference < ϵ
Yes
No
Return
worst-case operational cost c^*_{total},
first-stage routing solution $\mathbf{x}^*$, and
second-stage candidate solutions $\{\mathbf{y}^{s*}\}$.
End
Is the CCG iteration number i < ι
Yes
No
Add Scenario to the RMP
add $\mathbf{z}^p$ to $\bar{\mathcal{U}}$;
introduce new second-stage decision variables $\mathbf{y}^p$ and add to $\{\mathbf{y}^{s*}\}$;
enforce all constraints in Eq. 3.59 and Eq. 3.60 on $\mathbf{y}^p$;
OSLP initialization
initialize set of explored scenarios $\hat{\mathcal{U}}$ as a copy of the RMP set $\bar{\mathcal{U}}$;
consider the latest RMP scenario $\mathbf{z}^r$ and set second-stage decisions $\mathbf{y}^r$ as a starting point $\mathbf{y}^*$;
Solve OSLP Maximization (Model 16)
consider fixed routing decisions $\mathbf{x}^*$,
and second-stage decisions $\mathbf{y}^*$;
find violating scenario $\mathbf{z}^p$;
add scenario $\mathbf{z}^p$ to $\hat{\mathcal{U}}$;
Was the identified scenario $\mathbf{z}^p$ already in $\hat{\mathcal{U}}$?
No
Yes
Solve OSLP Minimization (Model 13)
consider fixed routing decisions $\mathbf{x}^*$,
and a single violating scenario $\mathbf{z}^p$;
find adapted second-stage decisions $\mathbf{y}^p$;
set $\mathbf{y}^p$ as new OSLP starting point $\mathbf{y}^*$;
Do the adapted second-stage decisions $\mathbf{y}^p$ make the routes in $\mathbf{x}^*$ feasible under scenario $\mathbf{z}^p$?
Yes
No

Figure 3.1: Flow chart of the CCG solution approach

Algorithm 1 – Column and Constraint Generation Framework

```
Input: Uncertainty set $\mathcal{U}$, RMP solver RMPSolver, maximum number of iterations $\iota$,
       tolerance parameter $\epsilon$
Output: Anticipated worst-case total operational cost $c^*_{total}$, best first-stage decisions $\boldsymbol{x}^*$,
        second-stage candidate solutions $\{\boldsymbol{y}^{s*}\}$
$\bar{\mathcal{U}} \leftarrow \{\boldsymbol{z}^{nom}\}$;
$\{\boldsymbol{y}^{s*}\} \leftarrow \emptyset$;
$\boldsymbol{x}^* \leftarrow \boldsymbol{0}$;
$c^*_{cost} \leftarrow \infty$;
$i \leftarrow 0$;
while $i < \iota$ do
    $(c_{total}, \boldsymbol{x}, \{\boldsymbol{y}^s\}) \leftarrow$ RMPSolver$(\bar{\mathcal{U}}, \boldsymbol{x}^*, \{\boldsymbol{y}^{s*}\})$;
    if $|c^*_{total} - c_{total}| < \epsilon$ then
        break;
    end
    $c^*_{total} \leftarrow c_{total}$;
    $\boldsymbol{x}^* \leftarrow \boldsymbol{x}$;
    $\{\boldsymbol{y}^{s*}\} \leftarrow \{\boldsymbol{y}^s\}$;
    $\boldsymbol{y}^r \leftarrow \boldsymbol{y}^s$ for most recently added $s \in \mathcal{S}(\bar{\mathcal{U}})$;
    $\boldsymbol{z}^p \leftarrow$ OSLP$(\boldsymbol{x}, \boldsymbol{y}^r, \bar{\mathcal{U}}, \mathcal{U})$;
    if $\boldsymbol{z}^p \in \bar{\mathcal{U}}$ then
        break;
    end
    $\bar{\mathcal{U}} \leftarrow \bar{\mathcal{U}} \cup \{\boldsymbol{z}^p\}$;
    $i \leftarrow i + 1$;
end
return $(c^*_{total}, \boldsymbol{x}^*, \{\boldsymbol{y}^{s*}\})$
```

Here, the RMPSolver is passed as an argument to the algorithm to allow different solution methods to be used. We further detail the employed solvers in Section 3.5. The OSLP procedure corresponds to Algorithm 3. The current best RMP solution $(\boldsymbol{x}^*, \{\boldsymbol{y}^{s*}\})$ is passed to the RMPSolver in order to allow warm restarts where the variables are initialized using the current first- and second-stage decisions. Parameter ι determines the computational budget for the CCG procedure. Setting low values for ι ensures that a solution is found quickly. However, it may not be robust to all scenarios. In contrast, setting ι high ensures that the procedure is only stopped whenever the algorithm is unable to find any violating scenarios. Nevertheless, this does not guarantee robust feasibility, as we will discuss shortly. The value of ϵ decides which improvements to the objective value are deemed significant enough to continue the procedure.

3.3.1. Established Formulations of the Pessimization Problem

To check the robustness of first-stage solution $\boldsymbol{x}$, resulting from solving the RMP from Model 13, against uncertainty realizations in $\mathcal{U} \setminus \bar{\mathcal{U}}$, we solve a pessimization problem that identifies constraint violating scenarios $\boldsymbol{z}^p \in \mathcal{U} \setminus \bar{\mathcal{U}}$. We begin by presenting two established formulations of such a pessimization problem from the literature, which serve as the foundation for our novel OSLP pessimization approach.

Naive Pessimization Approach

In the original CCG approach outlined by Zeng and Zhao [155], the pessimization problem is formulated as the problem of finding a new uncertainty realization $\boldsymbol{z}^p \in \mathcal{U} \setminus \bar{\mathcal{U}}$ such that at least one constraint is violated for every scenario-specific solution $(\boldsymbol{x}, \boldsymbol{y}^s)$ with $s \in \mathcal{S}(\bar{\mathcal{U}})$, and the overall minimum constraint violation κ is maximized. This problem can be formulated as an MILP, as shown in Model 14.

Model 14 – Naive Pessimization Problem

$$\max_{\boldsymbol{z},\{v_i\},\{w_j\},\kappa} \quad \kappa \tag{3.62}$$

s.t.

$$\boldsymbol{G}(\boldsymbol{z})^i\boldsymbol{x} + \boldsymbol{H}(\boldsymbol{z})^i\boldsymbol{y}^s \geq \boldsymbol{j}(\boldsymbol{z})^i + \kappa - M(1-v_i) \quad \forall s \in \mathcal{S}(\bar{\mathcal{U}}),\, \forall i \in [m_3] \tag{3.63}$$

$$\boldsymbol{L}(\boldsymbol{z})^j\boldsymbol{x} \geq \boldsymbol{l}(\boldsymbol{z})^j + \kappa - M(1-w_j) \quad \forall s \in \mathcal{S}(\bar{\mathcal{U}}),\, \forall j \in [m_4] \tag{3.64}$$

$$\sum_{i=1}^{m_3} v_i + \sum_{j=1}^{m_4} w_j \geq 1 \tag{3.65}$$

$$v_i \in \mathbb{B} \quad \forall i \in [m_3] \tag{3.66}$$

$$w_j \in \mathbb{B} \quad \forall j \in [m_4] \tag{3.67}$$

$$\kappa \in \mathbb{R}_{\geq 0} \tag{3.68}$$

$$\boldsymbol{z} \in \mathcal{U} \tag{3.69}$$

Here, we decompose matrices $\boldsymbol{G}(\boldsymbol{z})$ and $\boldsymbol{H}(\boldsymbol{z})$ and vector $\boldsymbol{j}(\boldsymbol{z})$ back into their m_3 rows i that define coupling constraints between the adaptive and non-adaptive variables. We do the same for constraint matrix $\boldsymbol{L}(\boldsymbol{z})$ and vector $\boldsymbol{l}(\boldsymbol{z})$, giving m_4 rows j that define constraints on the non-adaptive variables under uncertainty. This ensures that we can impose restrictions on the violations of individual constraints. The binary variable v_i indicates whether a specific constraint in Eq. (3.63) is violated, whereas w_j does the same for Eq. (3.64). The big-M term is conditionally subtracted from the right-hand side of the equations to ensure that the constraints are violated by at least κ only if the corresponding binary constraint violation variable is set to 1. Finally, Eq. (3.65) ensures that at least one constraint is violated.

Initially, we implemented the naive pessimization approach within our framework and observed that, for many of the identified scenarios $\boldsymbol{z}^p$, the corresponding second-stage feasible region $\Omega(\boldsymbol{x}, \boldsymbol{z}^p)$ was non-empty. This revealed the presence of false positives where the constraints in Model 14 were violated, yet feasible second-stage solutions $\boldsymbol{y}$ did exist, albeit not among the current candidate set $\{\boldsymbol{y}^s\}$. Even minor variations in $\boldsymbol{z}$ required different $\boldsymbol{y}$ values to satisfy all constraints, making it unlikely for any of the existing candidate solutions to remain feasible. This effect was amplified by the continuous nature of the decision variables. As an example, when the traversal duration on a specific edge increased, the current service start time at the next customer would become invalid, as it occurred before the vehicle's revised arrival time, thus requiring the generation of a new candidate decision to preserve feasibility. This issue arose even when the new arrival still fell within the customer's time window. Consequently, many scenarios were added to the RMP not because they led to meaningful violations (e.g., battery depletion or late arrivals), but simply because their feasible second-stage decisions were not already captured in the current candidate set. This significantly reduced the effectiveness of the pessimization method, requiring a large number of CCG iterations to convergence.

Bilinear Pessimization Approach

Jeong et al. [113] propose an alternative pessimization approach that introduces slack variables $\boldsymbol{\kappa} = (\boldsymbol{\kappa_1}, \boldsymbol{\kappa_2})$ and associated penalty terms $\boldsymbol{\pi} = (\boldsymbol{\pi_1}, \boldsymbol{\pi_2})$ to relax both the second-stage and coupling constraints, modelling the objective function as a corridor-based generalized cost function [32, 184]. Unlike the naive pessimization approach, this bilinear method does not restrict the second-stage decision variable $\boldsymbol{y}$ to the current candidate solutions $\boldsymbol{y}^s$ for $s \in \mathcal{S}(\bar{\mathcal{U}})$. Instead, it identifies new scenarios by solving a bilevel optimization problem that maximizes the generalized cost over $\boldsymbol{z}$ while minimizing it over all possible $\boldsymbol{y}$. This results in identifying a scenario $\boldsymbol{z}^p$ for which the best possible second-stage response $\boldsymbol{y}^p$ results in the highest generalized cost.

Such a bilevel optimization problem, as displayed in Model 15, cannot be solved with many off-the-shelf MILP solvers directly as it is non-convex. However, as the inner minimization problem over $\boldsymbol{y}$ is an LP, it can be dualized as a maximization problem over dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$, yielding a bilinear term only in the objective function, such that the feasible region can be separated between primal variables $\boldsymbol{z}$ and dual variables $(\boldsymbol{\lambda}, \boldsymbol{\phi})$. For $\boldsymbol{z}$ the region simply corresponds to $\mathcal{U}$. For $(\boldsymbol{\lambda}, \boldsymbol{\phi})$ the corresponding dual feasible region is described by Definition 16.

Model 15 – Bilinear Pessimization Problem

(Primal)

$$\max_{\boldsymbol{z}\in\mathcal{U}} \min_{\boldsymbol{y}} \quad \boldsymbol{d}^\top \boldsymbol{y} + \boldsymbol{\pi}^\top \boldsymbol{\kappa} \tag{3.70}$$

s.t.

$$\boldsymbol{F}(\boldsymbol{z})\boldsymbol{y} \leq \boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{\kappa}_1, \tag{3.71}$$

$$\boldsymbol{G}(\boldsymbol{z})\boldsymbol{x}^* + \boldsymbol{H}(\boldsymbol{z})\boldsymbol{y} \leq \boldsymbol{j}(\boldsymbol{z}) + \boldsymbol{\kappa}_2, \tag{3.72}$$

(Dual)

$$\max_{\boldsymbol{z},\boldsymbol{\lambda},\boldsymbol{\phi}} \quad \boldsymbol{\phi}^\top (\boldsymbol{G}(\boldsymbol{z})\boldsymbol{x}^* - \boldsymbol{j}(\boldsymbol{z})) - \boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{z}) \tag{3.73}$$

s.t.

$$-\boldsymbol{\lambda}^\top \boldsymbol{F}(\boldsymbol{z}) - \boldsymbol{\phi}^\top \boldsymbol{H}(\boldsymbol{z}) \leq \boldsymbol{d}^\top \tag{3.74}$$

$$0 \leq \boldsymbol{\lambda} \leq \boldsymbol{\pi}_1 \tag{3.75}$$

$$0 \leq \boldsymbol{\phi} \leq \boldsymbol{\pi}_2 \tag{3.76}$$

$$\boldsymbol{z} \in \mathcal{U} \tag{3.77}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^{m_2} \tag{3.78}$$

$$\boldsymbol{\phi} \in \mathbb{R}^{m_3} \tag{3.79}$$

Definition 16 – Dual Feasible Region

$$\Pi := \{(\boldsymbol{\lambda}, \boldsymbol{\phi}) \;:\; \textit{Eq. } (3.74) - \textit{Eq. } (3.79)\} \tag{3.80}$$

The problem's separability enables the use of the Alternating Direction (AD) cutting plane algorithm introduced by Konno [175], which is guaranteed to converge to a Karush-Kuhn-Tucker (KKT) point [185]. Although the optimal solution is always among the KKT points, there is no guarantee that the found scenario is in fact the optimal solution. Algorithm 2 details the procedure, which alternately solves the maximization problem over $(\boldsymbol{\lambda}, \boldsymbol{\phi})$ and the maximization problem over $\boldsymbol{z}$ to obtain a Lower Bound (LB) and Upper Bound (UB) on the dualized cost function, while keeping $\boldsymbol{z}$ and $(\boldsymbol{\lambda}, \boldsymbol{\phi})$ fixed, respectively. The algorithm starts by considering the current first-stage decisions $\boldsymbol{x}$ obtained from solving the RMP under the nominal scenario $\boldsymbol{z}^{nom} \in \mathcal{U}$, and is stopped whenever the difference between upper and the lower bound falls below a predefined tolerance term ϵ. The scenario $\boldsymbol{z}^p$ identified through this process is then added to the RMP. The corresponding upper bound on the generalized cost serves as a benchmark for the optimal robust feasible solution over $\mathcal{U}$ and can be used to define a stopping criterion for the CCG procedure.

Algorithm 2 – Alternating Direction (AD)

```
Input: Uncertainty set $\mathcal{U}$, tolerance parameter $\epsilon$, first-stage decisions $\boldsymbol{x}$
Output: Chosen scenario $\boldsymbol{z}^p \in \mathcal{U}$, upper bound on generalized cost $UB$
$\boldsymbol{z}^p \leftarrow \boldsymbol{z}^{nom}$ ;
$LB \leftarrow 0$, $UB \leftarrow \infty$ ;
while $UB - LB > \epsilon$ do
    Compute: $LB \leftarrow \max_{(\boldsymbol{\lambda},\boldsymbol{\phi})\in\Pi} \boldsymbol{\phi}^\top(\boldsymbol{G}(\boldsymbol{z}^p)\boldsymbol{x} - \boldsymbol{j}(\boldsymbol{z}^p)) - \boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{z}^p)$ and let $(\boldsymbol{\lambda}^*, \boldsymbol{\phi}^*)$ be its
     optimal solution;
    Compute: $UB \leftarrow \max_{\boldsymbol{z}\in\mathcal{U}} \boldsymbol{\phi}^{*\top}(\boldsymbol{G}(\boldsymbol{z})\boldsymbol{x} - \boldsymbol{j}(\boldsymbol{z})) - \boldsymbol{\lambda}^{*\top}\boldsymbol{f}(\boldsymbol{z})$ and let $\boldsymbol{z}^*$ be its optimal solution;
    $\boldsymbol{z}^p \leftarrow \boldsymbol{z}^*$;
end
return $(\boldsymbol{z}^p, UB)$
```

A significant drawback of the bilinear pessimization method is its lack of theoretical guarantees regarding the infeasibility of the scenarios it identifies. Since the objective function is a generalized cost function, combining both actual costs and penalties for constraint violations, it cannot reliably differentiate between scenarios that are truly infeasible and those that simply lead to high costs. Consequently, the method may identify a scenario $\boldsymbol{z}^p$ for which the second-stage feasible region $\Omega(\boldsymbol{x}, \boldsymbol{z}^p)$ is non-empty, thereby limiting the practical relevance of such a scenario in guiding robust decision-making.

3.3.2. Novel One-Step Look-ahead Pessimization Method

Because neither the naive nor the bilinear pessimization approach offers explicit guarantees about the infeasibility of the scenarios they identify, we propose a novel pessimization approach that takes inspiration from both methods but focuses on finding scenarios that lead to proven constraint violations with practical relevance, such as running out of energy, arriving late, or experiencing overlapping charging sessions. We name this the One-Step Look-ahead Pessimization (OSLP) algorithm as it repeatedly performs a One-Step look-ahead for a given RMP solution $(\boldsymbol{x}, \{\boldsymbol{y}^s\})$ to identify uncertainty realizations $\boldsymbol{z}^p \in \mathcal{U} \setminus \bar{\mathcal{U}}$ for which the second-stage feasible region $\Omega(\boldsymbol{x}, \boldsymbol{z}^p)$ is empty. The key innovation of OSLP lies in its ability to sidestep the complexity of solving a non-convex bilevel optimization problem by decomposing it into two alternating procedures, each solvable via linear programming and without the need to relax any of the original problem constraints. By focusing only on practically relevant violations, OSLP avoids adding false positives so that the identified scenarios are guaranteed to be infeasible, thereby accelerating CCG convergence.

Fig. 3.2 illustrates the concept of a one-step look-ahead to detect potential constraint violations. Based on the current routing solution and a chosen second-stage decision candidate, the pessimization procedure seeks values for the uncertain parameters that would cause a violation at the next location in the route, considering the implied new second-stage decisions. In this example, a time window violation occurs at customer C3 due to an increased traversal duration on the edge from C2 to C3.

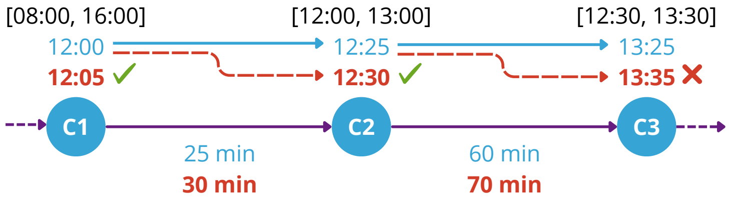


Figure 3.2: Example of a time window violation occurring one step ahead in the current routing solution given the current arrival times and increased traversal durations under pessimization

The OSLP algorithm consists of two sub-procedures that are alternately executed. In the Maximization procedure, potentially infeasible scenarios are uncovered by identifying those that lead to the greatest constraint violations one step ahead in the routing and charging schedule, given the current first- and second-stage decisions. In the MINIMIZATION procedure, we attempt to mitigate these violations by adjusting the second-stage decisions, thereby verifying whether the scenario is truly infeasible. The second-stage candidate decisions corresponding to the most recently added scenario in the RMP are used as the starting point for pessimization. Intuitively, this scenario reflects the most challenging conditions encountered so far. Using its associated decisions as a starting point is beneficial because if they fail under a new scenario, this new scenario must be even harder, such that none of the candidate solutions currently in the RMP will work. This approach helps guide the search towards progressively harder scenarios.

Maximization Procedure

A scenario is marked potentially infeasible by the maximization procedure if, for any of the vehicles, it leads to at least one of the following violations:

1. A vehicle depletes its battery before reaching the next customer due to increased energy consumption, given its current SoC upon departure.
2. A vehicle arrives at the next customer after the latest possible service start time due to increased travel, service, and/or waiting durations, given it's current departure time (such as in Fig. 3.2).
3. A vehicle's charging session overlaps with a subsequent session at the same charger due to reduced charging rates, given the current start times of both charging sessions.

To identify such scenarios, we take the second-stage decision vector $\boldsymbol{y}^r$ corresponding to scenario r that was last added to the RMP as a starting point and solve the LP from Model 16, maximizing the total potential constraint violation, and returning the corresponding scenario $\boldsymbol{z}^p$.

Model 16 – Maximization LP within OSLP

$$\max_{\boldsymbol{z}} \sum_{j \in \mathcal{V}'_{N+1}} (\kappa^j_{time} + \kappa^j_{energy}) + \sum_{f,o \in \mathcal{F}'\; o>f,\; Parent(o)=Parent(f)} \kappa^{fo}_{overlap} + c_{small} \sum_{i \in [l],\; z^{dev}_i > 0} \frac{z_i - z^{nom}_i}{z^{dev}_i} \tag{3.81}$$

s.t.

One-Step-Ahead Violation Constraints:

Potential energy violations after a charger visit are tracked:

$$Y^k_f - h(d_{fj}, t_{fj}, m(u^k_j)) \le 0 - \hat{\kappa}^j_{energy} \quad \forall k \in \mathcal{K},\ \forall f \in \mathcal{F}',\ \forall j \in \mathcal{V}'_{N+1},\ x^k_{fj} = 1. \tag{3.82}$$

Potential energy violations after a customer visit are tracked:

$$y^k_i - h(d_{ij}, t_{ij}, m(u^k_j)) \le 0 - \hat{\kappa}^j_{energy} \quad \forall k \in \mathcal{K},\ \forall i \in \mathcal{V}'_0,\ \forall j \in \mathcal{V}'_{N+1},\ x^k_{ij} = 1. \tag{3.83}$$

Potential time violations after a charger visit are tracked:

$$\tau_f + g^k_f(Y^k_f - y^k_f) + t_{fj} + w_j \ge l_j + \hat{\kappa}^j_{time} \quad \forall k \in \mathcal{K},\ \forall f \in \mathcal{F}',\ \forall k \in \mathcal{K},\ \forall j \in \mathcal{V}'_{N+1},\ x^k_{fj} = 1. \tag{3.84}$$

Potential time violations after a customer visit are tracked:

$$\tau_i + \nu_i + t_{ij} + w_j \ge l_j + \hat{\kappa}^j_{time} \quad \forall i \in \mathcal{V}'_0,\ \forall j \in \mathcal{V}'_{N+1},\ \sum_{k \in \mathcal{K}} x^k_{ij} = 1. \tag{3.85}$$

Potential time violations after departure from the depot are tracked:

$$\tau_k + \nu_0 + t_{0j} + w_j \ge l_j + \hat{\kappa}^j_{time} \quad \forall k \in \mathcal{K},\ \forall j \in \mathcal{V}'_{N+1},\ x^k_{0j} = 1. \tag{3.86}$$

Potential charging overlap violations are tracked:

$$\tau_f + g^k_f(Y^k_f - y^k_f) \ge \tau_o + \hat{\kappa}^{fo}_{overlap} \quad \forall k \in \mathcal{K},\ \forall f, o \in \mathcal{F}',\ o > f,\ Parent(o) = Parent(f). \tag{3.87}$$

Route Membership Constraints:
Only positive violations occurring within the current routing solution are counted towards the optimization objective:

$$\kappa^{j}_{time} \geq \hat{\kappa}^{j}_{time} \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.88}$$

$$\kappa^{j}_{time} \leq \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}'_0} x^{k}_{ij} M \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.89}$$

$$\kappa^{j}_{energy} \geq \hat{\kappa}^{j}_{energy} \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.90}$$

$$\kappa^{j}_{energy} \leq \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}'_0} x^{k}_{ij} M \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.91}$$

$$\kappa^{fo}_{overlap} \geq \hat{\kappa}^{fo}_{overlap} \quad \forall f,\, o \in \mathcal{F}',\, o > f,\, Parent(o) = Parent(f), \tag{3.92}$$

$$\kappa^{fo}_{overlap} \leq \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}'_0} x^{k}_{if} \quad \forall f,\, o \in \mathcal{F}',\, o > f,\, Parent(o) = Parent(f), \tag{3.93}$$

$$\kappa^{fo}_{overlap} \leq \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}'_0} x^{k}_{io} \quad \forall f,\, o \in \mathcal{F}',\, o > f,\, Parent(o) = Parent(f). \tag{3.94}$$

Domain Constraints for the Decision Variables:

$$\kappa^{j}_{time},\, \kappa^{j}_{energy} \in \mathbb{R}_{\geq 0} \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.95}$$

$$\hat{\kappa}^{j}_{time},\, \hat{\kappa}^{j}_{energy} \in \mathbb{R} \quad \forall j \in \mathcal{V}'_{N+1}, \tag{3.96}$$

$$\kappa^{fo}_{overlap} \in \mathbb{R}_{\geq 0} \quad \forall f,\, o \in \mathcal{F}',\, o > f,\, Parent(o) = Parent(f), \tag{3.97}$$

$$\hat{\kappa}^{fo}_{overlap} \in \mathbb{R} \quad \forall f,\, o \in \mathcal{F}',\, o > f,\, Parent(o) = Parent(f), \tag{3.98}$$

$$\boldsymbol{z} \in \mathcal{U}. \tag{3.99}$$

Here, the uncertain parameters under optimization are indicated in red. The current second-stage decisions only serve as a starting point for the one-step look-ahead and are not adapted during the maximization procedure to maintain the convexity of the problem. The $\hat{\kappa}_{\bullet}$ variables are auxiliary variables to track the slack, which only take positive values in case a violation occurs. By restricting the values $\kappa_{\bullet}$ to be non-negative and 0 for locations not included in the route, only actual violations are counted towards the objective value under maximization.

The final term in the objective function from Eq. (3.81) rewards higher relative deviations from the nominal value for each uncertain parameter, and is multiplied by a very small constant $c_{small} = 10^{-5}$, so that it does not influence the total objective too much. This encourages full utilization of the available uncertainty budget, helping to prevent premature convergence (as discussed later). The uncertainty set membership constraint in Eq. (3.99) can be enforced linearly for any convex compact set. For brevity, the corresponding linear constraints are omitted.

Minimization Procedure

To confirm that a potentially infeasible scenario $\boldsymbol{z}^p$ is truly infeasible, meaning $\Omega(\boldsymbol{x}, \boldsymbol{y}^s) = \emptyset$, the minimization procedure subsequently solves the RMP from Model 13 with $\boldsymbol{x}$ fixed and the uncertainty set restricted to $\bar{\mathcal{U}} = \{\boldsymbol{z}^p\}$, so that it becomes an LP. If the resulting total cost c_{total} is finite, then this indicates that a feasible second-stage solution exists, so that $\boldsymbol{z}^p$ is not truly infeasible. This could happen for example when the maximization step suggests that a vehicle arrives late due to increased travel time, after which the minimization step adjusts the timing or charging schedule, so that timely arrival is still possible under adjusted decisions $\boldsymbol{y}^p$. In such a case, we set $\boldsymbol{y}^p$ as the new starting point for the maximization procedure and repeat the process. The OSLP algorithm terminates either when the maximization step fails to uncover a new violating scenario or when the minimization step confirms that $\Omega(\boldsymbol{x}, \boldsymbol{z}^p) = \emptyset$. In both cases, the latest identified scenario $\boldsymbol{z}^p$ is returned to be included in the RMP. In the first case, this serves the purpose of checking if there exists an alternative routing solution $\boldsymbol{x}'$ for $\bar{\mathcal{U}} \cup \{\boldsymbol{z}^p\}$ that has a lower worst-case cost than the current feasible solution $\boldsymbol{x}$. In the latter case, a feasible solution $\boldsymbol{x}'$ for $\bar{\mathcal{U}} \cup \{\boldsymbol{z}^p\}$ is yet to be found. The full procedure is detailed in Algorithm 3.

Algorithm 3 – One-Step Look-ahead Pessimization (OSLP)

Input: First-stage decisions $\boldsymbol{x}$, latest second-stage decision candidate $\boldsymbol{y}^r$, restricted uncertainty set $\bar{\mathcal{U}}$, full uncertainty set $\mathcal{U}$

Output: Identified scenario $\boldsymbol{z}^p$

```
$\boldsymbol{y}^* \leftarrow \boldsymbol{y}^r$;
$\hat{\mathcal{U}} \leftarrow \bar{\mathcal{U}}$;
$\boldsymbol{z}^p \leftarrow \boldsymbol{0}$;
while True do
    $\boldsymbol{z}^* \leftarrow$ MAXIMIZATION(Model 16, $\boldsymbol{x}, \boldsymbol{y}^*$);
    if $\boldsymbol{z}^* \in \hat{\mathcal{U}}$ then
        break;
    end
    $\boldsymbol{z}^p \leftarrow \boldsymbol{z}^*$;
    $(c_{total}, \boldsymbol{y}^p) \leftarrow$ MINIMIZATION(Model 13, $\boldsymbol{x}, \{\boldsymbol{z}^p\}$);
    if $c_{total} = \infty$ then
        break;
    end
    $\boldsymbol{y}^* \leftarrow \boldsymbol{y}^p$;
    $\hat{\mathcal{U}} \leftarrow \hat{\mathcal{U}} \cup \{\boldsymbol{z}^p\}$;
end
return $\boldsymbol{z}^p$
```

Discussion

While the proposed algorithm effectively avoids false positives, it remains vulnerable to premature convergence. This issue arises when no scenario within the uncertainty set triggers a one-step-ahead violation, and the maximization step fails to uncover scenarios that lead to violations occurring further in the future. We refer to such undetected scenarios as false negatives. As a result, the CCG procedure may terminate early, yielding a solution that is not robustly feasible.

To remedy this, the additional term in the objective function incentivizes the solver to fully utilize the uncertainty budget, while distributing it in a way that most adversely affects one-step-ahead violations along the current route. When residual budget remains, uncertain parameters outside the route also attain higher values, ensuring that the resulting scenarios always lie at the extreme points of the uncertainty set.

As all constraints are linear in the uncertain parameters, the scenario that reveals multi-step-ahead violations must also be among the extreme points of the uncertainty set. Hence, this approach is effective for directly identifying the single extreme point $\boldsymbol{z}^{worst} \in \mathcal{U}_{box}$. However, it does not eliminate the issue of premature convergence under a budgeted uncertainty set such as $\mathcal{U}_{budget}$ and $\mathcal{U}_{local}$, where the budget can be distributed in many different ways, giving rise to many extreme points. The likelihood of blindly selecting the scenario that reveals multi-step-ahead violations among all of them remains low. For a more detailed discussion, we refer to the formal argument in Appendix B.

Note that the additional term in the maximization objective also ensures that the OSLP algorithm always terminates under convex and compact uncertainty sets. In the most computationally intensive case, the MAXIMIZATION procedure systematically enumerates all extreme points $\boldsymbol{z}^{ext} \in ext(\mathcal{U})$, incrementally extending the set $\hat{\mathcal{U}}$. For every identified scenario, the MINIMIZATION step identifies that $\Omega(\boldsymbol{x}, \boldsymbol{z}^{ext}) \neq \emptyset$, so that the process continues until $\hat{\mathcal{U}} = ext(\mathcal{U})$, at which point the MAXIMIZATION procedure must select a point already in $\hat{\mathcal{U}}$, thereby terminating the algorithm.

3.4. Scalability through Adaptive Large Neighbourhood Search

As the problem size grows, solving the HGEV-FPP-TECU exactly becomes computationally infeasible due to the exponential growth in the number of decision variables needed to account for all possible routes. This is further amplified by the replication of adaptive decision variables across all scenarios included in the RMP as a result of making the model adaptive. To address scalability to larger instances, we adopt a metaheuristic solution method to solve the RMP, in line with the work by Jeong et al. [113].

Their approach is centered around a complex corridor-based generalized cost function, as proposed by Schiffer and Walther [172]. This method maps energy consumption to the time domain by considering the time required to recharge the consumed energy. As a result, corridors can be constructed to represent all possible arrival times at a location, depending on the energy that can or needs to be replenished at preceding chargers along the route. By comparing these corridors with the service time windows at each location, violations related to time and energy constraints can be detected and penalized. This cost function only implicitly incorporates second-stage decisions, enabling the use of a relatively simple Variable Neighbourhood Search (VNS) that focuses solely on modifying route structures.

We take a fundamentally different approach, in which we employ Adaptive Large Neighbourhood Search (ALNS), explicitly model second-stage decision candidates and utilize more sophisticated operators that act on both adaptive and non-adaptive variables, while relying on a much simpler generalized cost function. In essence, the intelligence is shifted from the objective function to the operators. By leveraging insights from second-stage constraints, the operators construct neighbourhoods that inherently exclude routing choices likely to be dismissed due to their high generalized cost. This enables a more informed exploration of the solution space. Moreover, the explicit representation of second-stage decisions is essential for our OSLP algorithm, which relies on the most recent second-stage decision candidate to identify one-step-ahead violations. The selection of ALNS as a metaheuristic is primarily motivated by its extensive and successful application in previous works addressing various forms of the EVRP [31, 44, 48, 110], including those involving uncertainty [4, 111, 112].

In Subsection 3.4.1, we slightly revise the definition of HGEV-FPP-TECU to better align with the ALNS framework. Following this, we introduce a customized generalized cost function to effectively guide the search process (Subsection 3.4.2), along with a modified nearest neighbour heuristic to construct an initial solution (Subsection 3.4.3). We adopt commonly used destroy operators from existing literature (Subsection 3.4.4) and design a custom repair operator that guarantees the energy feasibility of the resulting routes (Subsection 3.4.5). We make sure that new scenarios can easily be added to the problem, as described in Subsection 3.4.6. The employed adaptive selection mechanism and acceptance criterion are described in Subsection 3.4.7 and Subsection 3.4.8, respectively. Subsection 3.4.9 outlines the resulting ALNS procedure for solving the RMP, integrating all components.

3.4.1. Problem Redefinition

We adopt a tuple-based representation of routes to bypass the use of binary decision variables x_{ij}^k required by the MILP formulation. To reduce the size of the neighbourhoods, we do not consider dummy vertices and include only the parent chargers $f \in \mathcal{F}$, recording the values of the second-stage decision variables over the entries within the routes. Formally, this leads to the following redefinition of the HGEV-FPP-TECU:

Definition 17 – Redefinition of HGEV-FPP-TECU for ALNS

Consider the HGEV-TECU from Definition 12 with the following modifications:

Let $\mathcal{F}' := \mathcal{F}$, omitting dummy notes to decrease the size of the neighbourhoods. The original task of the dummy nodes, enabling each vehicle to visit a charger up to n_{visits} times, is now fulfilled by tracking all charging sessions per charger and ensuring that no vehicle exceeds the allowed n_{visits}.

Let the route set be redefined as

$$\mathcal{R} := \{R_k : k \in \mathcal{K}\},$$

with redefined vehicle routes

$$R_k := (v_1^k, v_2^k, \dots v_{n_k}^k),$$

where $n_k \in \mathbb{N}$ denotes the variable route length for vehicle $k \in \mathcal{K}$. The route starts at $v_1^k = 0$ and ends at $v_{n_k}^k = N + 1$. The initial depot charger is represented by the decision variable $v_2^k \in \mathcal{F}''$. For $j = 3, \dots, n_k - 1$, the decision variables v_j^k define the remainder of the route by taking values in $\mathcal{V}'$.

Let the first stage-decisions $\boldsymbol{x}$ solely include $\mathcal{R}$.

Let the second-stage decisions $\boldsymbol{y}^s$ for each scenario $s \in \mathcal{S}(\bar{\mathcal{U}})$ include decision variables

- τ_j^{ks} tracking the service start time,
- σ_j^{ks} tracking the service end time,
- y_j^{ks} tracking the SoC on arrival, and
- Y_j^{ks} tracking the SoC on departure

for every vehicle $k \in \mathcal{K}$ and location $v_j^k \in R_k$.

That is, τ_k^s from Definition 12 corresponds to τ_1^{ks}, δ_k^s to σ_2^{ks} and σ_k^s to $\sigma_{n_k}^{ks}$ for all $k \in \mathcal{K}$ and $s \in \mathcal{S}(\bar{\mathcal{U}})$. Furthermore, r_k now corresponds to $n_k > 0$ and u_j^k to $\sum_{v_j^k \in R_k} q_{v_j^k}$, both of which are implied by $\mathcal{R}$.
This results in a problem redefinition suitable for the application of ALNS.

3.4.2. Generalized Cost Function

Other than in the exact approach, where feasibility of the solutions is always enforced, we allow infeasible solutions during the execution of the ALNS, to promote exploration. In order to assess the quality of a solution and guide the search towards feasibility, we employ a Generalized Cost (GC) function over the RMP solution $(\boldsymbol{x}, \{\boldsymbol{y}^s\})$. This function is a special variant of the objective function from Eq. (3.2) that also penalizes constraint violations, much like in Lagrangian relaxation [29].

Definition 18 – Generalized Cost Function

$$GC(\boldsymbol{x}, \{\boldsymbol{y}^s\}) := DC(\boldsymbol{x}) + \pi_{cv} CV(\boldsymbol{x}) + \max_{s \in \mathcal{S}(\bar{\mathcal{U}})} \{SC(\boldsymbol{x}, \boldsymbol{y}^s)\}, \tag{3.100}$$

where

$$DC(\boldsymbol{x}) := \sum_{k \in \mathcal{K}} \sum_{i=1}^{n_k - 1} \sum_{j=i+1}^{n_k} d_{v_i^k v_j^k} c_{km}^k \tag{3.101}$$

accounts for the distance costs,

$$CV(\boldsymbol{x}) := \sum_{k \in \mathcal{K}} max\Big(\sum_{v_i^k \in R_k} q_{v_i^k} - C_k, 0\Big) \tag{3.102}$$

accounts for load-capacity violations penalized by $\pi_{cv} \in \mathbb{R}_{\geq 0}$,

$$SC(\boldsymbol{x}, \boldsymbol{y}^s) := TC(\boldsymbol{x}, \boldsymbol{y}^s) + EC(\boldsymbol{x}, \boldsymbol{y}^s) + \pi_{tv} TV(\boldsymbol{x}, \boldsymbol{y}^s) + \pi_{ev} EV(\boldsymbol{x}, \boldsymbol{y}^s) + \pi_{ov} OV^s(\boldsymbol{x}, \boldsymbol{y}^s), \tag{3.103}$$

accounts for the scenario-dependent cost under RMP scenario $s \in \mathcal{S}(\bar{\mathcal{U}})$, with

$$TC(\boldsymbol{x}, \boldsymbol{y}^s) := \sum_{k \in \mathcal{K}} (\sigma_{n_k}^{ks} - \sigma_2^{ks}) c_{min}^k \tag{3.104}$$

accounting for the time costs,

$$EC(\boldsymbol{x}, \boldsymbol{y}^s) := \sum_{k \in \mathcal{K}} \sum_{v_j^k \in R_k \cap \mathcal{F}'} (Y_j^{ks} - y_j^{ks}) \tag{3.105}$$

accounting for the energy costs,

$$TV(\boldsymbol{x}, \boldsymbol{y}^s) := \sum_{k \in \mathcal{K}} \sum_{v_j^k \in R_k} max(\tau_j^{ks} - l_j, 0) \tag{3.106}$$

accounting for time violations penalized by $\pi_{tv} \in \mathbb{R}_{\geq 0}$,

$$EV(\boldsymbol{x}, \boldsymbol{y}^s) \coloneqq \sum_{k \in \mathcal{K}} \sum_{v_j^k \in R_k} max(-y_j^{ks}, 0) \tag{3.107}$$

accounting for energy violations penalized by $\pi_{ev} \in \mathbb{R}_{\geq 0}$, and

$$OV(\boldsymbol{x}, \boldsymbol{y}^s) \coloneqq \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{K},\, k \neq l} \sum_{v_i^k \in R_k \cap \mathcal{F}'} \sum_{v_j^l \in R_l,\, v_i^k = v_j^l} max(min(\sigma_i^{ks}, \sigma_j^{ls}) - max(\tau_i^{ks}, \tau_j^{ls}), 0) \tag{3.108}$$

accounting for charging overlap violations penalized by $\pi_{ov} \in \mathbb{R}_{\geq 0}$.

This generalized cost function builds upon the one employed by Wang, Zheng, and Zhou [112] by adding the notion of multiple scenarios and extending it with the the charging overlap term from Eq. (3.108).

We also define the insertion cost $IC(i, u, j)$ for inserting a location $u \in \mathcal{V}'$ between i and j in a given route R_k as the added traversal distance resulting from that insertion:

Definition 19 – Insertion Cost

$$IC(i, u, j) \coloneqq d_{iu} + d_{uj} - d_{ij}. \tag{3.109}$$

3.4.3. Construction Heuristic

To construct an initial solution that serves as a starting point for the ALNS procedure, we follow the approach proposed by Keskin and Çatay [31]. Their nearest neighbour method incrementally constructs routes by adding the closest unvisited customer until the vehicle's capacity is reached or further additions would result in time window violations. In contrast to Wang, Zheng, and Zhou [112], who implement a similar method considering the worst-case realizations of the uncertain parameters, we use the nominal scenario $\boldsymbol{z}^{nom}$ as the baseline for route construction, as this aligns with the first scenario considered by the RMP during CCG.

Concretely, every route k is started by selecting an initial depot charger $f \in \mathcal{F}''$ that, considering the inverse charging rate $g_f^{k(nom)}$, enables the vehicle to reach its maximum state of charge at the earliest possible time. We then apply the nearest neighbour insertion procedure, continuing to append the nearest customer $c \in \mathcal{C}$, in terms of distance, to the end of the route until doing so would result in a capacity violation or time window violation directly at the customer or upon a subsequent arrival at the depot. At that point, the vehicle is routed back to the depot.

If the vehicle risks depleting its battery before reaching the next customer, a greedy charger insertion routine is triggered. This routine considers potential chargers $f \in \mathcal{F}$ in order of increasing insertion cost, checking whether each charger is reachable with the current SoC and whether it can recharge the vehicle to full capacity within a time frame that allows timely arrival at the next customer, following the approach proposed by Keskin and Çatay [31]. A novelty introduced by our heuristic is that charger availability is coordinated across the fleet, respecting non-overlapping constraints for charging sessions and seeking gaps between scheduled charging sessions to recharge vehicles as timely as possible.

If no charger allows the timely arrival at the subsequent customer, the vehicle is redirected to the depot, resolving any remaining energy deficits by inserting intermediate chargers as needed. Time window and charging overlap constraints are relaxed during this process, so that a feasible charger can always be found. After all vehicles are assigned routes through this process, any unassigned customers are allocated to vehicles with spare load capacity, even if doing so violates time window constraints. If no such vehicle is available, the customer is assigned to the vehicle with the greatest remaining load capacity, ensuring that all customers are eventually assigned a vehicle.

The worst-case time complexity of the construction heuristic is $\mathcal{O}(N \cdot (N + F))$, as each of the N customer insertions requires checking up to N neighbours and evaluating up to $F = |\mathcal{F}|$ candidate chargers.

3.4.4. Destroy Heuristics

As discussed in Subsection 2.3.2, ALNS works with implicitly defined neighbourhoods, constructed through a two-phase process. In the first phase, destroy heuristics are dynamically chosen to partially destroy existing routes. In the second phase a repair heuristic reconstructs them. Among our destroy heuristics, we differentiate between three types of operators: customer removal, route removal, and charger removal.

Customer Removal

Customer removal operators destroy the route set $\mathcal{R}$ by removing $n = \rho N$ customers, where ρ denotes the customer removal rate that can be tuned based on the instance size. Removed customers are inserted into a removal set $\mathcal{L}$ that can be passed to the repair operator. All routes R_k are considered at once. To simplify the notation used in the time complexity expressions, we use $V = |\mathcal{V}|$ and $S = |\mathcal{S}(\bar{\mathcal{U}})|$.

1. **Random Removal**: Removes n randomly selected customers from across all routes to introduce diversity. The worst-case time complexity of this operator is $\mathcal{O}(N)$.
2. **Worst Removal** [31, 112]: This group of operators aims to remove the worst customers based on a given criterion. n customers are removed by repeatedly drawing a number $\lambda \in [0, 1]$ uniformly at random and selecting the customer at position $\lfloor n\lambda^{\chi_{\text{worst}}} \rfloor$ in the sorted cost list for removal. This mechanism ensures that customers with higher cost values are more likely to be removed, while the randomness introduced by λ and determinism factor χ_{worst} prevents overly deterministic behavior. Specifically, we adopt the following operators using this scheme:
   (a) **Worst Objective Removal**: Removes customers contributing most to the generalized cost. For each customer, we calculate the GC of the solution $(\boldsymbol{x}', \{\boldsymbol{y}'^s\})$ obtained by removing the customer from its route and subtract it from $GC(\boldsymbol{x}, \{\boldsymbol{y}^s\})$. The customers are sorted by descending difference value $GC(\boldsymbol{x}, \{\boldsymbol{y}^s\}) - GC(\boldsymbol{x}', \{\boldsymbol{y}'^s\})$. The worst-case time complexity for this operator is $\mathcal{O}(N \cdot V \cdot S)$ as the computation of $GC(\boldsymbol{x}', \boldsymbol{y}'^s)$ has complexity $\mathcal{O}(V \cdot S)$.
   (b) **Worst Time Violation Removal**: Removes customers i causing the largest time window violations over all RMP scenarios by sorting them by descending value $\max_{s\in\bar{\mathcal{U}}}(\tau_j^{ks} - l_j)$. The worst-case time complexity of this operator is $\mathcal{O}(N \cdot S)$.
   (c) **Worst Energy Violation Removal**: Removes customers i causing the largest energy violations over all RMP scenarios by sorting them by ascending value $\max_{s\in\bar{\mathcal{U}}}(y_j^{ks})$. The worst-case time complexity of this operator is $\mathcal{O}(N \cdot S)$.
   (d) **Worst Distance Removal**: Removes customers u between i and j causing the greatest detours, sorting them by descending insertion cost $IC(i, u, j)$. The worst-case time complexity of this operator is $\mathcal{O}(N)$.
3. **Shaw Removal** [186, 110]: This group of operators aims to remove customers that are similar to each other with respect to spatial, temporal, and/or demand characteristics. First, a customer i is selected randomly. Then the remaining customers are sorted by ascending relatedness measure $r_{ij} = \phi_1 d_{ij} + \phi_2|e_i - e_j| + \phi_3|q_i - q_j| + \phi_4 k_{ij}$, where a smaller r_{ij} corresponds to high similarity and $k_{ij} = -1$ if i and j are in the same route and 1 otherwise. Then the remaining customers are selected as in worst removal, but using a separate determinism factor χ_{shaw}. We adopt the following variants, each with a worst-case time complexity of $\mathcal{O}(N^2)$:
   (a) **Original Shaw Removal**: Removes customers based on ascending relatedness measure r_{ij} for given Shaw parameters ϕ_1, ϕ_2, ϕ_3, ϕ_4.
   (b) **Proximity-Based Removal**: Removes customers that are geographically close to each other, exploiting spatial correlation, setting $\phi_1 = 1$ and all others to 0.
   (c) **Time-Based Removal**: Removes customers with similar service time windows, setting $\phi_2 = 1$ and all others to 0.
   (d) **Demand-Based Removal**: Removes customers with similar demand levels to balance the load in the remaining solution, setting $\phi_3 = 1$ and all others to 0.
4. **Zone Removal based on K-Means Clustering** [110]: Spatially clusters customers into k zones using K-Means [123]. A single zone is randomly selected for removal to disrupt a local area. The worst-case time complexity of this operator is $O(N^2)$.

Route Removal

Route removal operators remove entire routes at once, trying to make significant improvements by redistributing their customers over other vehicles. The worst-case time complexity of these operators is $\mathcal{O}(V)$.

1. **Random Route Removal**: Randomly selects a single route R_k and removes it from $\mathcal{R}$.
2. **Greedy Route Removal** [187]: Randomly selects a percentage $\rho_r \in [\rho_r^{min}, \rho_r^{max}]$ of the routes to remove and selects the corresponding number of routes greedily according to a given criterion:
   (a) **Length-Based Removal**: Removes routes with the fewest customers, thus smallest $|R_k \cap \mathcal{C}|$, aiming for solutions with fewer deployed vehicles.
   (b) **Worst Capacity Violation Removal**: Removes routes with the largest capacity violation $\sum_{v_j \in R_k \cap \mathcal{C}} q_{v_j} - C_k$ to improve feasibility.

Charger Removal

In addition to customer and route removal, we adopt a random charger removal operator, which removes a percentage ρ_f of the charging stations in the route set $\mathcal{R}$. This encourages the algorithm to reconsider second-stage charging decisions by either reinserting the same charger with a different charging amount during the repair phase or a different charger altogether. The worst-case time complexity of this operator is $\mathcal{O}(F)$. While more advanced charger removal strategies, such as those proposed by Keskin and Çatay [31], have not yet been incorporated, we follow their approach by introducing two additional variants for each customer removal operator, complementing the previously described Removing Customer Only (RCO) variant:

- **Remove Customer with Preceding Station (RCwPS)**: This variant removes each customer in the removal set $\mathcal{L}$ along with the immediately preceding charger, if one exists. The rationale is that the preceding charger visit may no longer be necessary when the customer is removed from the route.
- **Remove Customer with Succeeding Station (RCwSS)**: This variant removes the customer and the immediately succeeding charger together, following the same rationale.

The time complexities for these variants are the same as for the RCO variant.

Propagation of Timing and Charging Decisions

Since second-stage decisions $\{\boldsymbol{y}^s\}$ are integrated into the ALNS procedure, they must be updated for each scenario s in the RMP set whenever a customer or charger is removed from the route set. In case of a customer removal, we execute a propagation subroutine that begins at the start of the route, since the removal of the corresponding customer demand changes the load level and therefore the energy consumption over the entire route.

This potentially enables more cost-efficient timing and charging decisions. As we traverse the route, we update the SoC levels according to the newly calculated energy consumption using the energy consumption function from Definition 15. When a charging station is encountered, we ensure that the SoC upon departure matches the maximum of the departure SoC pre-removal and the arrival SoC post-removal. This allows the charging session to be shortened whenever the SoC upon arrival is higher than before, or even to be skipped entirely in case the new arrival SoC exceeds the previous departure SoC.

However, chargers are not removed from the route in the latter case, as their inclusion in the route may prove beneficial when customers are reinserted during the repair phase. Arrival times are recalculated as the maximum of the earliest service start time e_i and the earliest possible arrival time, allowing for earlier arrivals at customers succeeding the removed location when permitted by their time windows.

If a charger is removed from the route, the load levels remain unchanged. As a result, propagation begins only from the preceding location, with both timing and charging decisions updated accordingly for the rest of the route. For the RCwPS and RCwSS operators, two propagation passes are made. One for the customer removal, followed by one for the charger removal. The worst-case time complexity of the propagation procedure is $\mathcal{O}(V \cdot S)$, which leads to an overall complexity of $\mathcal{O}(V^2 \cdot S)$ for the destroy phase, when paired with one of the removal operators.

3.4.5. Repair Heuristics

During the repair phase, we apply a single, customized repair heuristic to reinsert customers from the removal set $\mathcal{L}$ back into the route set $\mathcal{R}$. This heuristic dynamically adjusts the second-stage decisions $\{\boldsymbol{y}^s\}$ for each scenario s in the RMP set, while proactively addressing potential battery, time window, and charging overlap constraint violations on a best-effort basis. The repair process consists of two main components: a customer insertion heuristic and a charger insertion heuristic.

Customer Insertion

We employ a greedy customer insertion heuristic, where for every customer in $\mathcal{L}$, we evaluate the insertion cost from Definition 19 for all possible insertion positions within $\mathcal{R}$. Insertions are rejected whenever a load or time window violation is introduced in any RMP scenario. We track the insertion positions corresponding to the lowest insertion cost. If no feasible position was found among existing routes, we consider starting a new route with the undeployed vehicle corresponding to the highest load capacity. If all vehicles are deployed, we consider inserting the customer at the end of the route corresponding to the vehicle with the largest remaining load capacity and record the insertion cost for that. Then we insert the removed customer with the overall lowest insertion cost into the corresponding position, repeating the process until all removed customers are reinserted, so that $\mathcal{L} = \emptyset$. This procedure has a worst-case time complexity of $\mathcal{O}(N^2 \cdot V)$, as each insertion involves evaluating all customers in $\mathcal{L}$.

Alternative Heuristics

We also tested a sequential greedy insertion variant in which $\mathcal{L}$ was randomly shuffled and customers were inserted in order, reducing the complexity to $\mathcal{O}(N \cdot V)$. However, despite being faster, this approach led to significantly poorer performance and was therefore ultimately discarded.

A key limitation of the greedy insertion heuristic is its myopic nature. While it selects the customer with the lowest immediate insertion cost, this short-term focus can lead to scenarios where customers with limited good insertion options are later forced into poor positions due to earlier greedy choices. To address this, we also explored the Regret-k insertion heuristic proposed by Potvin and Rousseau [188]. This method evaluates, for each customer in $\mathcal{L}$, the difference in insertion cost between its best and k-th best feasible insertion position (regret), and prioritizes customers with the highest regret for insertion. In our preliminary experiments, Regret-2 and Regret-3 insertion proved significantly slower than the greedy heuristic, despite the same time complexity of $\mathcal{O}(N^2 \cdot V)$, likely due to a memory-inefficient implementation. Consequently, we opted to continue using only the greedy insertion heuristic.

Charger Insertion

Note that our customer insertion heuristic only considers time and load violations, basing insertion cost on a distance metric. Implications in the energy domain are not considered directly upon customer insertion. However, to ensure energy feasibility, we repair energy deficits resulting from a customer insertion as follows. We define an energy deficit $ED(i,j,k,s)$ between locations i and j in route the route of vehicle k under scenario s as the difference between the SoC upon arrival at j and the energy required to reach both j as well as the closest (in the energy domain) charger f in its vicinity:

Definition 20 – Energy Deficit

$$ED(i,j,k,s) \coloneqq y_i^{ks} - (h_{ij}^{ks} + \min_{f \in \mathcal{F}} h_{jf}^{ks}), \tag{3.110}$$

where $h_{ij}^{ks} \coloneqq h(d_{ij}, t_{ij}^s, m(u_j^k))$ and $h_{jf}^{ks} \coloneqq h(d_{fj}, t_{fj}^s, m(u_j^k))$.

By resolving energy deficits after each customer insertion, we ensure that, for feasible problem instances, the routes generated by our ALNS heuristic **always remain energy feasible**. However, this may come at the expense of introducing time window violations.

Similar to the work by Keskin and Çatay [31], we first consider the previous charger in the route if any is present and try to resolve energy deficits by increasing the amount of energy charged there. What is novel is that we also keep track of the charger bookings in each of the scenarios to prevent charging overlap as much as possible. We check the timing interval required to resolve the energy deficit by obtaining a high enough SoC at the previous charger. If there are any charger bookings that overlap that interval, we only recharge until the earliest start of such a booking, resolving the energy deficit only partially and for every scenario separately. This addition is especially useful to extend initial charging sessions whenever the chargers are still available so that no new and more costly on-the-go chargers need to be inserted.

In case an energy deficit remains for at least one of the RMP scenarios, we proceed by greedily inserting a new charger into the route. Like in the construction heuristic, we first sort the chargers by insertion cost, and then check whether the energy deficit can be resolved by booking a new charging session at the earliest possible start time, respecting non-overlapping constraints, so that the vehicle can still arrive in time at the next customer. If that is not possible, we move to the next charger. If no chargers satisfy the constraints, we relax the time window restrictions, falling back to the reachable charger with the smallest insertion cost to address the energy deficit. Multiple chargers may need to be inserted between i and j to eliminate the energy deficit $ED(i, j, k, s)$. In this case, after adding the first charger f, we only consider adding other chargers f' with $h^{ks}_{f'j} < h^{ks}_{fj}$ to ensure that the vehicle is getting closer to the goal of reaching j. After repairing the energy deficits, we try to push the initial charging session at the depot charger forward in time as much as possible, maintaining the arrival time at the first customer in line. This helps reduce additional time costs incurred by leaving the depot too early, so that a paid driver has to wait at the location of the first customer for it to begin service, while waiting at the depot without any driver before departure would save time costs. Following each insertion, we perform a propagation procedure similar to that used during removal, updating the second-stage decisions to account for the newly introduced detour and/or the resulting changes in load levels for the entire route. The worst-case time complexity of the charger insertion procedure including propagation is $\mathcal{O}(V \cdot S \cdot F)$, which leads to an overall complexity of $\mathcal{O}(N \cdot (NV \cdot + S \cdot F))$ or $\mathcal{O}(V^3 \cdot S)$ for the repair phase, when paired with greedy customer insertion.

3.4.6. Adding Scenarios to the RMP

In subsequent iterations of the RMP, the use of the construction heuristic is no longer necessary. Instead, we initialize the process using the previous RMP solution $(\boldsymbol{x}, \{\dots, \boldsymbol{y}^r\})$. For the newly introduced scenario p, we set the departure from the depot to timestep 0 for every route R_k, while replicating the charging amount decisions for chargers $f \in \mathcal{F}$ on R_k from the previously added scenario $\boldsymbol{y}^r$. We then propagate the associated timing and state-of-charge levels along the routes. Any resulting violations are incorporated into the generalized cost function and fixed by the repair heuristic after application of the destroy operator. The worst-case time complexity of this procedure is $\mathcal{O}(V)$.

3.4.7. Adaptive Operator Selection

In ALNS, the adaptive selection of operators plays a crucial role in guiding the search process. Using the framework by Wouda and Lan [189], we employ the roulette wheel scheme to dynamically adjust the weights of the destroy operators based on their past performance. Since we only employ a single repair operator, it will always be selected. At the start of the algorithm, each destroy operator o is initialized with a weight $\omega_o = 1$. During each iteration, an operator is selected probabilistically, with its selection probability proportional to its normalized weight $\bar{\omega}_o$. The quality of the candidate solution after repair is evaluated using the generalized cost function from Definition 18, resulting in one of four possible outcomes:

1. The candidate is a new global best.
2. The candidate improves upon the current solution but is not globally best.
3. The candidate is accepted (but not better).
4. The candidate is rejected.

Each outcome $j \in \{1, 2, 3, 4\}$ is associated with a score s_j. We ensure $s_1 > s_2 > s_3 > s_4$, so that higher scores correspond to a better performance. The weight of the selected operator o is then updated using a convex combination of the current weight and the score

$$\omega_o \leftarrow \theta \omega_o + (1 - \theta) s_j, \tag{3.111}$$

where $\theta \in [0, 1]$ is the memory decay rate, controlling the influence of past versus current performance [189].

3.4.8. Acceptance Criterion

To avoid premature convergence of the ALNS algorithm and promote thorough exploration of the solution space, various acceptance criteria for candidate solutions can be employed. Santini, Røpke, and Hvattum

[190] investigated several such criteria across standard ALNS benchmark problems, including the CVRP. Based on their recommendations, we adopt the Record-to-Record Travel (RRT) acceptance criterion introduced by Dueck [191]. To minimize the need for parameter tuning, we implement a linear acceptance threshold that gradually decreases to zero over time.

The RRT criterion is similar to the Simulated Annealing (SA) metaheuristic discussed in Subsection 2.3.2 in the sense that it allows for the acceptance of non-improving solutions to promote the exploration of the search space. However, instead of using a probabilistic acceptance rate, RRT accepts candidate solutions if the absolute difference between their cost and that of the global best solution is less than a threshold T, which is dynamically updated in each iteration to gradually tighten the acceptance criterion. T is updated using

$$T \leftarrow \max\{0,\ T - \gamma\}, \tag{3.112}$$

with γ being the step size or decay factor. T is initially set to T_{start}, which is a parameter that can be tweaked next to γ.

This combination of adaptive operator selection with a flexible acceptance mechanism enables ALNS to balance intensification and diversification effectively, improving its ability to escape local optima and converge towards high-quality solutions [189, 190].

3.4.9. Full ALNS Metaheuristic for Solving the RMP

The full ALNS metaheuristic for solving the RMP is outlined in Algorithm 4.

The algorithm begins by checking for the existence of a previous RMP solution. If one is available, second-stage solutions for the newly introduced scenario p are copied from the previously added scenario r and propagated using the AddScenario procedure. Otherwise, an initial solution is constructed using the described nearest neighbour ConstructionHeuristic.

A StoppingCriterion determines whether to proceed with another destroy-repair cycle. This criterion is based on either a maximum number of iterations or a maximum number of iterations without improvement, as specified per experiment.

Next, one of the DestroyOperators is selected using the RouletteWheelSelection scheme to destroy the current solution $(\boldsymbol{x}, \{\boldsymbol{y}^s\})$. The RepairOperator then greedily inserts the customer and chargers into the destroyed solution $(\boldsymbol{x}', \{\boldsymbol{y}'^s\})$, ensuring energy feasibility.

The acceptance of the resulting candidate solution $(\boldsymbol{x}'', \{\boldsymbol{y}''^s\})$ is based on its generalized cost and governed by the RRTAcceptanceCriterion, which returns one of the scores (s_1, s_2, s_3, s_4) for the selected destroy operator based on the outcome. This score is then used to update the weight of the chosen operator o via Eq. (3.111). The acceptance threshold for RRT is updated using Eq. (3.112).

The remaining steps handle bookkeeping, ensuring that the current solution $(\boldsymbol{x}, \{\boldsymbol{y}^s\})$ and the best-known solution $(\boldsymbol{x}^*, \{\boldsymbol{y}^{s*}\})$ are properly tracked. The best solution is updated whenever a candidate solution $(\boldsymbol{x''}, \boldsymbol{y}''^s)$ is accepted by the acceptance criterion and ultimately returned.

Provided that the StoppingCriterion is set to a fixed maximum number of iterations, the worst-case time complexity of the full procedure becomes $\mathcal{O}(V^3 \cdot S)$. In other words, the metaheuristic is pseudo-polynomial because S can grow exponentially, rather than polynomially, with respect to the dimension l of the uncertainty set $\mathcal{U}$, even when $\bar{\mathcal{U}} = ext(\mathcal{U})$. However, as our results later show, the number of RMP scenarios typically remains limited, so that the pseudo-polynomial runtime does not compromise performance too much.

Algorithm 4 – ALNS Metaheuristic for Solving the RMP

```
Input: Restricted uncertainty set $\bar{\mathcal{U}}$, first-stage decisions $\boldsymbol{x}$, second-stage decision
       candidates $\{\boldsymbol{y}^s\}$, stopping criterion StoppingCriterion, acceptance starting
       threshold $T_{start}$, acceptance decay factor $\gamma$, operator scoring vector
       $\boldsymbol{s} = (s_1, s_2, s_3, s_4)$, operator decay factor $\theta$
Output: Best generalized cost $c^*_{gcost}$, best first-stage decisions $\boldsymbol{x}^*$, second-stage candidate
        solutions $\{\boldsymbol{y}^{s*}\}$
if $\boldsymbol{x} = \boldsymbol{0}$ then
    $(\boldsymbol{x}, \{\boldsymbol{y}^s\}, c_{gcost}) \leftarrow$ ConstructionHeuristic($\boldsymbol{z}^{nom}$);
end
else
    $\boldsymbol{y}^r \leftarrow \boldsymbol{y}^s$ for previously added $s \in \mathcal{S}(\bar{\mathcal{U}})$;
    $\boldsymbol{y}^p \leftarrow$ AddScenario($\boldsymbol{x}, \boldsymbol{y}^r$);
    $\{\boldsymbol{y}^s\} \leftarrow \{\dots, \boldsymbol{y}^r\} \cup \{\boldsymbol{y}^{\boldsymbol{p}}\}$;
    $c_{gcost} \leftarrow GC(\boldsymbol{x}, \{\boldsymbol{y}^{\boldsymbol{s}}\})$;
end
$\boldsymbol{\omega} \leftarrow \boldsymbol{1}$;
$\boldsymbol{x}^* \leftarrow \boldsymbol{x}$;
$\{\boldsymbol{y}^{s*}\} \leftarrow \{\boldsymbol{y}^{\boldsymbol{s}}\}$;
$i = 1,\ i^* = 1$;
$c^*_{gcost} = c_{gcost}$;
$T \leftarrow T_{start}$;
while not StoppingCriterion($i, i^*$) do
    $(o,$ DestroyOperator$) \leftarrow$RouletteWheelSelection($\boldsymbol{\omega}$);
    $(\boldsymbol{x}', \{\boldsymbol{y}'^s\}, \mathcal{L}) \leftarrow$ DestroyOperator($\boldsymbol{x}, \{\boldsymbol{y}^s\}$);
    $(\boldsymbol{x}'', \{\boldsymbol{y}''^s\}) \leftarrow$RepairOperator($\boldsymbol{x}', \{\boldsymbol{y}'^s\}, \mathcal{L}$);
    $(s_j, \mathit{Accepted}) \leftarrow$RRTAcceptanceCriterion($c_{gcost}, c^*_{gcost}, T, \boldsymbol{s}$);
    $\omega_o \leftarrow \theta\omega_o + (1 - \theta)s_j$;
    $T \leftarrow \max\{0, T - \gamma\}$;
    if $\mathit{Accepted}$ then
        $\boldsymbol{x} \leftarrow \boldsymbol{x}''$;
        $\{y^s\} \leftarrow \{\boldsymbol{y}''^s\}$;
        $c_{gcost} \leftarrow GC(\boldsymbol{x}'', \{\boldsymbol{y}''^s\})$;
        if $c_{gcost} < c^*_{gcost}$ then
            $\boldsymbol{x}^* \leftarrow \boldsymbol{x}''$;
            $\{y^{s*}\} \leftarrow \{\boldsymbol{y}''^s\}$;
            $c^*_{gcost} \leftarrow c_{gcost}$;
            $i^* \leftarrow i$;
        end
    end
    $i \leftarrow i + 1$;
end
return $(c^*_{gcost}, \boldsymbol{x}^*, \{\boldsymbol{y}^{s*}\})$
```

3.5. Solver Synthesis

Having introduced all key methodological components, we now synthesize several solver variants as part of the CertℰRoute framework to address the HGEV-FPP-TECU under different problem settings.

Exact Deterministic Solver (EXA-DET):

- Solves Model 13 directly for a single scenario q, using the uncertainty set $\bar{\mathcal{U}} = \{\boldsymbol{z}^q\}$.
- Returns the total cost c_{total} and static solution $(\boldsymbol{x}, \boldsymbol{y}^q)$.
- Guarantees optimality under the given scenario.

Exact Adaptive Robust Optimization Solver (EXA-ARO):

- Solves Model 13 taking the role of the RMPSolver within the CCG framework from Algorithm 1.
- Returns the total cost c_{total}, first-stage decisions $\boldsymbol{x}$, and second-stage decision candidates $\{\boldsymbol{y}^s\}$ for all scenarios $s \in \mathcal{S}(\bar{\mathcal{U}})$.
- Guarantees optimality under $\bar{\mathcal{U}}$ but might suffer from lacking robustness due to premature convergence.

Exact Robust Optimization Solver (EXA-RO):

- Similar to EXA-ARO but considers a single second-stage decision candidate $\boldsymbol{y}$ to perform static robust optimization.
- Guarantees optimality under $\bar{\mathcal{U}}$ but might suffer from lacking robustness due to premature convergence.

Metaheuristic Deterministic Solver (META-DET):

- Similar to EXA-DET but employs the ALNS procedure from Algorithm 4 to find the static solution.
- Provides no optimality under $\bar{\mathcal{U}}$ and might suffer from lacking robustness due to premature convergence.

Metaheuristic Adaptive Robust Optimization Solver (META-ARO):

- Similar to EXA-ARO but employs the ALNS procedure from Algorithm 4 to solve the RMP.
- Guarantees no optimality under the given scenario.

Justification for Deterministic Solver Variants

The inclusion of the deterministic solvers EXA-DET and META-DET addresses practical scenarios where only nominal or average-case estimates are available. When robustness is not a priority, or costs of expected failure can be anticipated and incorporated in the objective function, optimizing directly for the nominal scenario can yield cost-efficient and pragmatic solutions.

In the adaptive optimization setting, the optimal second-stage decisions $\boldsymbol{y}^q$ for a realized scenario q may not be part of the candidate set $\{\boldsymbol{y}^s\}$, as the objective in Eq. (3.2) focuses on minimizing the worst-case operational cost. Consequently, it suffices to identify feasible, though not necessarily optimal, second-stage decision candidates for less adverse scenarios. The deterministic solvers can be used post-realization to refine these decisions.

Justification for Static Solver Variant

While the adaptive solver variants EXA-ARO and META-ARO offer the flexibility to adjust second-stage decisions during route execution based on the realized scenario, potentially enhancing cost-efficiency, this adaptability may not always be practical. It relies on accurate real-time tracking of vehicle energy consumption and timing realizations. Inaccurate measurements can compromise robustness, and constantly shifting optimal timing decisions may prevent reliable arrival time estimates, complicating customer communication and integration with reservation systems for chargers. In such cases, a non-adaptive solver like EXA-RO may be more suitable, providing a single, robust solution with stronger guarantees, albeit potentially at the cost of reduced cost-efficiency.

4

Experimental Setup

This chapter focuses on the experimental setup required to address the research sub-question:

Research Sub-question 4

How do the proposed solution methods perform when applied to realistic HGEV fleet planning scenarios, in terms of robustness, cost-efficiency, scalability, and adaptability?

Section 4.1 details the process of generating realistic HGEV-FPP-TECU instances. Section 4.2 describes how we model the energy consumption of the HGEVs in our experiments. Section 4.3 covers the implementation and configuration of the models and algorithms, as well as the hardware setup used during experimentation. The actual experiments and their results are discussed in detail in Chapter 5.

4.1. Instance Generation

To evaluate our methodology, we introduce five distinct classes of synthetic yet realistic HGEV-FPP-TECU instances, each containing five benchmark instances. The first two classes, Toy Single Vehicle (T-SV) and Toy Dual Vehicle (T-DV), include small-scale, illustrative instances involving one and two vehicles, respectively. These toy instances are solvable to optimality using exact methods and are particularly useful for visual interpretation and analysis, which becomes challenging with larger instances.

The Small (S) class simulates operations of small companies managing a limited number of HGEVs, each serving up to four customers a day. For simplicity, these instances include only a single depot charger, such that the satisfaction of non-overlap constraints for charging sessions can easily be verified.

The Medium (M) and Large (L) classes represent more complex instances with 30 or more customers and 10 or more vehicles. These are typically computationally intractable for exact methods and are thus used to assess the performance of our metaheuristic solution methods. An overview of the key design parameters for all instance classes is provided in Table 4.1.

Class	Customers	Vehicles	Depot Chargers	On-the-Go Chargers	Demand UB
T-SV	3	1	1	2	$0.7 \cdot C_{max}$
T-DV	5	2	1	2	$0.7 \cdot C_{max}$
S	10	3	1	5	$0.7 \cdot C_{max}$
M	30	12	10	20	$0.5 \cdot C_{max}$
L	100	40	30	30	$0.3 \cdot C_{max}$

Table 4.1: Instance classes and their design parameters. C_{max} denotes the maximum vehicle load capacity.

The procedure for sampling charger and customer locations is detailed in Subsection 4.1.1. Baseline vehicle parameters are described in Subsection 4.1.2. The construction of the uncertainty set for each instance is outlined in Subsection 4.1.3. Subsection 4.1.4 provides an overview of the cost parameters adopted for the experiments.

4.1.1. Location Sampling

To make the instances as realistic as possible, we sample locations that are commonly visited by transport companies in actual operations. To do this, we utilize a synthetic dataset derived from European road freight traffic, comprising 1,514,573 directed transport flows between 1,630 origin points and 1,667 destination points all across Europe, including the Netherlands. It provides data on freight volumes in tons as well as the number of vehicles involved [192]. Specifically for 2019, the dataset includes 385 destination locations within the Netherlands. We construct a probability distribution over these data points, weighting every location by the total tonnage transported to that destination and used this to sample customer locations for our instances. We sample the location of the depot at the center of the bounding box containing The Netherlands, which corresponds to a location in the village Eemnes (Noord-Holland). For the Large instance class, we pose the constraint that the sampled customers can only be located in the eastern half of the Netherlands, to improve the likelihood of finding feasible instances. Additionally, the depot is preferentially sampled further east, resulting in its placement in Deventer (Overijssel).

For the chargers, we distinguish between depot chargers, located at the depot and on-the-go chargers, which are situated elsewhere. Among the on-the-go chargers, we further differentiate between three types:

- **Private chargers**: sampled among the sampled customer locations.
- **Semi-public chargers**: sampled from the full set of potential customer locations in the synthetic dataset.
- **Public chargers**: sampled from a set of service area locations.

In reality, the current public charging infrastructure for HGEVs is limited and insufficient to cover the entire country. Sampling public chargers only from the current known charger locations would make the generation of feasible routing instances very challenging. To remedy this, we instead sample potential charger locations from a set of highway service areas, which are considered to be representative for future charger locations based on a policy report prepared for the Dutch government. This report specifically recommends service area locations for the expansion of public charging infrastructure [193]. We extracted the coordinates of all current highway service areas from Wikipedia [194] using a Python script and compiled these into a dataset from which the charger locations are sampled uniformly at random.

We apply sampling weights of $[0.1, 0.8, 0.1]$ for the on-the-go charger categories consistently across all experiments. This ensures that smaller instances predominantly feature public chargers, while larger instances incorporate a more diverse mix. An illustrative example corresponding to instance S-1 demonstrates the outcome of our location sampling method, as shown in Fig. 4.1.

Customer demand is sampled between 10% and an upper bound, which is initially set at 70% of the maximum vehicle load capacity C_{max}, following Wang et al. [69]. For larger instances, this upper bound is reduced to increase the likelihood of generating capacity-feasible instances. If the total demand exceeds the fleet's total load capacity, the upper bound is iteratively decreased by 5%, resampling customer demands until a capacity-feasible set of demands is found.

The operation window is assumed to span the full day, from 00:00 to 24:00. Time windows $[e_i, l_i]$ for the depot and chargers $i \in \mathcal{F}' \cup \{0, N+1\}$ are set accordingly, assuming charger availability unless booked by other vehicles. For customer locations $i \in \mathcal{C}$, time window lengths are randomly drawn between 3 and 6 hours, with earliest start times e_i sampled between 06:00 and 18:00 to represent typical opening hours for customers.

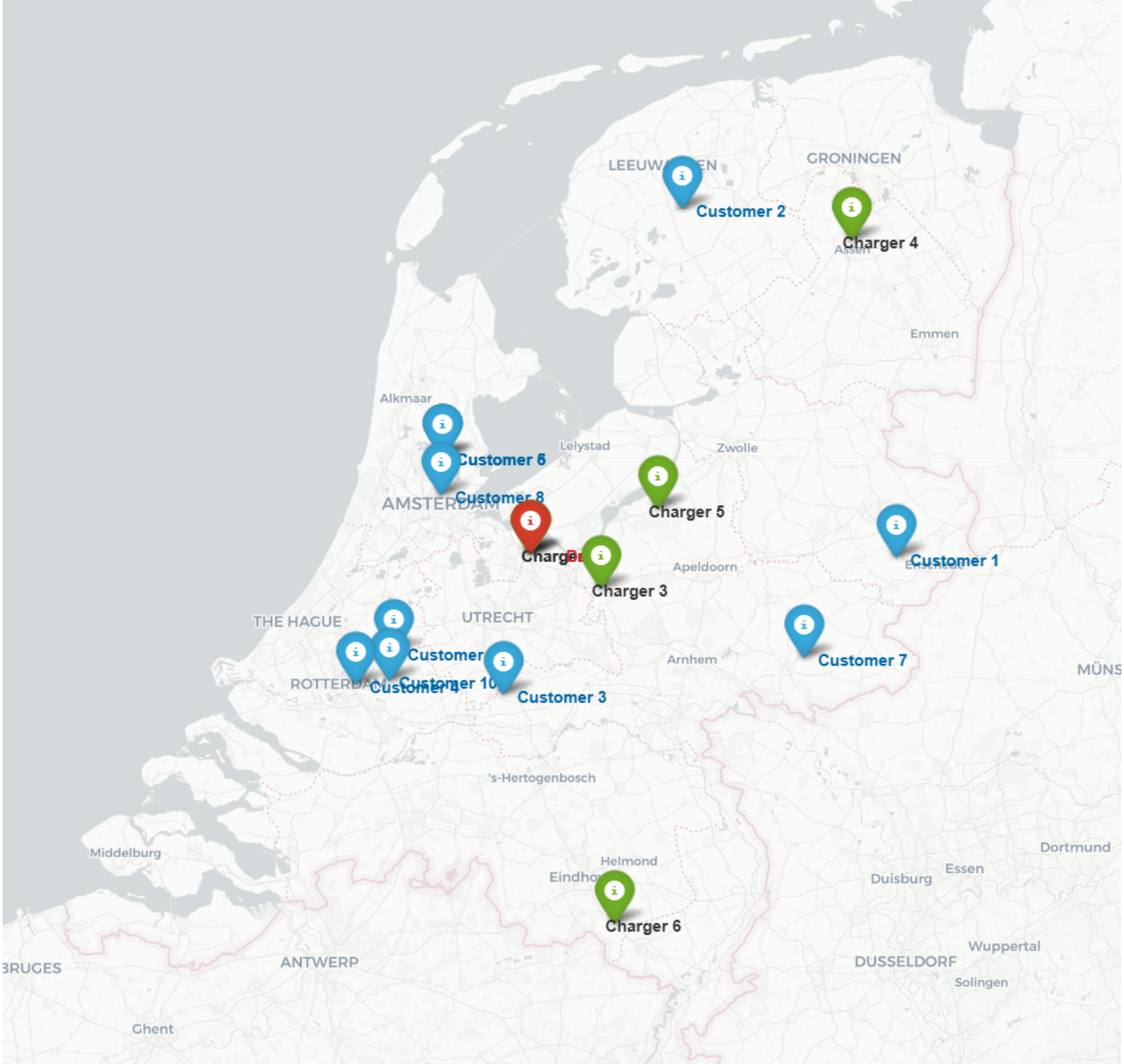


Figure 4.1: Example of sampled locations with the depot location indicated in red, charger locations in green and customer locations in blue. The sampled locations correspond to those in the instance S-1.

4.1.2. Crafting a Realistic Baseline Vehicle

To further promote the realism of our experiments, we construct a representative baseline vehicle informed by a brief market analysis and actual configurations observed among Shell clients. According to the International Council on Clean transport, long-haul tractor-trailer combinations account for approximately 75% of the CO_2 emissions produced by the entire HGV fleet. The predominant configuration in this segment features a 4 × 2 drivetrain and a gross combination weight of 40 tonnes [136]. Electrification of this vehicle type is progressing rapidly, with all seven major European HGEV manufacturers currently offering at least one electric model [195].

The Tractor

Volvo has led the European market in this segment for the past five years, achieving a 47% market share in 2024 [196]. Its flagship model, the Volvo FH Electric, as shown in Fig. 4.2, exhibits an average energy consumption of 1.1 kWh/km and a range of up to 300 kilometres. It delivers a power output between 330 and 490 kW, supports charging at 250 kW, and accommodates 4 to 6 batteries for a total capacity of 360–540 kWh [197]. The curb mass of the vehicle is 9,890 kg.

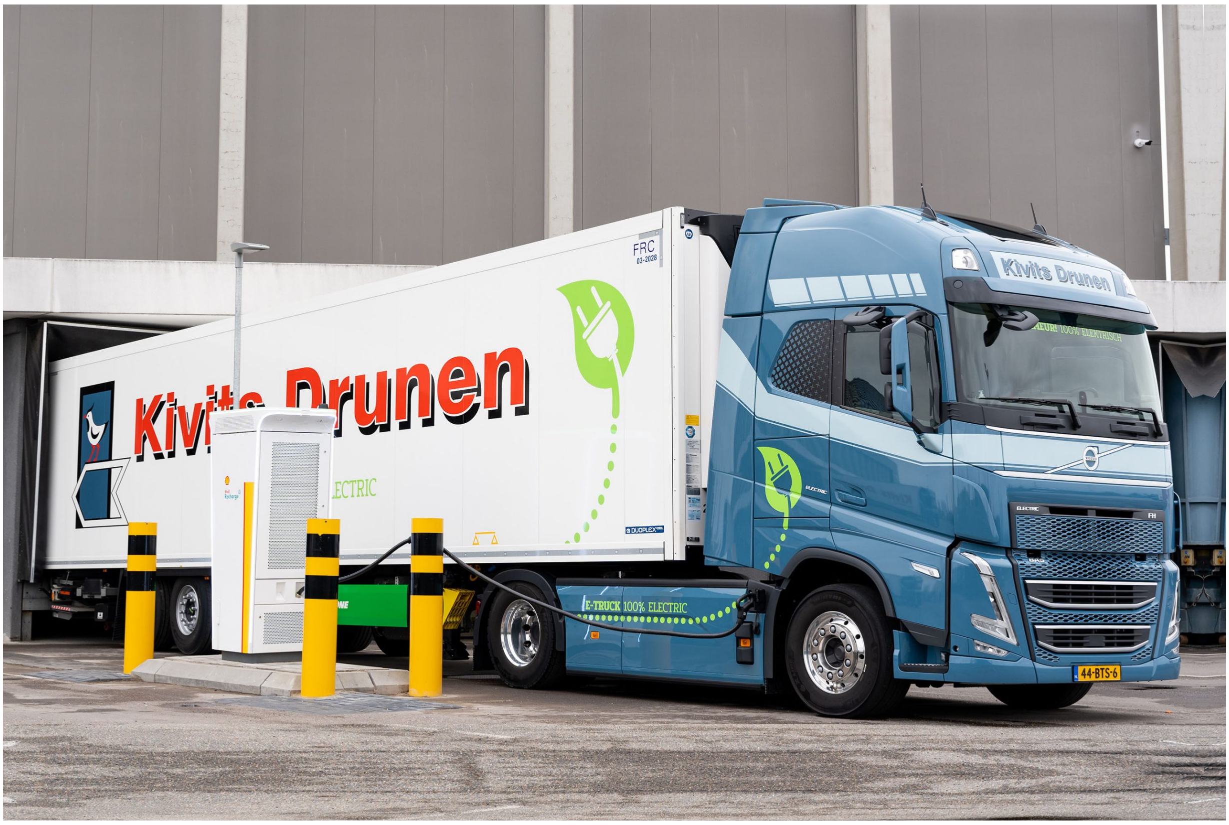


Figure 4.2: Volvo FH Electric Tractor coupled to a Shell AC Fast Charger. Image published by Shell

The Trailer

In the trailer segment, Schmitz Cargobull holds a leading position in Europe with a market share of approximately 23% [198]. Of particular relevance is its fully electric refrigerated S.KOe COOL trailer [199], illustrated in Fig. 4.3, which is frequently used in for example supermarket distribution applications. Although the trailer is equipped with a 32 kWh battery that enables up to 4.5 hours of standalone operation, a Shell client in the Netherlands chose to power the 7,980 kg trailer directly from the Volvo FH Electric tractor via its Power Take-Off (PTO) system. This results in a demanding but interesting scenario, where the refrigeration unit imposes an additional continuous auxiliary power demand of 7.1 kW [200].

Figure 4.3: Schmitz S.KOe COOL Trailer. Image adapted from [200]

We select the Volvo FH Electric, configured with a 540 kWh battery and coupled to the Schmitz S.KOe COOL trailer, as the baseline vehicle for our experimental study. Note that our model formulation from Model 11 allows for a heterogeneous fleet of vehicles. However, for our experimental study, we only consider a single vehicle type to begin with. Due to the unavailability of exact data on the vehicle's frontal area, auxiliary power demand, and tire rolling resistance, standard reference values are adopted from Oscar Delgado and Muncrief [136]. For the driver, we consider a maximum mass of 100 kg. An overview of all relevant baseline vehicle parameters and their values is provided in Table 4.2.

Usable Battery Range

The usable battery range of HGEVs is often guided by the 20 to 80% rule, which recommends maintaining the battery's SoC within this range to extend battery lifespan and account for uncertainties in energy consumption. However, we aim to explicitly account for energy consumption uncertainty in our optimization such that applying this rule on top of that would make our already conservative model, even more restrictive. To strike a better balance between realism and robustness, we instead adopt a usable SoC range of 10 to 90%. This approach avoids excessive conservatism while still preserving a buffer for unexpected events that are difficult to predict or model accurately. Mathematically, applying this rule is equivalent to reducing

Parameter	Description	Value
c_r^k	Rolling resistance coefficient	0.0055
$(c_d \cdot A_f)_k$	Air drag area [in m^2]	5.63
m_c^k	Total curb mass [in kg]	$9{,}890 + 7{,}980 + 100$
Q_k	Battery capacity [in kWh]	$0.8 \cdot 540$
g_k	Inverse charging rate [in minutes/kWh]	0.24
P_0^k	Auxiliary power demand [in kW]	$5.5 + 7.1$

Table 4.2: Baseline Vehicle Parameters

the battery capacity to 80%, while maintaining the SoC within 0 to 100% of this reduced capacity. This explains the multiplier 0.8 for the battery capacity in Table 4.2.

4.1.3. Bounding Uncertainty

To construct our uncertainty sets, we establish realistic bounds on the uncertainties in the time domain, which include the service and waiting durations, charging rates, and traversal durations. In the following, we detail this process, which is based on informed assumptions and expert consultation. The resulting parameter values are summarized in Table 4.3.

Service and Waiting Durations

We utilize a private case study conducted by Shell for a major road transport client in the Netherlands to gain insights into unloading durations. The case study shows an average waiting duration of 27.5 minutes, with over 50% of the values lying within the range of 15 to 40 minutes and more than a quarter of the values falling between 15 and 20 minutes. For the construction of our uncertainty set, we align the lower and upper bounds with the found first and third quartile, considering waiting durations between 15 and 40 minutes. This corresponds to setting ν_i^{nom} to $\mathbf{27.5}$ and ν_i^{dev} to $\mathbf{12.5}$ for all locations $i \in \mathcal{C} \cup \{0\}$, so that the nominal value within the uncertainty set is close to the mean of the actual distribution.

We assume that the bounds for unloading at customers can also be used to bound the loading duration at the depot, for which no data is available. This assumption is based on the premise that a full truck load can often be loaded quickly from a dock using methods such as cross-docking, where cargo is unloaded from incoming vehicles and almost immediately loaded into outgoing vehicles with minimal delay [201]. Other companies prepare the cargo for a Full Truck Load (FTL) at the dock, allowing it to be loaded in one piece using automated truck loading systems. However, during distribution operations, partial unloading at customer locations often cannot be done as efficiently, resulting in significantly longer unloading durations.

As for waiting durations, we envision a scenario where the timeslots for visits to public or semi-public chargers are reserved ahead of time through a booking system, eliminating any waiting time at charging stations. Additionally, we assume that the waiting durations at the depot and unloading locations are already included within the unloading duration. This corresponds to setting $w_i^{nom} = w_i^{dev} = \mathbf{0}$ min for all locations $i \in \mathcal{V}$.

Charging Rates

In our experiments, we consider two charger types: a 50 kW Alternating Current (AC) charger for overnight charging at the depot, and a 400 kW Direct Current (DC) charger aimed at fast charging. These represent typical infrastructure found at Shell eDepots and public charging facilities, respectively. To illustrate, the AC charger could hypothetically fully charge the Volvo FH Electric's 540 kWh battery in approximately 11 hours, whereas the DC charger would only require 1.5 hours to do so.

However, the power levels of the chargers represent their theoretical maximum. For the AC charger, the corresponding C-rate, which gives the ratio of charger power (in kW) to battery capacity (in kWh), is approximately $\frac{50}{540} \approx 0.093$. Such a low C-rate indicates a slow, battery-friendly charging process. As a result, AC charging is typically stable and predictable, justifying the use of a constant charging rate of 50 kW in our experiments. This corresponds to setting $g_f^{k(nom)} = \mathbf{1.2}$, $g_f^{k(dev)} = \mathbf{0}$ min/kWh for all $f \in \mathcal{F}^{AC}$, where $\mathcal{F}^{AC} \subset \mathcal{F}$ denotes the set of all AC chargers.

In contrast, the DC charger, while rated at 400 kW, is limited by the vehicle's maximum intake of 250 kW, yielding a C-rate of $\frac{250}{540} \approx 0.46$. At this higher rate, charging becomes more dynamic. As the battery's SoC increases, internal resistance and heat buildup cause the vehicle's Battery Management System (BMS) to gradually reduce power intake. This results in a tapering charging curve, especially beyond 80 to 90% SoC. However, due to the large battery size of HGEVs, this effect is less pronounced than for passenger BEVs, as elucidated in Fig. 4.4. Hence, for simplicity, we assume a constant DC charging rate of 250 kW during our experiments, which is rather conservative for 400 kWh chargers but reflects the vehicle's charging limitations. This corresponds to setting $g_f^{k(nom)} = \mathbf{0.24}$, $g_f^{k(dev)} = \mathbf{0}$ min/kWh for all $f \in \mathcal{F}^{AC}$, where $\mathcal{F}^{DC} \subset \mathcal{F}$ denotes the set of all DC chargers.

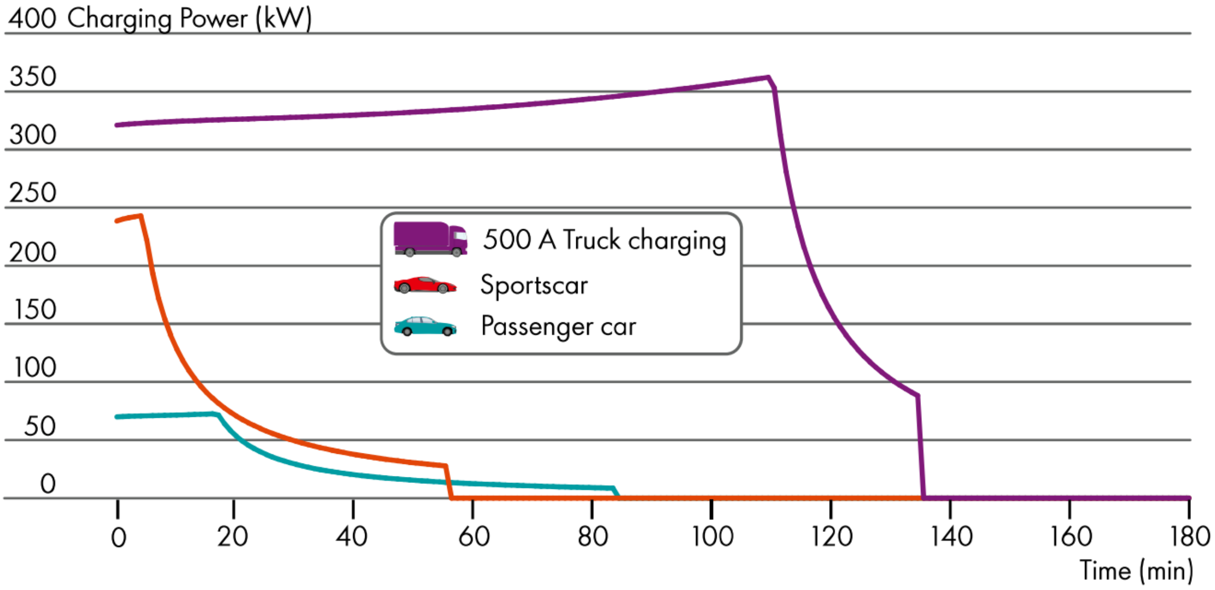


Figure 4.4: Synthetic charging curves showing the relation between charging power and time comparing HGEVs with passenger BEVs. Figure adapted from Liebig et al. [202]

Within our scope, overnight charging at public and semi-public chargers is not considered relevant, since the HGEVs are assumed to complete a single trip per day and remain at the depot overnight. As such, only DC fast chargers are deployed at these locations. Due to the current limited deployment of eDepot infrastructure, statistical data on the distribution of charger types at depots is not yet available. Therefore, we assume that **60**% of future depot chargers will be AC chargers for overnight charging, while the remaining **40**% will be DC fast chargers. This balance is intended to ensure operational resilience in case of DC charger outages and to support daytime charging needs. We adopt these percentages in our charger sampling procedure, except for the toy instances, where only DC chargers are sampled for simplicity.

Traversal Durations

According to experts in the field, European HGEVs travel around 50 to 60 km/h, accounting for traffic congestion, driving regulations and speed limits. We make use of the open-source OpenRouteService Matrix API developed by Neis and Zipf [203] to determine the edge distances d_{ij}^{API} and traversal durations t_{ij}^{API} between all sampled locations $i, j \in \mathcal{V}$. This API uses Dijkstra's Algorithm [204] to find the shortest paths from one location to the other based on the traversal time per underlying road segment, as taken from OpenStreetMap (OSM) [205] data. The traversal time for each road segment is calculated using the known speed limits for HGVs on different waytypes, which are then adjusted for the surface types and qualities of the road segments if such annotations are available. If multiple annotations apply to a road segment, the most restrictive one is used [206]. As a small validation experiment, we randomly sampled 100 different locations in the Netherlands and used the OpenRouteService API to calculate the mutual distances and traversal durations, from which we derived the average velocity. The resulting distribution of the average velocity values is shown in Fig. 4.5 and shows ranges between 40 to 60 km/h with both the median and mean value around 50, which corresponds to observations made by field experts.

To account for uncertainty in traversal durations, we utilize statistics published by the Dutch government

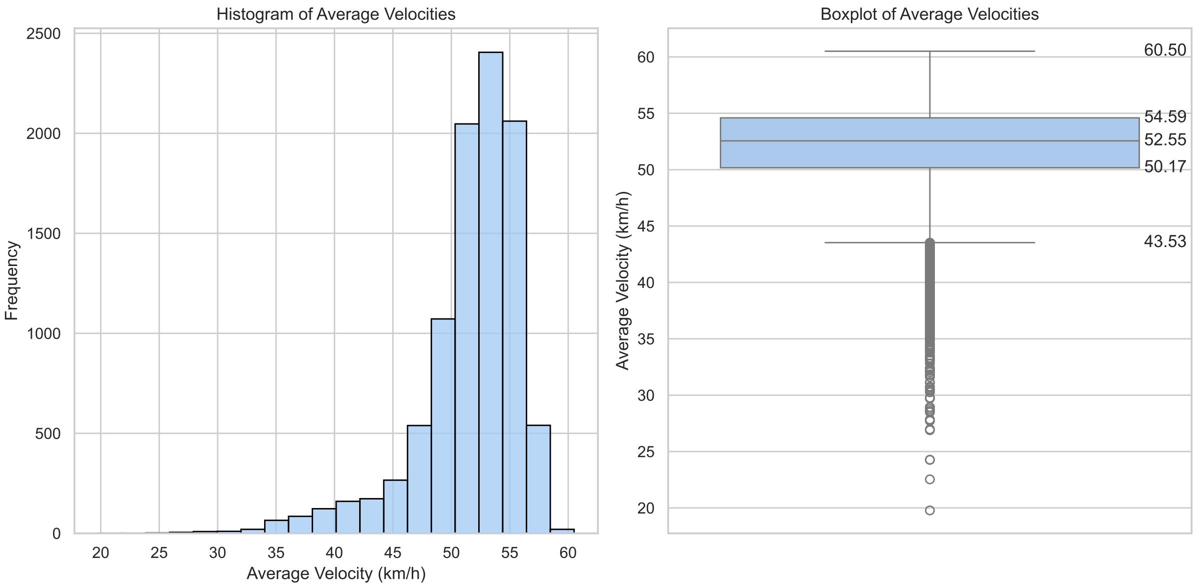


Figure 4.5: Average velocities for all possible trajectories between 100 randomly sampled locations in the Netherlands calculated using the OpenRouteService Matrix API

[207] on the annual additional travel hours incurred due to driving below the reference speed of 100 km/h in the Netherlands. However, the maximum speed of HGVs in the Netherlands is restricted to 80 km/h, which means they always experience additional travel time according to this metric. Despite this limitation, this metric provides some insight into the potential impact of road congestion, for which no specific measures exist specifically for HGVs. The data indicates that in 2024, 62 million hours were lost over a total of 69 billion kilometres driven annually. This translates to an additional 3.23 seconds per kilometre. For our uncertainty set, we consider the traversal durations returned by the API as the lower bound and construct an upper bound by multiplying the distance by the extra travel duration of **0.0538** minutes per kilometre.

Table 4.3 summarizes the resulting parameters for the construction of the uncertainty sets. The uncertainty budgets are determined per experiment in Chapter 5.

Parameter	**Description**	**Value**	**Indices**
d_{ij}	Traversal distances [in km]	d_{ij}^{API}	$i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1}$
t_{ij}^{nom}	Nominal traversal duration [in min]	t_{ij}^{API}	$i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1}$
t_{ij}^{dev}	Maximum deviation in traversal duration [in min]	$0.0538 \cdot d_{ij}$	$i \in \mathcal{V}'_0,\ j \in \mathcal{V}'_{N+1}$
ν_i^{nom}	Nominal service duration [in min]	27.5	$i \in \mathcal{C} \cup \{0\}$
ν_i^{dev}	Maximum deviation in service duration [in min]	12.5	$i \in \mathcal{C} \cup \{0\}$
$g_f^{k(\text{nom})}$	Nominal AC inverse charging rate [in min/kWh]	1.2	$f \in \mathcal{F}^{AC},\ k \in \mathcal{K}$
$g_f^{k(\text{dev})}$	Maximum AC deviation in inverse charging rate [in min/kWh]	0	$f \in \mathcal{F}^{AC},\ k \in \mathcal{K}$
$g_f^{k(\text{nom})}$	Nominal DC inverse charging rate [in min/kWh]	0.24	$f \in \mathcal{F}^{DC},\ k \in \mathcal{K}$
$g_f^{k(\text{dev})}$	Maximum DC deviation in inverse charging rate [in min/kWh]	0	$f \in \mathcal{F}^{DC},\ k \in \mathcal{K}$
w_j^{nom}	Nominal waiting duration [in min]	0	$j \in \mathcal{V}$
w_j^{dev}	Maximum deviation in waiting duration [in min]	0	$j \in \mathcal{V}$

Table 4.3: Parameters used to define the uncertainty sets

4.1.4. Setting Cost Parameters

Throughout our experiments, the main objective is to minimize the total operational costs, as defined by Eq. (3.2) and embedded in the generalize cost function (Definition 18) for ALNS. We consider a distance cost c^k_{km} of **0.08** €/km and a time cost c^k_{min} of **0.83** €/min (equivalent to €50 per hour for a driver) for each vehicle $k \in \mathcal{K}$. Additionally, we account for an energy cost c^f_{kWh} of **0.25** €/kWh for all private depot chargers $f \in \mathcal{F}''$ and a cost of **0.60** €/kWh for all on-the-go chargers in $f \in \mathcal{F}' \setminus \mathcal{F}''$, based on cost evaluations made by the Shell sales team.

4.2. Energy Consumption Modelling

As discussed in Subsection 4.2.1, energy consumption is inherently complex and uncertain. To address this, we adopt a modelling approach based on simulation and function approximation. This method allows us to account for energy-related uncertainties, without explicitly including them into the optimization problem. Based on a large number of energy consumption simulations, we approximate a function that yields the expected energy consumption for every edge in the transport network, given a realization of the uncertain traversal duration, load level, and distance over that edge. By multiplying the function values with a certain robustness multiplier, the level of robustness to uncertainties in the energy domain can be tweaked. The following two subsections elaborate on this methodology.

4.2.1. Simulation of the Energy Consumption

While we aim to capture the significant [51] impact of acceleration and deceleration on HGEV energy consumption, we do not perform high-fidelity simulations over the road network $\mathcal{E}_r$ due to the lack of detailed vehicle and traffic simulators and access to paid APIs, unlike in prior works [51, 69, 146].

Instead, we simulate energy consumption using the longitudinal dynamics model from Model 5 across representative HGV duty cycles from VECTO [208], including UrbanDelivery, InterUrban, SubUrban, RegionalDelivery, and LongHaul profiles. These profiles capture varying driving conditions, from low-velocity stop-and-go traffic to continuous high-velocity motorway driving. We preprocess each profile for simulation, with results shown in Fig. 4.6 and further details in Appendix C.

For each duty cycle, we perform 100,000 Monte Carlo (MC) simulations over the uncertain model parameters, fixing vehicle mass at five discrete values from curb weight to 40 tonnes. To account for weather variability, we sample air density values (ρ_a) [137] based on temperatures from −10 °C to 30 °C observed in the 2024 Dutch weather data [209]. Motor and generator efficiencies (η_M, η_G) are varied within the sensitivity ranges from Asamer et al. [6], reflecting environmental and vehicle-specific uncertainties. Auxiliary power demand (P_0) is simulated using the range from Oscar Delgado and Muncrief [136], adjusted for the continuous load of the refrigerated baseline trailer.

We adopt Gaussian distributions for sampling the uncertain parameters, as they more accurately reflect the probabilistic nature of environmental phenomena compared to uniform distributions [210]. Each random variable is modeled with a marginal Gaussian distribution centered at the midpoint of its uncertainty range, with endpoints corresponding to the 99% confidence interval. While we currently assume independence among variables due to the lack of correlation data, we structure the sampling process to support multivariate Gaussian distributions. This is achieved via a Cholesky decomposition [211] of the covariance matrix, which defaults to the identity matrix under the independence assumption. Table 4.4 summarizes the parameters subject to simulation and their associated uncertainty intervals. All other vehicle parameters follow the baseline from Subsection 4.1.2. Each simulation yields the energy consumption per kilometre for a given average velocity and mass level.

Variable	Description	Uncertainty Interval	Distribution
ρ_a	Air density [in kg/m^3]	$[1.341, 1.127]$	$\mathcal{N}(1.246, 0.318)$
η_M	Motor energy efficiency [in %]	$[0.63, 0.90]$	$\mathcal{N}(0.75, 0.0582)$
η_G	Generator energy efficiency [in %]	$[0.64, 0.82]$	$\mathcal{N}(0.70, 0.388)$
P_0	Auxiliary power demand [in kW]	$[13.5, 16.5]$	$\mathcal{N}(12{,}600; 388.22)$

Table 4.4: Sampled energy consumption parameters with their uncertainty ranges and sample distribution

The resulting mean energy consumptions per kilometre and the corresponding standard deviations are summarized in Table 4.5.

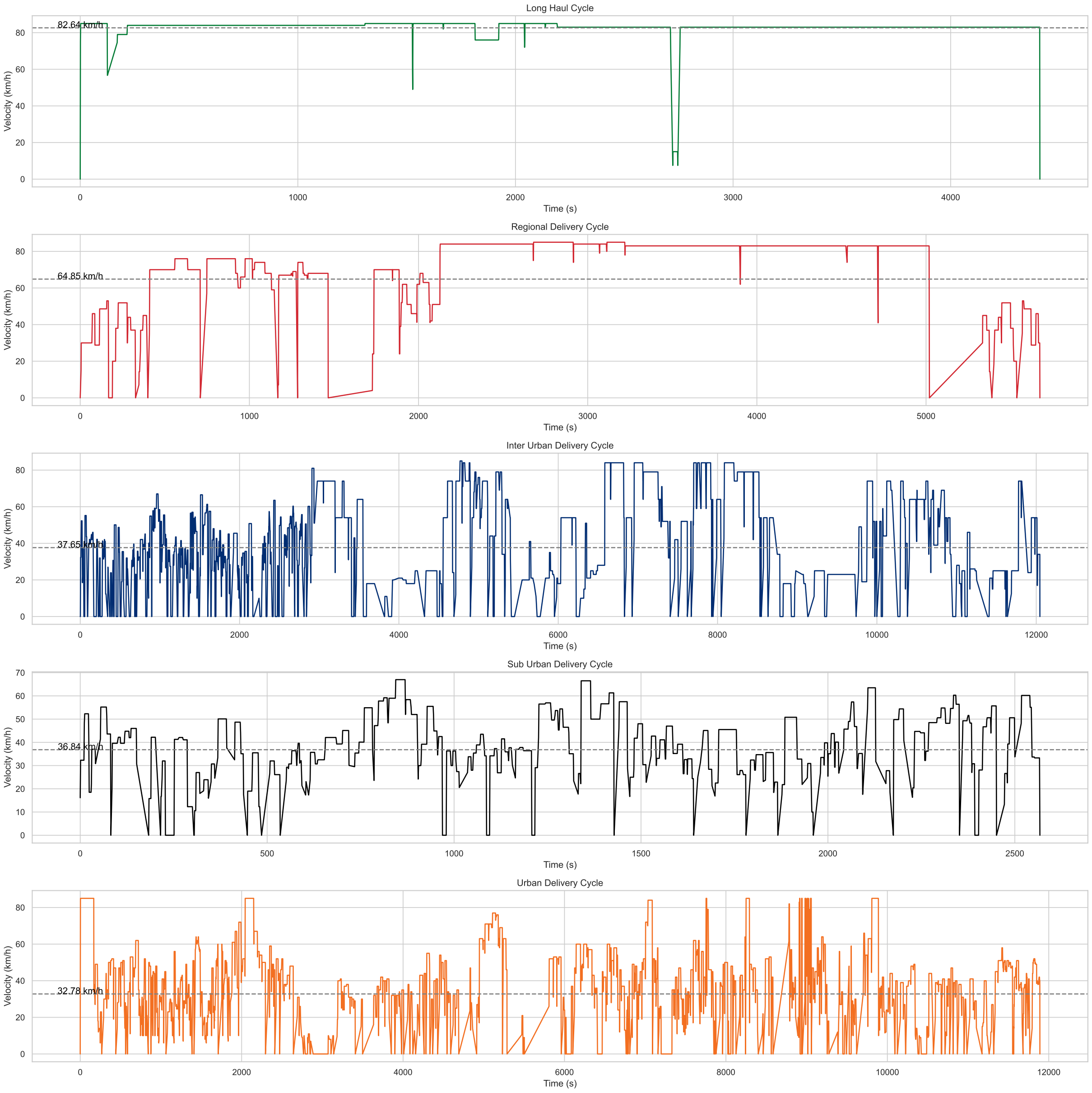


Figure 4.6: Duty cycles for LongHaul, RegionalDelivery, InterUrban, SubUrban and UrbanDelivery delivery operations showing the recorded velocity and elapsed time over different trajectories. The dashed line indicates the characteristic average velocity for the profile in the time domain.

Cycle	Velocity (km/h)	Inv. Velocity (min/km)	Load Level (kg)	Consumption (kWh/km)
LongHaul	82.64	0.726	17,970	0.964 ± 0.068
			25,000	1.004 ± 0.072
			30,000	1.033 ± 0.076
			35,000	1.062 ± 0.078
			40,000	1.091 ± 0.080
RegionalDelivery	64.85	0.925	17,970	1.016 ± 0.073
			25,000	1.098 ± 0.081
			30,000	1.157 ± 0.087
			35,000	1.215 ± 0.093
			40,000	1.274 ± 0.100
InterUrban	37.65	1.594	17,970	1.428 ± 0.111
			25,000	1.719 ± 0.143
			30,000	1.926 ± 0.166
			35,000	2.133 ± 0.189
			40,000	2.340 ± 0.212
SubUrban	36.84	1.629	17,970	1.627 ± 0.130
			25,000	2.032 ± 0.175
			30,000	2.318 ± 0.207
			35,000	2.606 ± 0.238
			40,000	2.894 ± 0.269
UrbanDelivery	32.78	1.830	17,970	1.948 ± 0.158
			25,000	2.441 ± 0.210
			30,000	2.796 ± 0.248
			35,000	3.145 ± 0.286
			40,000	3.499 ± 0.323

Table 4.5: Monte Carlo simulation results presenting the mean and standard deviation of energy consumption per kilometre across various duty cycles and mass configurations

4.2.2. Function Approximation for the Energy Consumption

By aggregating simulation results across different profiles and mass levels, we construct a function approximation that maps inverse velocity and mass to the average energy consumption in kWh/km. The approximation process yields generalized parameters $(\alpha^k, \beta^k, \gamma^k, \kappa^k)$ for the energy function from Definition 15. These are applied uniformly across all instances, vehicles $k \in \mathcal{K}$ and edges $(i, j) \in \mathcal{E}$, as we consider a homogeneous fleet in our experiments and edge-specific simulations would require a velocity profile per road segment.

For the robustness multiplier ψ^k, we adopt the baseline value of 1 during our experiments. This reflects nominal-case optimization with respect to energy consumption uncertainty. Since HGEVs generally operate at average velocities that fall between those of the RegionalDelivery and InterUrban profiles (as shown in Fig. 4.5), and the approximated function already yields a conservative estimate for these profiles (see Fig. 4.7), we consider the estimates for energy consumption given by this function to be robust enough for our experiments. For more conservative planning, fleet operators may opt to use higher coefficients, up to $\psi = 1.45$. Using this coefficient roughly corresponds to applying a chance constraint that limits the risk of battery depletion under energy consumption uncertainty to below 5%, provided that average velocities remain within the simulated range $[32.78, 82.64]$ km/h. However, this approach may significantly overestimate energy consumption and operational costs at lower average velocities.

The final energy function used in our experiments corresponds to the one defined in Definition 15, with the estimated parameters applied. The corresponding Mean Squared Error (MSE) and the coefficient of determination (R^2) were found to be **0.02** and **0.95**, respectively, indicating that the function represents the simulated data well. Further details on the approximation process can be found in Appendix C.

$$h(d_{ij}, t^s_{ij}, m(u^k_j)) = (0.963 - 3.970 \times 10^{-5} m(u^k_j) - 1.125 \times 10^{-1} \frac{t^s_{ij}}{d_{ij}} + 5.511 \times 10^{-5} m(u^k_j) \frac{t^s_{ij}}{d_{ij}}) 1 d_{ij} \tag{4.1}$$

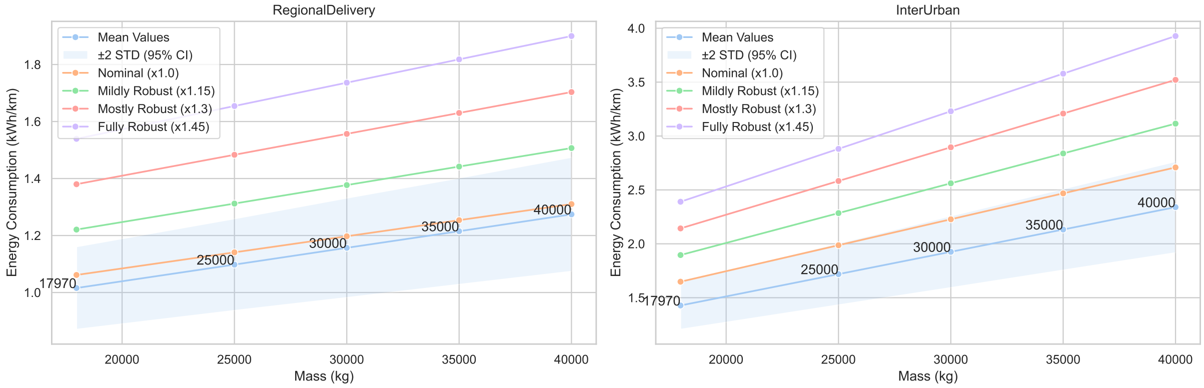


Figure 4.7: Simulated and approximated energy consumption for the RegionalDelivery and InterUrban duty cycles, illustrating the mean values and 95% confidence intervals from the Monte Carlo simulations, alongside the approximated function values for various robustness coefficients ψ^k

4.3. Implementation and Configuration Details

This section outlines the implementation details of the models and algorithms, along with the hardware configuration and parameter settings used in our experiments.

Implementation Details

All models and algorithms are implemented in Python (v3.12) using object-oriented programming principles. The MILPs for Model 9 and Model 16 are implemented using the Pyomo library [212], which provides a Pythonic syntax for constraint modeling and supports integration with various commercial solvers. Gurobi Optimizer [66] (v12.0.0) is selected to power the exact RMPSolver variants and the Maximization and Minimization procedures within the OSLP algorithm due to its top-tier performance and integration within the Shell technology stack. To implement the ALNS metaheuristic from Algorithm 4, we build on the framework by Wouda and Lan [189], adapting its roulette wheel selection and RRT acceptance criterion. The multi-scenario solution state and the destroy/repair operators are implemented from scratch.

Computational Setup

All experiments are conducted on an HP ZBook Fury 16 G10 Mobile Workstation, equipped with a 13th Gen Intel® Core™ i9-13950HX processor running at 2.20 GHz, featuring 24 cores and 32 logical processors, along with 64 GB of RAM. This setup is intentionally selected to demonstrate that our methodology can be used to solve reasonably large instances of the HGEV-FPP-TECU problem within a timeframe of several hours on a single, albeit high-performance, laptop. By avoiding reliance on costly and energy-intensive cloud infrastructure, our approach becomes more accessible and practical for transport companies to adopt.

ALNS and MILP Configuration

We employed the META-DET solver to verify the feasibility of potential benchmark instances during instance generation. For the toy and small instance classes, a maximum of 1000 iterations proved sufficient to find feasible instances due to their simplicity. For the medium and large instance classes, the solver was made to terminate after 100 iterations without improvement to avoid excessive computation on unpromising instances. Only instances feasible at termination were retained.

For the RRT acceptance criterion, we use the default starting threshold of $T_{start} = 0.02$ proposed by Wouda and Lan [189]. The decay factor γ is automatically calibrated so that T would reach zero at the final iteration. This is done in combination with a default stopping criterion of 1000 iterations. When using an alternative criterion based on a fixed number of iterations without improvement, where the total iteration count is unknown, we assume an upper bound of 100,000 iterations for calibration. For the roulette wheel selection mechanism, we also follow the defaults from Wouda and Lan [189], setting the score vector to $\boldsymbol{s} = (25, 5, 1, 0)$ and using a memory decay parameter of $\theta = 0.8$.

The parameters for the ALNS operators are mostly adopted from Keskin and Çatay [31], who performed parameter tuning on EVRPTW-PR instances with up to a 100 customers. The customer removal rate ρ is set to 0.2, 0.1, and 0.05 for toy/small, medium, and large instances, respectively. For zone removal, we use $k = 2$, 5, 10, and 25 clusters for toy, small, medium, and large instances. For the penalty coefficients in the generalized cost function, we employ the parameters $\pi_{cv} = 4$, $\pi_{ov} = 6$, $\pi_{tv} = 10$, $\pi_{ev} = 3$. These penalize time violations and charging overlap violations more harshly to guide the search into the direction of feasible instances, reflecting early observations that the algorithm struggled the most to satisfy timing related constraints, while capacity violations were less common and energy feasibility was maintained by design. For some experiments, we use higher values for π_{ov} to prevent situations where infeasible solutions with charging overlap have lower generalized cost than feasible solutions, as indicated in Chapter 5. Given the large number of parameters and time limitations, we perform little parameter tuning. Table 4.6 provides an overview of all chosen parameter values for ALNS.

Parameter	**Description**	**Value**
ρ	Customer removal rate	$(0.2, 0.2, 0.2, 0.1, 0.05)$
k	Number of zones	$(2, 2, 5, 10, 25)$
χ^{worst}	Worst-case determinism factor	4
χ^{shaw}	Shaw determinism factor	12
ϕ_1	Shaw distance coefficient	0.5
ϕ_2	Shaw time coefficient	13
ϕ_3	Shaw demand coefficient	0.15
ϕ_4	Shaw route coefficient	2
ρ_r^{min}	Route removal lower bound	0.1
ρ_r^{max}	Route removal upper bound	0.3
ρ_f	Charger removal rate	0.5
π_{cv}	Capacity violation coefficient	4
π_{ov}	Charging overlap violation coefficient	6
π_{tv}	Time window violation coefficient	10
π_{ev}	Energy violation coefficient	3
T_{start}	Candidate acceptance start gap	0.02
γ	Candidate acceptance decay rate	AUTO
s_1	Operator weighting score new global best	25
s_2	Operator weighting score improvement	5
s_3	Operator weighting score acceptance	1
s_4	Operator weighting score rejection	0
θ	Operator weighting memory decay rate	0.8

Table 4.6: ALNS parameters and their values

To speed up instance generation and repeated experiments, we use multiprocessing with up to 15 parallel processes. The CCG procedure described in Algorithm 1 is executed with a maximum of $\iota = 50$ iterations and a minimum improvement threshold of $\epsilon = 0.01$, disregarding objective improvements smaller than 1 cent. The procedure is modified to also terminate when, after pessimization fails to find a violating scenario, re-solving the RMP with META-ARO yields a worse objective. This prevents redundant iterations where the condition $|c^*_{\text{total}} - c_{\text{total}}| > \epsilon$ persists. Such behavior can occur because META-ARO does not guarantee solution consistency when no improvement is possible.

In the MILP formulation, each vehicle is limited to at most two charger visits ($n_{\text{visits}} = 2$) to prevent a combinatorial explosion in the number of dummy vertices. After solving the RMP with META-ARO, solutions are mapped to a dummy-augmented route set $\mathcal{R}'$ using the symmetry-breaking constraints from Eq. (3.18), ensuring compatibility with the pessimization MILPs. To reduce the number of irrelevant pessimization variables and ensure that the pessimization models fit into memory, index sets are intersected with $\mathcal{R}'$ for META-ARO. Identified scenarios are altered by removing dummy nodes, enabling their reintegration into the RMP. Lastly, we employ warm-restarts to accelerate the RMP and OSLP procedures.

5

Experiments and Results

This chapter focuses on the evaluation of the Cert$\mathcal{E}$Route framework, addressing the research sub-question:

Research Sub-question 4

How do the proposed solution methods perform when applied to realistic HGEV fleet planning scenarios, in terms of robustness, cost-efficiency, scalability, and adaptability?

In Section 5.1, we evaluate our exact deterministic solver on toy HGEV-FPP-TECU instances by comparing nominal and worst-case scenarios. This allows us to analyze the total impact of uncertainty on the decision-making and resulting cost-efficiency. In Section 5.2, we conduct a sensitivity analysis to identify which (uncertain) factors most significantly influence total cost, both within an individual instance and across multiple instances.

Section 5.3 focuses on assessing the robust optimization approach including the OSLP pessimization procedure. We compare our static and adaptive robust optimization solvers to evaluate the robustness of their solutions under budgeted uncertainty, and test the influence of the added adaptability on cost-efficiency. In Section 5.4, we test our metaheuristic deterministic solver by comparing its performance on toy and small instances against that of the exact deterministic solver in terms of runtime and cost-efficiency. We then evaluate its scalability on medium and large instances that are computationally intractable for exact solvers. Lastly, we assess the robustness of its adaptive robust counterpart under budgeted uncertainty.

5.1. Evaluation of the Deterministic Optimization Approach

In our initial experiment, we assess the performance of the Exact Deterministic Solver (EXA-DET) under two deterministic scenarios: the nominal scenario $\boldsymbol{z}^{nom}$ and the worst-case scenario $\boldsymbol{z}^{worst}$, both derived from the box uncertainty set $\mathcal{U}_{box}$. In Subsection 5.1.1, we examine the impact of uncertainty on the solver's runtime and the objective values of the obtained solutions. Subsequently, in Subsection 5.1.2, we analyze the solver's routing, timing, and charging decisions, uncovering the black box behind it through graphical representations of the solutions.

5.1.1. Analyzing Runtime and Objective Values

We first considered the five instances from the class Toy Single Vehicle (T-SV), with three customers and a single vehicle, which we solved exactly and to proven optimality. Table 5.1 shows the resulting runtimes and objective values, comparing the nominal and worst-case scenario for every instance.

Instance	Time Proven Opt. (s)		Opt. Obj. (€)		
	z^{nom}	z^{worst}	z^{nom}	z^{worst}	$\Delta\%$
T-SV-1	0.508 ± 0.045	0.580 ± 0.016	993.35	1051.15	5.82
T-SV-2	0.338 ± 0.019	0.294 ± 0.029	806.45	905.74	**12.3**
T-SV-3	0.220 ± 0.022	0.268 ± 0.013	943.14	1000.44	6.08
T-SV-4	1.048 ± 0.066	1.030 ± 0.044	566.30	609.71	7.67
T-SV-5	0.516 ± 0.049	0.540 ± 0.047	447.26	483.37	8.07
OVERALL	0.526 ± 0.292	0.542 ± 0.282	751.30	810.08	7.82

Table 5.1: Comparison of runtimes to proven optimality and optimal objective values between the nominal scenario $\boldsymbol{z}^{nom}$ and worst-case scenario $\boldsymbol{z}^{worst}$ across T-SV instances solved by EXA-DET over five runs

Every instance could be solved to proven optimality within milliseconds, with no significant difference in the runtimes between both scenarios. In terms of the objective value, we see that even with these toy instances, the difference in operational cost between the nominal and the worst-case scenario can be as high as **12.31**% for case T-SV-2, highlighting the relevance to account for uncertainties in the planning.

Dual Vehicle Instances

When we examined the slightly larger instances from the TOY DUAL VEHICLE (T-DV) class, involving two vehicles and five customers, Gurobi was able to find feasible solutions within a matter of seconds. For the nominal scenario, significant improvements in objective value were only made in the first ten seconds, after which the objective values stabilized. For the worst-case scenario, notable improvements were found within the first two minutes (see Fig. D.1). However, Gurobi was unable to prove optimality for any of the instances within an hour. The improvements in the MIP gap, which measures the difference in objective value between the lower bound obtained from the LP relaxation and the best known feasible solution, were minimal. The MIP gap decreased only slightly over the course of an hour, remaining above **90**% for all instances (see Fig. D.2). As the objective values ceased to improve by more than a euro cent after two minutes of computation, we decided to report the best objective value achieved at the two-minute mark, along with the timestep at which this solution was initially found, in addition to the objective value of the first identified solution. The corresponding results are presented in Table 5.2.

Instance	First Sol. Obj. (€)			Time Best Sol. (s)		Best Obj. (€)		
	z^{nom}	z^{worst}	$\Delta\%$	z^{nom}	z^{worst}	z^{nom}	z^{worst}	$\Delta\%$
T-DV-1	1390.10	1384.90	−0.37	0.800 ± 0.447	1.000 ± 0.000	821.88	899.62	**9.46**
T-DV-2	973.62	2882.58	196.07	1.000 ± 0.000	8.200 ± 0.447	933.67	1008.14	7.98
T-DV-3	1405.86	1638.65	16.56	4.400 ± 0.548	36.400 ± 1.342	1323.04	1404.87	6.18
T-DV-4	2202.20	3003.47	36.38	4.400 ± 0.548	23.200 ± 0.837	1506.71	1585.93	5.26
T-DV-5	1432.11	1697.53	18.53	1.000 ± 0.000	5.600 ± 12.522	797.41	865.61	8.55
OVERALL	1480.78	2121.43	43.26	2.120 ± 1.965	14.680 ± 14.519	1076.54	1152.83	7.09

Table 5.2: Comparison of runtimes for identifying first and best solutions and corresponding objective values between the nominal scenario $\boldsymbol{z}^{nom}$ and worst-case scenario $\boldsymbol{z}^{worst}$ across T-DV instances solved by EXA-DET over five runs with a two-minute timeout

In the nominal scenario, objective values all stabilized within the first five seconds, with two seconds being the average time required to find the best solution. In the worst-case scenario, an average of nearly 15 seconds was required to reach the best solution. Interestingly, the final worst-case solution for T-DV-1 was found within milliseconds, while for T-DV-3 it took over 30 seconds to find a stable solution. Overall, the gap in optimal objective values between the worst-case and nominal scenarios was slightly bigger for the T-SV class (7.8%) compared to the T-DV class (7.1%), though the latter still exhibited substantial differences, reaching up to **9.5**%.

Breakdown of the Objective Values

Fig. 5.1 shows the breakdown of the objective values into the distance, time, and energy-based costs associated with the solutions for the T-DV instances. The time costs have by far the largest overall share (**72.8**%), followed by energy costs (**22.6**%) and the distance cost (**4.5**%). These results align with the magnitudes of the chosen cost parameter values from Subsection 4.1.4.

Moving from the nominal to the worst-case scenario, the overall share of the time cost increased by **2.4** percentage points (pp), with the share of the distance cost decreasing by **0.3** pp and the share of the energy cost decreasing by **2.1** pp. The results for the T-SV instances showed the same trend. This suggests that making effective timing decisions becomes increasingly critical under more adverse conditions, which aligns well with our expectations.

The findings for T-DV-4 are especially noteworthy. In the worst-case scenario, both the total energy costs and their share of the overall operational costs decreased compared to the nominal scenario. The higher energy demand in the worst-case justified a shift from using an on-the-go charger to a cheaper depot charger, an option that was not advantageous under the nominal scenario, where the additional time costs associated with the detour outweighed the energy cost savings.

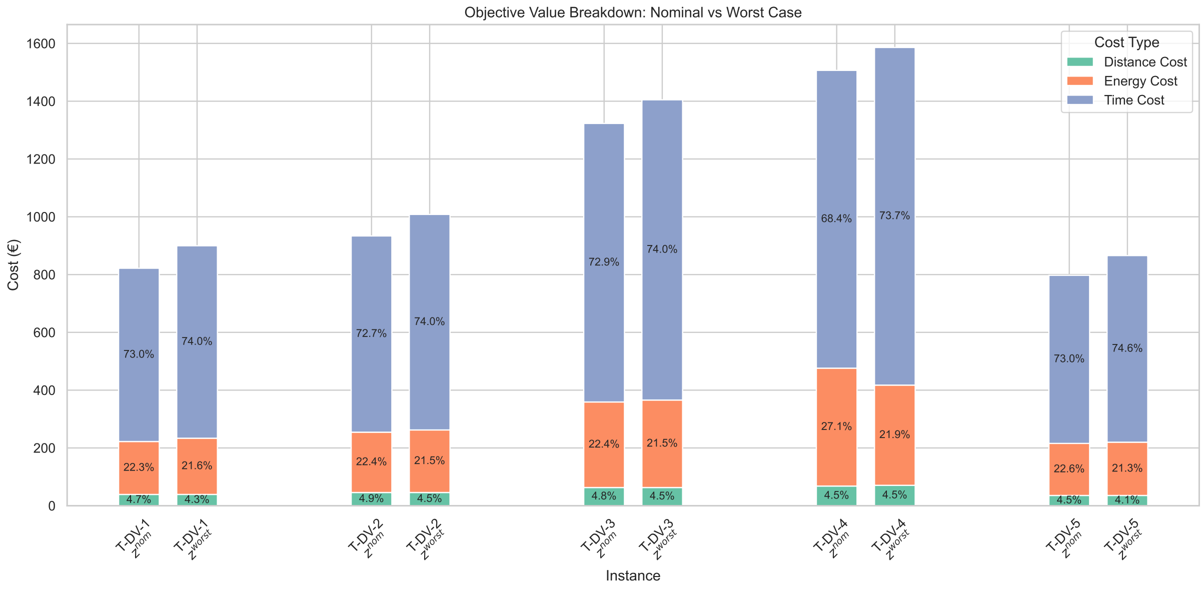


Figure 5.1: Breakdown of the objective values into distance, energy and time-related costs for the solutions to the T-DV instances for the nominal scenario $\boldsymbol{z}^{nom}$ and worst-case scenario $\boldsymbol{z}^{worst}$

Discussion

This results for the T-DV class show the main limitation of solving the HGEV-FPP-TECU exactly: even a small increase in instance size, such as adding a vehicle or a customer, leads to an exponential growth in the solution space. When more vehicles are available for a small number of customers, many potentially nearly optimal solutions arise, making it difficult for the solver to close the MIP gap. For example, one solution might use a single vehicle and incur higher timing costs, while another might deploy two vehicles, reducing travel time per vehicle but doubling driver costs, resulting in similar overall objective values.

Our MILP formulation of the HGEV-FPP-TECU from Model 11 might be further strengthened through the use of symmetry-breaking constraints, the incorporation of redundant variables with associated channeling constraints, and other advanced modeling techniques designed to reduce the search space and facilitate a tighter MIP gap closure. However, these enhancements lie beyond the scope of this research, where we work towards developing a metaheuristic approach capable of solving even larger, more realistic instances for which exact MILP formulations are generally intractable.

5.1.2. Interpreting Routing, Timing, and Charging Decisions

We have just established that the operational costs can vary significantly between the nominal and worst-case scenario, with timing-related costs holding a major share in the objective value. However, the solver's internal process for arriving at a solution that minimizes the objective for a given scenario remains abstract. To uncover this black box, we examine the solver's decision-making in both scenarios, using the instance T-SV-3 as a case study. It demonstrates how timing and charging decisions can be adjusted in an effort to make a nominal route feasible under worst-case conditions.

The EXA-DET solver produced provably optimal solutions for instance T-SV-3, which is shown geographically in Fig. 5.2. The routing and load level decisions for T-SV-3, as shown in Fig. 5.3, are identical for the nominal and worst-case scenarios. However, due to the differences in service and traversal duration as well as energy consumption, the timing and charging decisions differ between both scenarios.

In both scenarios, the vehicle begins by loading at the depot, requiring 27.5 minutes in the nominal case and 40 minutes in the worst-case. As shown in Fig. 5.4, the vehicle directly starts charging at the depot charger in the nominal scenario, after which it waits at the charger until departure at 09 : 16. In the worst-case scenario, the charging is started only at 07 : 08 so that the vehicle can depart immediately after finishing a full charging session. This difference does not influence the time costs, as they are counted only from the minute the vehicle departs from the depot charger. For both scenarios, the departure from the initial charger is scheduled as late as possible (constrained by the latest possible service start time at Customer 1) so that the driver spends no idle time waiting for service to start at any of the locations in the remaining route, thereby minimizing time costs.

During the initial charging session at the depot, the battery is fully charged, while respecting the usable range of 10% to 90%, as shown in Fig. 5.5. This strategy is cost-effective as the depot's electricity rate (0.25 €/kWh) is significantly lower than that of the on-the-go chargers (0.60 €/kWh), and no detours are required for the initial charging session.

The vehicle then visits the customers in order of their earliest service deadlines, beginning with Customer 3, whose deadline is at 11 : 17. After completing service there, the vehicle returns to the depot to take advantage of low-cost charging once again. This is feasible because service at Customer 1 does not begin until 14 : 05. The second charging session is carefully timed so that the vehicle can fully recharge its battery to the maximum state of charge and arrive just in time for the latest allowable service start at Customer 1. Consequently, the vehicle arrives at depot Charger 1 (F1') earlier in the worst-case scenario (13 : 41) than in the nominal case (13 : 47), allowing more time for recharging due to its lower SoC upon arrival.

On the way from Customer 1 to Customer 2, the vehicle makes a brief stop at Charger 3 to recharge just enough energy to return to the depot with the minimum allowed SoC of 54 kWh after completing the final delivery. Since only a small amount of additional energy is needed and Charger 3 is more conveniently located along the route than Charger 1, the time cost savings by preventing a detour via the depot outweigh the potential savings in energy costs.

After completing the visit to Customer 2, the vehicle returns to the depot as quickly as possible, arriving well before the end-of-day deadline, at 23 : 00 in the nominal scenario and 23 : 23 in the worst-case scenario. Notably, the combination of an earlier departure and a later arrival at the depot in the worst-case scenario leads to increased time and overall costs.

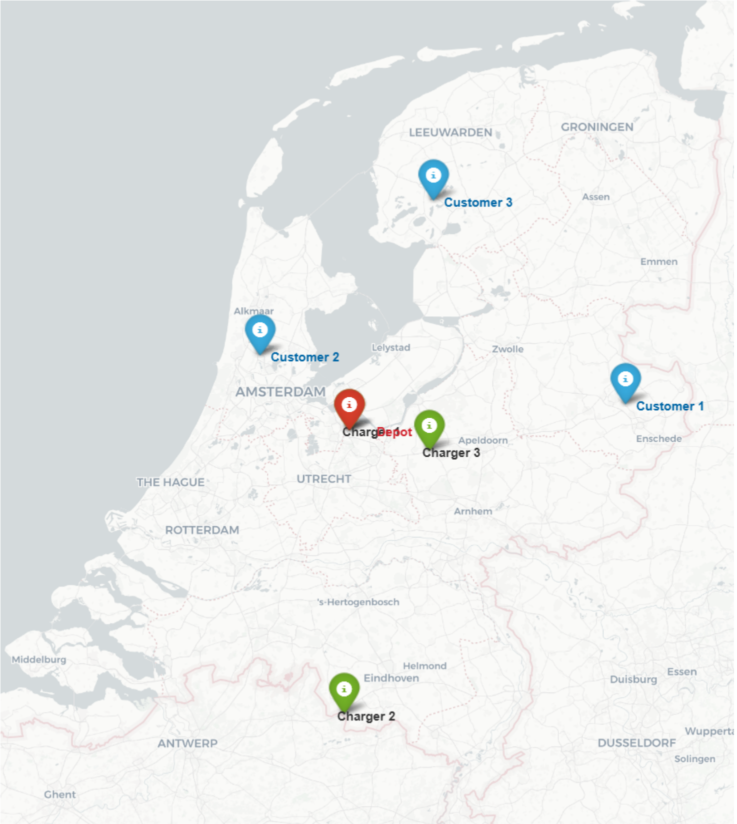


Figure 5.2: Geographical depiction of instance T-SV-3 with the locations of the depot, chargers, and customers on the map of The Netherlands

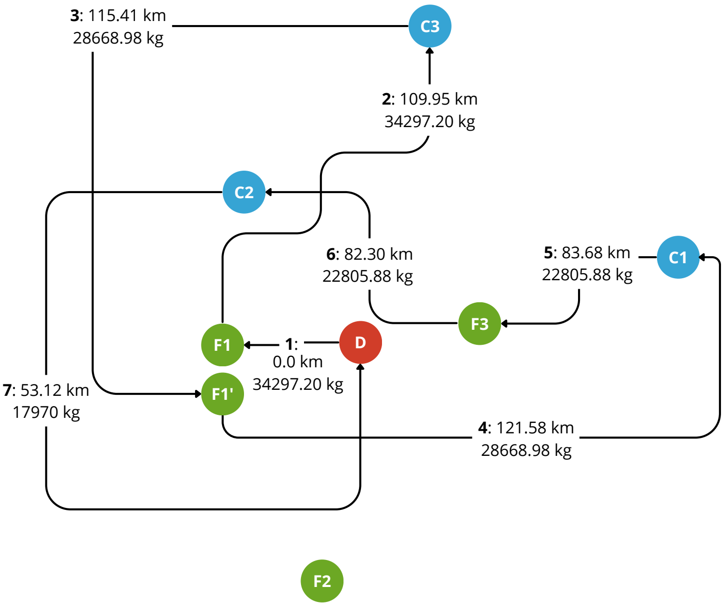


Figure 5.3: Graphical depiction of the optimal routing decisions for instance T-SV-3 for the nominal and worst-case scenario

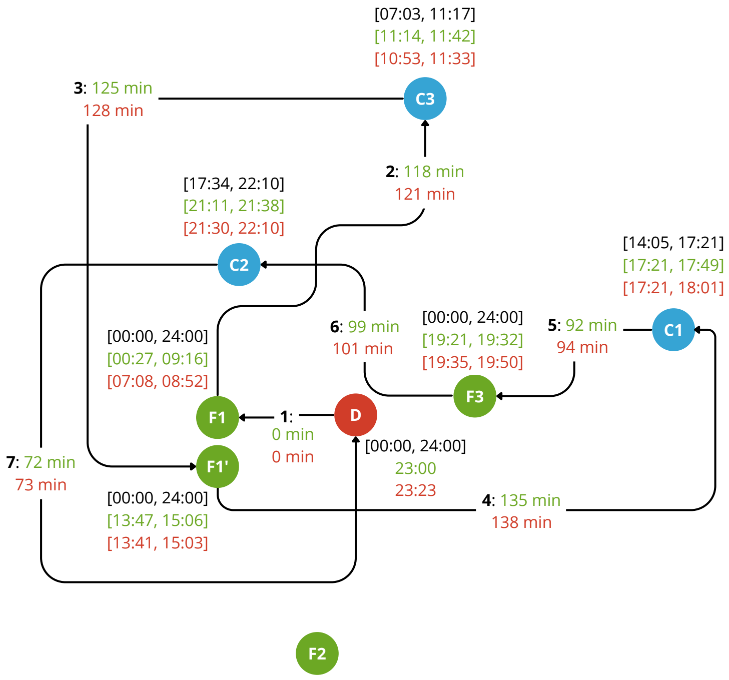


Figure 5.4: Graphical depiction of the optimal timing decisions for instance T-SV-3 for the nominal and worst-case scenario

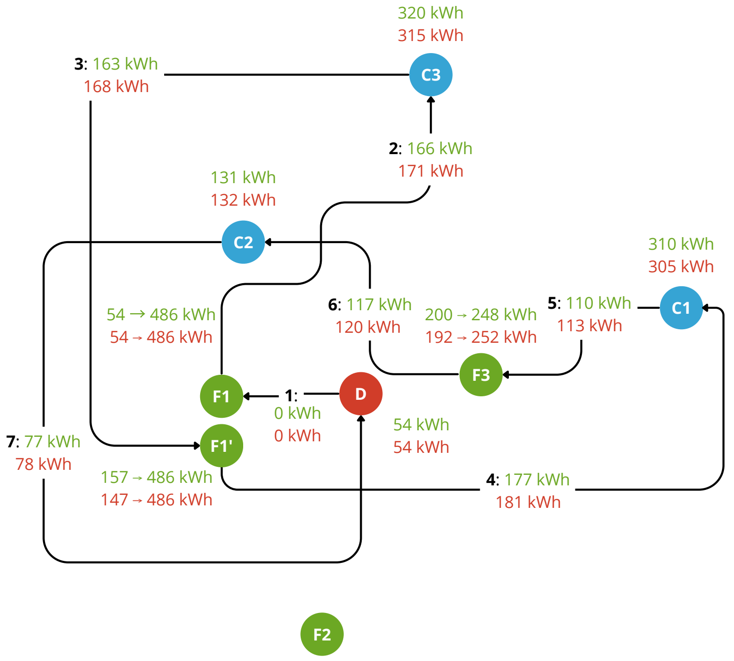


Figure 5.5: Graphical depiction of the optimal charging decisions for instance T-SV-3 for the nominal and worst-case scenario

5.2. Sensitivity Analysis

After evaluating our deterministic solver and developing intuition about how uncertainty affects its decisions, we now systematically quantify the influence of each uncertain parameter on the objective function. To achieve this, we employ a variance-based sensitivity analysis approach, as described in Subsection 5.2.1.

First, we conduct a local sensitivity analysis (Subsection 5.2.2), where we independently vary each uncertain parameter within its predefined bounds and observe the resulting changes in the objective value for a single instance. Next, we perform a global sensitivity analysis across the full set of T-SV instances (Subsection 5.2.3), examining how global variations in the realizations of the uncertainty sets influence the objective values across multiple instances, to provide a broader view of sensitivity. In Subsection 5.2.4, we also include an assessment of the impact of vehicle and charging infrastructure parameters.

5.2.1. Details on the Sensitivity Analysis

To assess the sensitivity of the optimization model outputs with respect to various input parameters, we employ a variance-based sensitivity analysis using Sobol indices. This analysis was introduced by Sobol [213] and is frequently used for sensitivity analysis [214]. We consider the first-order Sensitivity Index (SI) (Definition 21) which focuses on individual contributions of the parameters, and the Total Sensitivity Index (TSI) (Definition 22), which focuses on interactions with other parameters.

Definition 21 – First-Order Sobol Index

The first-order Sobol index SI_i quantifies the proportion of the total output variance that can be attributed to variations in the input parameter X_i, independent of interactions with other parameters:

$$SI_i := \frac{\mathbb{V}\left(\mathbb{E}[Y \mid X_i]\right)}{\mathbb{V}(Y)},$$

where $\mathbb{E}[Y \mid X_i]$ is the expected value of the output conditioned on X_i, and $\mathbb{V}(Y)$ is its total variance.

Definition 22 – Total Sobol Index

The total Sobol index TSI_i measures the overall contribution of X_i to the output variance, including both its first-order effect and all higher-order interaction effects with other parameters:

$$TSI_i := \frac{\mathbb{E}\left[\mathbb{V}(Y \mid X_{\sim i})\right]}{\mathbb{V}(Y)},$$

where $X_{\sim i}$ denotes the set of all input parameters except X_i and $\mathbb{V}(Y \mid X_{\sim i})$ represents the variance of the output when all inputs but X_i are fixed. The expectation is taken over all possible values of $X_{\sim i}$.

We again consider the EXA-DET solver, now modeled as a black-box function $f : \mathbb{R}^l \to \mathbb{R}$ that maps a vector of l independent, uniformly distributed input parameters $\boldsymbol{X} = \{X_1, \ldots, X_l\}$ to the total operational cost $Y = f(\boldsymbol{X}) = c^*_{\text{total}}$. The uniform distribution reflects a neutral assumption due to the lack of prior knowledge about the distribution of the inputs for the HGEV-FPP-TECU instance.

Computing the Sobol indices allows us to decompose the total variance in cost into individual and interactive contributions, assuming that the input parameters are independent. Since the function $f(\boldsymbol{X})$ is deemed a black box, for which the variances and expected values cannot be estimated analytically, Monte Carlo simulations are used to approximate these terms. The inputs are sampled in a quasi-random manner using Saltelli's extension of the Sobol sequence, as implemented by the SALib library [215]. For more details on the sampling procedure, we refer to Campolongo, Saltelli, and Cariboni [216].

Discussion

A limitation of using a uniform sampling distribution is that it assigns equal probability to all combinations of parameter values, including those that are highly improbable in practice, which may distort the sensitivity analysis by overrepresenting unrealistic scenarios. However, such scenarios are also inherently considered within the robust optimization framework under the box uncertainty set. Therefore, for the purposes of this study, it suffices to establish a general methodological framework for sensitivity analysis, with the flexibility to incorporate more accurate distributions as additional data becomes available.

5.2.2. Local Sensitivity Analysis

Local sensitivity analysis identifies customers and road connections within the network that significantly impact operational costs for a specific instance. This information helps to further uncover the black box of the optimization model and can be used by fleet planners to assess risks associated with a model-generated solution, enabling informed manual adjustments. For example, planners might optimize for the nominal scenario to avoid overly conservative planning and ensure that links with a substantial negative effect on costs, when their traversal durations deviate from their nominal value, are avoided.

To provide an example, we again considered instance T-SV-3 and calculated the first-order and total Sobol indices for the uncertain service durations $\{\nu_i^s : i \in \mathcal{C} \cup \{0\}\}$ and traversal duration $\{t_{ij}^s : i \in \mathcal{V}'_0, j \in \mathcal{V}'_{N+1}\}$ within the associated box uncertainty set $\mathcal{U}_{box}$. This resulted in $L = 99$ random variables X_i, with ranges $\mathcal{J}(X_i)$ set according to $\mathcal{U}_{box}$, for which $M(2L+2) =$ 25,600 MC simulations were performed using sampling parameter $M = 128$. The resulting indices with non-zero values can be found in Table 5.3:

Parameter	Uncertainty Range	SI_i	TSI_i
ν_i : Customer 1	$[15, 40]$	0.292	**0.275**
ν_i : Customer 2	$[15, 40]$	**0.303**	0.272
ν_i : Customer 3	$[15, 40]$	0.266	**0.275**
t_{ij} : Charger 1 → Customer 3	$[115.16, 121.08]$	0.052	0.045
t_{ij} : Charger 1 → Customer 1	$[131.63, 138.18]$	0.075	0.074
t_{ij} : Charger 3 → Customer 2	$[96.28, 100.71]$	0.013	0.018
t_{ij} : Customer 1 → Charger 3	$[89.82, 94.32]$	0.022	0.017
t_{ij} : Customer 3 → Charger 1	$[122.04, 128.26]$	0.053	0.055
t_{ij} : Customer 2 → Depot	$[70.50, 73.36]$	0.006	0.010

Table 5.3: First-order and total Sobol indices for uncertain parameters in the T-SV-3 instance

Discussion

We find that the cost in this instance is insensitive to the service duration at the depot. The vehicle can be loaded early and remain idle at the depot or charger, thereby minimizing the time-related cost, which only start accumulating once it leaves the charger. The service durations at the customers give rise to the highest Sobol indices, together accounting for **86%** of the variance of the total operational cost. Keep in mind that the uncertainty intervals for service durations are significantly broader than those for traversal durations, allowing them to fluctuate more and thereby impact cost more severely.

Notably Customer 2 has a higher SI compared to Customers 1 and 3, whereas its TSI is slightly lower. This may be attributed to Customer 2's later time window, which results in it being placed at the end of the route, thereby directly influencing the route's end time so that it has a slightly greater impact on time-related costs. Fig. 5.4 showed that Customers 1 and 3 have time windows that are closer together than those of Customers 2 and 3. Since the mutual traversal durations and distances are similar, this leads to stronger coupling between Customers 1 and 3. While the coupling between Customers 1 and 2 due to time windows is comparable, the longer distance and traversal duration between them reduce their overall interaction, resulting in slightly lower TSI for Customer 2 compared to Customers 1 and 3.

The remaining **14%** of the variance is attributed to the traversal durations. All relevant links are part of the routing solution from Fig. 5.3, which remained constant when moving from the nominal to the worst-case realization of the uncertainty set. The critical link appears to be the link from Charger 1 to Customer 1. Recall from our case study and Fig. 5.4 that the vehicle tries to arrive as late as possible at Customer 1, while charging as much as possible at Charger 1, which is a depot charger with a lower tariff. The traversal duration between Charger 1 and Customer 1 is therefore pivotal in determining the departure time from the charger and therefore the amount of energy that can be recharged, explaining the high SI and TSI value. Similarly, the traversal duration between Customer 3 and Charger 1 shows a high sensitivity score, as it determines when the vehicle can start charging at the depot and therefore also influences the amount of energy that can be recharged at low costs. Based on that, we can conclude that the operational cost of the solution is highly influenced by the intermediate charging session at the depot. More surprisingly, the link from Customer 2 to the Depot, which directly influences the route end-time, has low SI and TSI scores. This may be attributed to the link having the narrowest uncertainty interval, resulting in small increments or decrements to the costs.

5.2.3. Global Sensitivity Analysis

Global sensitivity analysis evaluates the impact of global variations across different types of parameters over multiple instances, rather than focusing on individual parameters. For example, instead of examining the influence of traversal duration for a specific link, we consider the overall effect of traversal durations. We expand this approach beyond the uncertainty set to analyze the impact of energy-related parameters associated with fleet and charging infrastructure investments, such as the charging rate, battery capacity, and the energy consumption function. This analysis can offer valuable insights into the most critical factors in routing, regardless of specific instances, and assist fleet operators in assessing the suitability of various vehicle types and charging infrastructure configurations.

To provide an example, we considered the full set of T-SV instances and set the worst-case scenario $\boldsymbol{z}^{worst} \in \mathcal{U}_{box}$ for every instance as the baseline for its parameter values. We then sampled multipliers $\{X_1, \ldots, X_4\}$ in range $[0.5, 1.0]$ for the service duration, traversal duration, inverse charging rate and robustness multiplier from Eq. (4.1) respectively, whilst sampling a multiplier $X_5 \in [1.0, 1.5]$ for the battery capacity. We again employed sampling parameter $M = 128$, resulting in 1,536 MC simulations during which the sampled multipliers were uniformly applied across all instances and Y was set to the mean optimal objective values c^*_{total} over all instances.

The chosen sampling intervals allow each parameter to fluctuate by 50% and ensure that the instances remain feasible, giving rise to more optimistic scenarios than the worst-case scenario. Conversely, making the scenario more pessimistic in a similar manner would render some instances infeasible for certain parameter combinations that were not anticipated for during instance generation. Note that a scenario with a traversal duration multiplier of 0.5 would correspond to the vehicles driving at twice their nominal velocity. While this is an unrealistic scenario, it helps us to provide an insight into the theoretical global influence of varying the traversal duration. The resulting Sobol indices are listed in Table 5.4:

Parameter	**Sampling Range**	SI_i	TSI_i
X_1 : Service duration multiplier	$[0.5, 1.0]$	0.026	0.027
X_2 : Traversal duration multiplier	$[0.5, 1.0]$	**0.891**	**0.927**
X_3 : Inverse charging rate multiplier	$[0.5, 1.0]$	0.000	0.000
X_4 : Energy consumption multiplier	$[1.0, 1.5]$	0.084	0.074
X_5 : Battery capacity multiplier	$[0.5, 1.0]$	0.004	0.003

Table 5.4: First-order and total Sobol indices for parameter types across the T-SV instances

Discussion

The total operational cost seems to be most sensitive to the traversal duration, which accounts for **89**% of the variance in the objective value. The TSI value for traversal duration is even higher, indicating a stronger combined effect with other factors. Energy consumption, which has the second largest SI and TSI values, accounts for approximately **8.4**% of the variation in the objective value. The high TSI value was expected since the total energy consumption over a link is a function of the traversal duration, linking the two in their impact on energy and timing costs.

The pronounced impact of traversal duration is consistent with our expectations, given that HGEVs in the studied distribution transport setting spend the majority of their time in transit rather than unloading or recharging. However, this pattern may differ in other HGEV operational contexts. While service durations exhibited high SI and TSI indices in the local sensitivity analysis, largely due to their broader uncertainty ranges compared to traversal times, they contributed only **2.6**% to the total variation in operational costs at the global level. Nonetheless, we expect service durations to play a more prominent role than traversal duration in transport settings characterized by shorter travel distances and more frequent stops.

The inverse charging rate appears to have no impact on costs, suggesting sufficient flexibility within the T-SV instances to fit charging sessions between customer visits, utilizing idle time. This observation is consistent with our case study from Subsection 5.1.2 and local sensitivity analysis, where vehicles return to the depot to recharge while waiting for the next customer to begin service. Similarly, battery capacity seems to have a minimal effect on the operational costs, as there is enough time to recharge with minimal detours, limiting the impact of a larger battery. It is important to note that the time windows in our instances are relatively wide, ranging from 3 to 6 hours. In practice, these windows may be significantly narrower, although we currently lack data to confirm this. Under tighter time constraints, we expect the influence of charging rate and battery capacity to become more pronounced.

5.2.4. Further Evaluation of Battery Capacity and Charging Rate

The results from the global sensitivity analysis appeared somewhat counterintuitive in the sense that the total operational cost was shown to be relatively insensitive to both the charging rate and battery capacity compared to other factors, while we expected these factors to have a more profound influence. To demonstrate that battery capacity and charging rate can still significantly impact the operational costs of a given instance, we created two modifications of the instances in the T-SV class and compared the operational costs of these modified instances with those of the original instances under both nominal and worst-case realizations of the uncertainty set. These instances also mimic a future scenario where both the charging rate and battery capacity increase for HGEVs.

In the first modification class, denoted by T-SV-B, the battery capacity of the baseline vehicle from Subsection 4.1.2 was increased from 540 to 780 kWh, aligning with the upcoming second generation of the Volvo FH Electric, whose production is expected to start in early 2026 [217]. In the second modification class, denoted by T-SV-C, the maximum charging rate of the vehicle was increased from 250 kW to 400 kW to match the maximum rate of modern DC fast chargers at Shell sites. The resulting objective values are listed in Table 5.5:

Instance	**Optimal Objective (€)**			
	z^{nom}	Δ%	z^{worst}	Δ%
T-SV-1-B	901.36	-9.26	954.78	-11.55
T-SV-2-B	700.85	-13.09	756.03	-16.67
T-SV-3-B	755.21	**-19.93**	804.85	**-19.58**
T-SV-4-B	535.63	-5.41	576.69	-6.23
T-SV-5-B	447.26	0.00	483.37	0.00
Overall T-SV-B	668.06	**−9.54**	715.14	**−10.81**
T-SV-1-C	962.90	**-3.07**	1019.18	**-3.04**
T-SV-2-C	789.89	-2.05	885.56	-2.22
T-SV-3-C	914.77	-3.01	970.50	-2.99
T-SV-4-C	561.90	-0.78	604.52	-0.85
T-SV-5-C	447.26	0.00	483.37	0.00
Overall T-SV-C	735.344	**−1.78**	799.07	**−1.82**

Table 5.5: Comparison of objective values across the modified instances of class T-SV-B and T-SV-C for the nominal scenario z^{nom} and worst-case scenario z^{worst}. Δ% denotes the percentage difference in objective value between the modified instance and the original instance from the T-SV class.

Discussion

Increasing battery capacity seems to significantly impact the total cost, with reductions of up to **20%** for instance T-SV-3, and an overall average decrease of around **10-11%** across all T-SV instances in both nominal and worst-case scenarios. A larger battery allows the vehicle to initially recharge more energy at the depot, taking advantage of cheaper rates and avoiding costly on-the-go recharges that require detours.

Although increasing the charging rate to 400 kW appears less impactful than boosting the battery to 780 kWh, it still achieves cost reductions of up to **3%**, averaging around **1.8%** for both scenarios. In the logistics industry, where profit margins are often just a few percentage points, due to high operating costs and intense competition, even a modest improvement in cost-efficiency can have a meaningful impact on profitability. That is, a nearly 2% reduction in costs through optimization is quite significant.

For both instance classes, reductions are slightly higher in the worst-case scenario compared to the nominal one. This is expected, as the problem becomes more constrained, making a larger battery or faster charging more influential on operational costs. Notably, for T-SV-5, neither modification affects the vehicle's performance, as it can complete the route using less than 540 kWh of energy and has sufficient time for its initial charging session.

5.3. Evaluation of the Robust Optimization Approach

So far, we have focused on the Exact Deterministic Solver (EXA-DET), which optimizes for a single fixed scenario. We now evaluate the Exact Robust Solver (EXA-RO) and Exact Adaptive Robust Solver (EXA-ARO), which account for uncertainty using our One-Step Look-ahead Pessimization (OSLP) algorithm to extend the scenario set starting from the nominal scenario.

In Subsection 5.3.1, we compare scenario-specific solutions from the robust EXA-RO and EXA-ARO solvers to those from the deterministic EXA-DET on T-SV and T-DV instances under the box uncertainty set, evaluating cost-efficiency and adaptability. In Subsection 5.3.2, we assess the robustness of the solutions to the T-SV instances obtained using the EXA-RO and EXA-ARO solvers under the locally budgeted uncertainty set, for which premature convergence might occur, to test the robustness of the resulting solutions.

5.3.1. Assessing Adaptability and Cost-Efficiency

Recall that EXA-RO aims to find a single solution $(\boldsymbol{x}, \boldsymbol{y})$ that is feasible and cost-efficient across the entire uncertainty set. This often leads to conservative solutions biased towards worst-case scenarios. In contrast, EXA-ARO fixes the routing decisions $\boldsymbol{x}$ but allows timing and charging decisions $\boldsymbol{y}^s$ to adapt per scenario $s \in \mathcal{S}(\mathcal{U})$, enabling more flexible and potentially less conservative outcomes. To quantify this adaptability, we compare the best objective values under $\boldsymbol{z}^{nom}$ and $\boldsymbol{z}^{worst}$, given the decisions made by EXA-DET and EXA-ARO under the uncertainty set $\mathcal{U}_{box}$, with those obtained through direct optimization for these scenarios using EXA-DET. The results are presented in Table 5.6.

The EXA-RO and EXA-ARO solvers both required two RMP iterations per instance to reach convergence. In the first iteration, they identified the worst-case scenario $\boldsymbol{z}^{worst}$ and incorporated it into the RMP. The final solution was found in the subsequent RMP iteration so that both solvers achieved the same worst-case objective value as EXA-DET. Since the objective function minimized the worst-case cost, EXA-ARO did not tune the second-stage decisions for the nominal case automatically. To assess nominal performance, we applied EXA-DET to find the best second-stage decisions given the routing solutions generated by EXA-ARO.

For the T-SV instances, EXA-ARO adjusted its routing decisions to anticipate the worst-case scenario only for T-SV-2, which led to a higher cost compared to EXA-DET under the nominal scenario. For all other T-SV instances, its nominal performance matched that of EXA-DET. For the T-DV instances, routing changes occurred for T-DV-1, again resulting in higher costs for EXA-ARO. However, for T-DV-4, EXA-ARO produced a better nominal solution than EXA-DET, indicating that the latter had not yet reached optimality at the two-minute mark. This suggests that anticipating worst-case scenarios can accelerate the discovery of high-quality nominal solutions. The adaptability of EXA-ARO proved beneficial, yielding cost reductions of **6.2**% and **6.8**% over EXA-RO for the T-SV and T-DV instances under the nominal scenario, respectively.

Instance	**Runtime (s)**			**Best Obj. z^{nom} (€)**			**Best Obj. z^{worst} (€)**
	EXA-RO	EXA-ARO	EXA-DET (z^{worst})	EXA-RO	EXA-ARO	EXA-DET	All
T-SV-1	1.393 ± 0.140	1.544 ± 0.083	0.580 ± 0.016	1051.15	993.35	993.35	1051.15
T-SV-2	0.812 ± 0.035	1.316 ± 0.085	0.294 ± 0.029	905.74	**849.66**	806.45	905.74
T-SV-3	0.966 ± 0.039	1.291 ± 0.047	0.268 ± 0.013	1000.44	943.14	943.14	1000.44
T-SV-4	2.512 ± 0.210	3.855 ± 0.155	1.030 ± 0.044	609.71	566.30	566.30	609.71
T-SV-5	1.626 ± 0.093	2.487 ± 0.145	0.540 ± 0.047	483.37	447.26	447.26	483.37
T-SV Overall	$\mathbf{1.462 \pm 0.613}$	$\mathbf{2.099 \pm 0.987}$	0.542 ± 0.282	**810.08**	**759.94**	751.30	810.08
T-DV-1	242.196 ± 0.149	242.401 ± 0.422	1.000 ± 0.000	899.62	**831.76**	821.88	899.62
T-DV-2	243.255 ± 0.573	242.121 ± 0.333	8.200 ± 0.447	1008.14	933.67	933.67	1008.14
T-DV-3	242.394 ± 0.135	242.441 ± 0.177	36.400 ± 1.342	1404.87	1323.04	1323.04	1404.87
T-DV-4	242.268 ± 0.183	242.044 ± 0.049	23.200 ± 0.837	1585.93	**1488.51**	1506.71	1585.93
T-DV-5	244.911 ± 1.437	242.349 ± 0.209	5.600 ± 12.522	865.61	797.41	797.41	865.61
T-DV Overall	243.005 ± 1.243	242.271 ± 0.314	14.680 ± 14.519	**1152.83**	**1074.88**	1076.54	1152.83

Table 5.6: Comparison of runtimes and scenario-specific objective values resulting from solving instances of the T-SV and T-DV classes under the box uncertainty set $\mathcal{U}_{box}$ using EXA-DET, EXA-RO and EXA-ARO over five runs with a timeout of two minutes

In terms of runtime, EXA-RO and EXA-ARO took 3 to 4 times longer than EXA-DET on the T-SV instances, due to the more elaborate solution process involving RMP solving, pessimization, and handling the overhead of translating results between the MILPs. About half of the total runtime was attributed to this overhead. For the T-DV instances, a two-minute timeout was enforced, limiting the comparability of runtimes. Nonetheless, overhead was minimal in these cases, with RMP solving dominating the runtime. Gurobi efficiently solved the linear pessimization problems in under **20** ms across all instances, leveraging its advanced LP presolve capabilities. EXA-ARO exhibited slightly longer runtimes than EXA-RO on the T-SV instances, primarily due to the second RMP iteration. While the second iteration in EXA-RO was **6**% slower (despite warm starts), it was **26**% slower for EXA-ARO, owing to the added complexity introduced by the additional second-stage decision variables.

5.3.2. Assessing Robustness under Budgeted Uncertainty

The box uncertainty set $\mathcal{U}_{box}$ used in our previous experiments leads to very pessimistic scenarios as all uncertain parameters can attain their worst case value simultaneously. In an effort to consider more realistic scenarios, where only a subset of the uncertain parameters attain their worst-case value simultaneously, we also experimented with the locally budgeted uncertainty set $\mathcal{U}_{local}$ from Definition 14.

We focused on instances from the T-SV class, all with three customers and a depot subject to the service duration budget Γ_ν, along with 100 edges $(i,j) \in \mathcal{E}$ governed by traversal duration budget Γ_t. We tested values $\Gamma_\nu \in \{0,1,2,3,4\}$ and $\Gamma_t \in \{0,25,50,75,100\}$ to construct different locally budgeted uncertainty sets $\mathcal{U}_{local}$. We recorded the objective values and the number of RMP iterations until convergence for solving the instances using the EXA-RO and EXA-ARO solvers. For EXA-ARO, we again applied the EXA-DET solver to adjust the second-stage decision variables after the uncertainty realization.

To assess the robustness of solutions after convergence, we sampled 1000 scenarios from the set of extreme points $ext(\mathcal{U}_{local})$ for each uncertainty configuration and instance. This involved randomly choosing Γ_t traversal duration parameters and Γ_ν service duration parameters, each of which was set to its maximum allowed deviation from the nominal value, while all other parameters were set to their nominal values. The EXA-ARO solver was then applied to derive second-stage decisions given the first-stage solutions obtained from EXA-RO and EXA-ARO. If a feasible solution could be found, this indicated robustness, whereas proven infeasibility suggested that OSLP had converged prematurely, without having considered this scenario. Table 5.7 summarizes the robustness percentages, average number of RMP iterations, and average reductions in worst-case cost relative to $\boldsymbol{z}^{worst}$ across all T-SV instances and variants of the locally budgeted uncertainty set.

When both Γ_ν and Γ_t were set to 0, the objective matched that of $\boldsymbol{z}^{nom}$. Conversely, $\Gamma_\nu = 4$ and $\Gamma_t = 100$ led to the same result as for $\boldsymbol{z}^{worst}$, validating our implementation of the locally budgeted uncertainty set. We note that cost reductions occur mainly in scenarios with lower traversal budgets Γ_t, while the influence of Γ_ν seems to be less pronounced.

Notably, with $\Gamma_t = 25$, both EXA-RO and EXA-ARO experienced premature convergence, with minimum robustness scores of **94.8**%. All other configurations achieved 100% robustness, leading to the high overall robustness rates of **99.4**% for EXA-RO and **99.0**% for EXA-ARO. Although the difference is small, the robustness for the more conservative EXA-RO is slightly higher. In contrast, the more adaptive EXA-ARO solver achieved a slightly greater reduction in worst-case cost compared to the EXA-RO solver, with overall reductions of approximately **3.6**% versus **3.3**%, respectively. Convergence was generally achieved within **2** to **3** RMP iterations, with EXA-ARO (2.1 iterations) converging slightly faster than EXA-RO (2.2 iterations) overall. The results indicate that the deployment of the OSLP algorithm for pessimization can ensure a high level of robustness by introducing only a small number of additional scenarios to the RMP, despite the risk of premature convergence under budgeted uncertainty.

$\boldsymbol{\Gamma_\nu}$	$\boldsymbol{\Gamma_t}$	**Avg. Scenario Rel. Obj. Improv.** (Δ%)		**Avg. Num. RMP Iterations**		**Sample Robustness (%)**	
		EXA-RO	EXA-ARO	EXA-RO	EXA-ARO	EXA-RO	EXA-ARO
0	0	-7.36 ±1.96	-7.36 ±1.96	1.00 ±0.00	1.00 ±0.00	100.00	100.00 ($\boldsymbol{z}^{nom}$)
0	25	-6.44 ±2.71	-6.62 ±2.56	2.20 ±0.75	2.20 ±0.75	**94.82**	**94.82**
0	50	-4.93 ±1.25	-4.93 ±1.25	2.20 ±0.40	2.20 ±0.40	100.00	100.00
0	75	-4.24 ±1.34	-4.24 ±1.34	2.40 ±0.49	2.40 ±0.49	100.00	100.00
0	100	-4.22 ±1.35	-4.22 ±1.35	2.00 ±0.00	2.00 ±0.00	100.00	100.00
1	0	-7.15 ±2.16	-7.15 ±2.16	1.20 ±0.40	1.20 ±0.40	100.00	100.00
1	25	-4.56 ±2.34	-5.52 ±2.87	2.60 ±0.49	2.00 ±0.63	**94.94**	**94.94**
1	50	-3.33 ±1.18	-3.50 ±0.80	2.20 ±0.40	2.40 ±0.49	100.00	100.00
1	75	-2.21 ±0.77	-2.85 ±0.93	2.40 ±0.49	2.20 ±0.40	100.00	100.00
1	100	-3.35 ±1.17	-3.78 ±1.71	2.20 ±0.40	2.00 ±0.00	100.00	100.00
2	0	-6.06 ±1.94	-6.06 ±1.94	2.00 ±0.00	2.00 ±0.00	100.00	100.00
2	25	-3.04 ±1.41	-4.75 ±2.76	2.80 ±1.17	2.00 ±0.00	**100.00**	**94.82**
2	50	-2.23 ±1.06	-2.49 ±0.86	2.60 ±0.49	2.20 ±0.40	100.00	100.00
2	75	-1.76 ±1.14	-1.82 ±0.91	2.40 ±0.49	2.20 ±0.40	100.00	100.00
2	100	-1.73 ±1.13	-2.82 ±0.91	2.40 ±0.49	2.00 ±0.00	100.00	100.00
3	0	-4.65 ±2.01	-4.65 ±2.01	2.00 ±0.00	2.00 ±0.00	100.00	100.00
3	25	-1.79 ±0.78	-3.24 ±2.81	3.00 ±1.10	2.40 ±0.49	**100.00**	**94.82**
3	50	-1.25 ±0.60	-1.25 ±0.60	2.40 ±0.49	2.40 ±0.49	100.00	100.00
3	75	-0.57 ±0.67	-0.57 ±0.67	2.40 ±0.49	2.40 ±0.49	100.00	100.00
3	100	-0.98 ±0.55	-1.41 ±0.45	2.20 ±0.40	2.00 ±0.00	100.00	100.00
4	0	-5.95 ±1.96	-5.95 ±1.96	2.00 ±0.00	2.00 ±0.00	100.00	100.00
4	25	-3.31 ±2.55	-3.93 ±3.31	2.60 ±0.49	2.20 ±0.40	**94.94**	**94.82**
4	50	-0.71 ±0.39	-0.71 ±0.39	2.20 ±0.40	2.20 ±0.40	100.00	100.00
4	75	-0.03 ±0.04	-0.03 ±0.04	2.20 ±0.40	2.20 ±0.40	100.00	100.00
4	100	-0.00 ±0.00	-0.00 ±0.00	2.00 ±0.00	2.00 ±0.00	100.00	100.00 ($\boldsymbol{z}^{worst}$)
OVERALL		$\mathbf{-3.27 \pm 2.64}$	$\mathbf{-3.59 \pm 2.76}$	$\mathbf{2.22 \pm 0.66}$	$\mathbf{2.07 \pm 0.49}$	**99.39**	**98.97**

Table 5.7: Comparison of average objective improvement, number of required RMP iterations, and robustness across T-SV instances solved by EXA-RO and EXA-ARO under locally budgeted uncertainty sets $\mathcal{U}_{local}$ with varying uncertainty budgets Γ_ν for service duration and Γ_t for traversal duration, using 1,000 random samples from $ext(\mathcal{U}_{local})$ to assess robustness. Δ% indicates the deviation from the worst-case objective value found for $\boldsymbol{z}^{worst}$.

5.4. Evaluation of the Adaptive Large Neighbourhood Search

So far, our experiments have only involved simplified toy instances from the T-SV and T-DV classes, which are significantly smaller than those encountered in real-world transport settings. However, as we will demonstrate, even deterministic exact optimization becomes infeasible when considering more realistic instances involving 10 or more customers and at least three vehicles. This is precisely where our metaheuristic solvers, Metaheuristic Deterministic Solver (META-DET) and Metaheuristic Adaptive Robust Solver (META-ARO), prove essential.

In Subsection 5.4.1, we compare META-DET with EXA-DET on T-SV, T-DV, and Small (S) instances, focusing on the cost-efficiency of the solutions and identifying the practical limitations of EXA-DET. In Subsection 5.4.2, we assess the scalability of META-DET by evaluating its performance on the Medium (M) and Large (L) instance classes. In Subsection 5.4.3 we assess the integration of all methodological components in Cert$\mathcal{E}$Route, using META-ARO to solve instances of the M class under budgeted uncertainty while assessing the robustness of the resulting solutions.

5.4.1. Comparison with the Exact Deterministic Solver

We first conducted exploratory testing with 100,000 iterations of the ALNS metaheuristic on toy and small instances to assess convergence. Based on the observed stagnation in objective values (see Fig. D.3 to Fig. D.5), we selected 100, 600, and 12,000 iterations for reporting results on the T-SV, T-DV, and S instances, respectively. Each instance was solved five times using different random seeds [42, 10, 99, 41, 69]. We reported the runtimes, means, and standard deviations of the resulting objective values, and compared the best solutions found by META-DET with those obtained via the exact EXA-DET solver, which was given a one-hour timeout. For the T-SV instances, the difference in objective value directly corresponds to the optimality gap. In contrast, for the other instance classes, this difference is referred to as a performance gap, since the exact solver could not confirm optimality within the one-hour time limit. The complete results for the worst-case scenario are presented in Table 5.8, while the full nominal results are provided in Appendix D (Table D.1).

Under the nominal scenario $\boldsymbol{z}^{nom}$, only instance T-SV-4 could be solved to optimality, with an average optimality gap of **10.9%** across the T-SV instances. Notably, T-SV-5 exhibited a substantial optimality gap of **35.7%**. All T-SV instances except T-SV-1 showed consistent results across seeds. For the T-DV instances, META-DET converged for two out of five instances (0 variance across seeds), yielding an average performance gap of 7.5%. For the Small instance class, EXA-DET was only able to solve S-2 and S-3 within the time limit, while META-DET found feasible solutions for all instances in under **100 seconds**. However, the results showed considerable variability across seeds, particularly for S-4. The performance gaps for S-2 and S-3 were **9.1%** and **10.7%**, respectively.

In the worst-case scenario $\boldsymbol{z}^{worst}$, the average optimality gap for the T-SV instances decreased to 8.15%, while both T-SV-2 and T-SV-4 could be solved to optimality. For the T-DV instances, the performance gap slightly increased to **8.2%**, but META-DET now demonstrated a significant runtime advantage. While the average runtime for EXA-DET increased from 2 seconds under the nominal scenario to over **14.5** seconds under the worst-case scenario, META-DET experienced only a marginal increase from 2 to **2.5** seconds. For the Small instance class, EXA-DET could only solve S-3, failing on S-2. In contrast, META-DET could solve all instances within **100 seconds**. On S-3, META-DET was on average 20 seconds faster than EXA-DET, though the best solution found had a performance gap of **13.1%**.

Instance	**Runtime (s)**		**Avg. Obj. (€)**		
	EXA-DET	META-DET	EXA-DET	META-DET	Δ%
T-SV-1	0.580 ± 0.016	1.178 ± 0.475	1051.15	1167.40 ± 29.12	9.18
T-SV-2	0.294 ± 0.029	0.807 ± 0.171	905.74	905.74 ± 0.00	**0.00**
T-SV-3	0.268 ± 0.013	0.450 ± 0.574	1000.44	1028.87 ± 0.00	2.84
T-SV-4	1.030 ± 0.044	0.900 ± 0.095	609.71	609.71 ± 0.00	**0.00**
T-SV-5	0.540 ± 0.047	0.614 ± 0.335	483.37	622.13 ± 0.00	**28.71**
T-SV Overall	0.542 ± 0.282	0.790 ± 0.451	810.08	866.77 ± 221.34	**8.15**
T-DV-1	1.000 ± 0.000	0.674 ± 0.189	899.62	899.64 ± 0.00	**0.00**
T-DV-2	8.200 ± 0.447	1.171 ± 0.000	1008.14	1094.07 ± 0.00	8.52
T-DV-3	36.400 ± 1.342	4.150 ± 2.411	1404.87	1469.98 ± 0.00	4.63
T-DV-4	23.200 ± 0.837	1.823 ± 0.308	1585.93	1897.05 ± 65.99	11.34
T-DV-5	5.600 ± 12.522	3.763 ± 1.567	865.61	1020.09 ± 26.89	16.29
T-DV Overall	**14.680 ± 14.519**	**2.534 ± 2.004**	1152.83	1310.85 ± 389.46	**8.16**
S-1	-	181.394 ± 82.165	-	2375.84 ± 2.17	-
S-2	-	81.173 ± 36.140	-	2517.63 ± 120.31	-
S-3	109.400 ± 3.715	93.573 ± 57.981	1519.34	1631.89 ± 4.26	**13.09**
S-4	-	8.933 ± 6.429	-	**2746.18 ± 216.81**	-
S-5	-	116.171 ± 112.376	-	2415.75 ± 69.16	-
S Overall	-	**96.249 ± 88.987**	-	2337.46 ± 392.74	-

Table 5.8: Comparison of runtimes and objective values for the worst-case scenario $\boldsymbol{z}^{worst}$ across T-SV, T-DV, and S instances solved by EXA-DET and META-DET over five runs with varying random seeds and a one-hour timeout. Δ% denotes the performance gap between the best objective value from META-DET and the value found by EXA-DET.

Breakdown of the Objective Values

To better understand the causes for the performance gaps, we once again decomposed the objective values into their constituent distance-, time-, and energy-related costs for the T-DV instances. This time, we compared the outcomes produced by EXA-DET with those from META-DET, as illustrated in Fig. 5.6.

On average, the energy costs associated with the solutions produced by META-DET are **16.1**% higher, while time costs increase by **6.3**%, and distance costs show only a marginal rise of **0.6**%. This suggests that META-DET performs comparably well in its routing decisions, occasionally even outperforming EXA-DET in terms of distance cost, as seen in instances like T-DV-2 and T-DV-4. However, it tends to be significantly less effective in its energy-related decisions and, to a lesser extent, in timing decisions, which leads to the overall worse objectives.

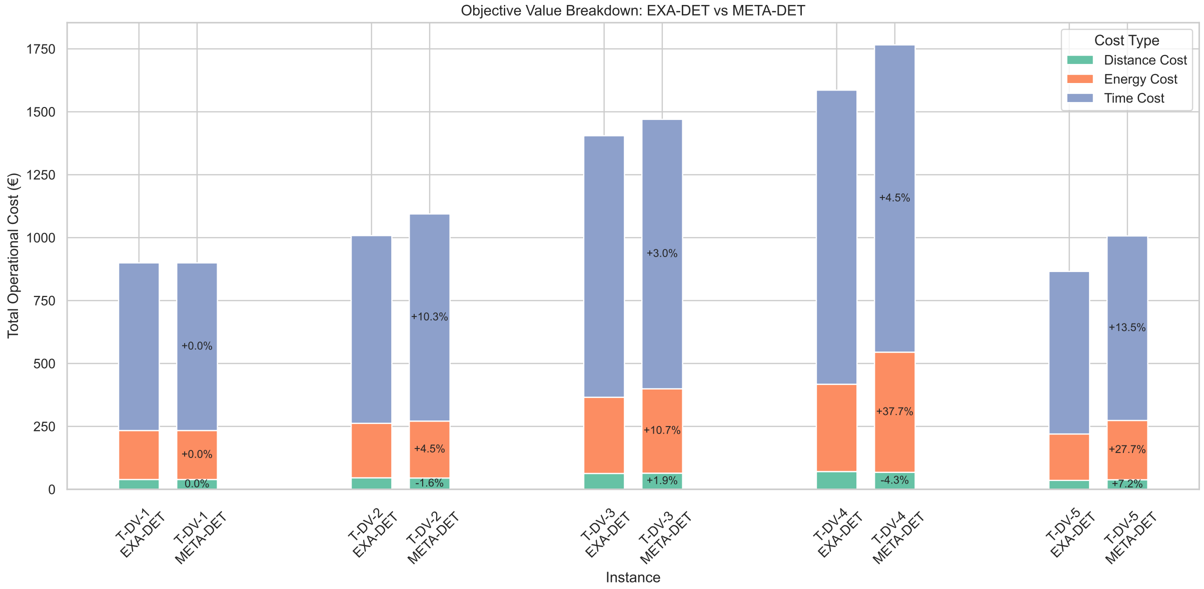


Figure 5.6: Breakdown of the objective values into distance, energy and time-related costs for the solutions to the T-DV instances for the worst-case scenario $\boldsymbol{z}^{worst}$, showing the percentage increase of the costs for the META-DET results compared to the EXA-DET results

Discussion

While the ALNS algorithm seems to be effective at modifying the route structure (first-stage decisions), it is less successful at anticipating the corresponding second-stage candidate solutions.

For instance T-SV-5, which is associated to the largest optimality gap, the routing and charging decisions made by META-DET are identical to those made by EXA-DET. However, the initial charging session is scheduled too early, leading to significantly higher time costs. Although our current ALNS strategy attempts to delay the first charging session as much as possible given the earliest service start time of the first customer, it does not account for the start times of subsequent customers. In this particular case, the early departure results in the vehicle arriving **166** minutes before the service window opens at the final customer, causing idle waiting time that contributes to the overall time cost. Incorporating the service start times of later customers could allow for further postponement of the initial charging session, thereby reducing the total time-related cost.

A possible explanation for the suboptimal energy decisions can be found in Fig. 5.7, which shows that the most effective destroy operator was the random charger removal operator. Currently, deleting and reinserting chargers is the only mechanism within ALNS to revise energy decisions. Consequently, the random removal operator was frequently applied in efforts to fine-tune these decisions, which significantly contributed to global improvements. However, the reliance on a single, random-based operator likely limited the algorithm's ability to consistently target the most detrimental charging sessions. Introducing more sophisticated charger removal operators could enhance the algorithm's ability to reduce energy-related costs.

Fig. 5.7 further indicates that removing customers that contribute the most to route distance, along with any preceding charger, was also effective. This again reflects an effort to refine charging decisions. Among the Shaw-based operators, time-based customer removal proved most effective. Interestingly, the random route removal operator outperformed others in its category. The randomness it introduced likely helped the algorithm escape local minima, as it forced the relocation of entire routes to different vehicles.

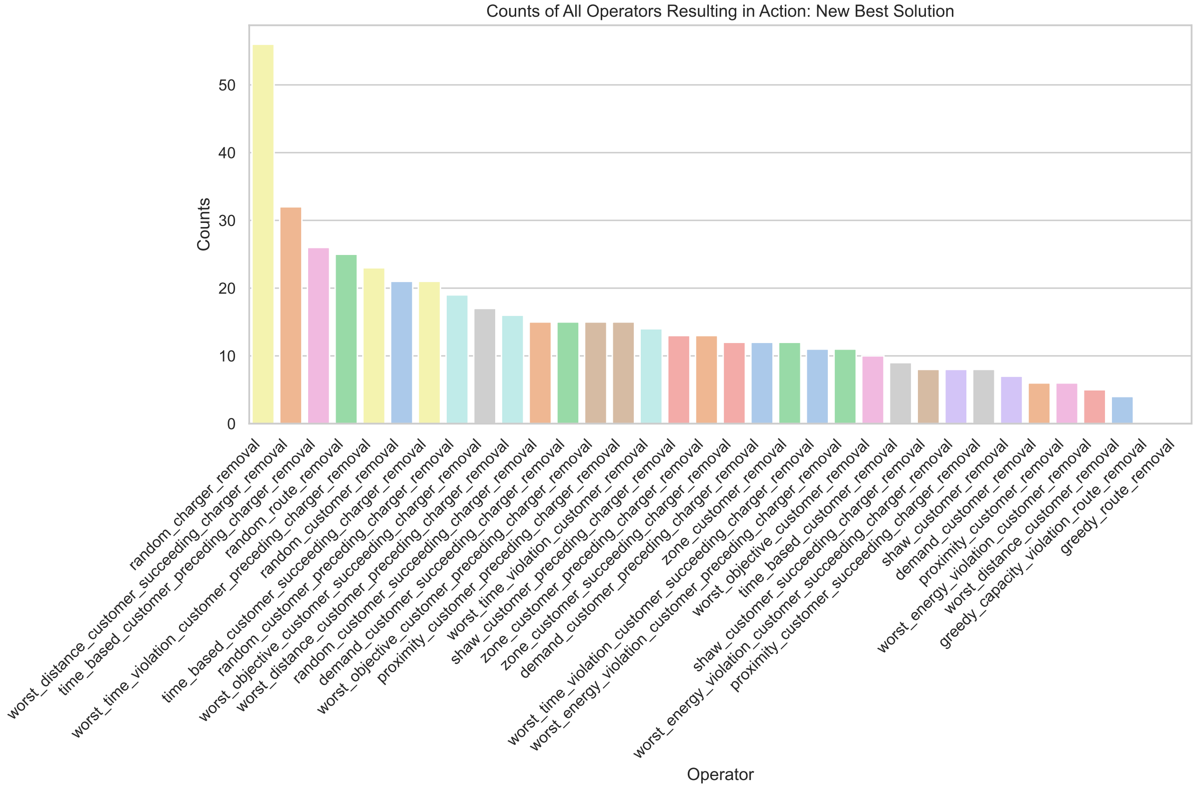
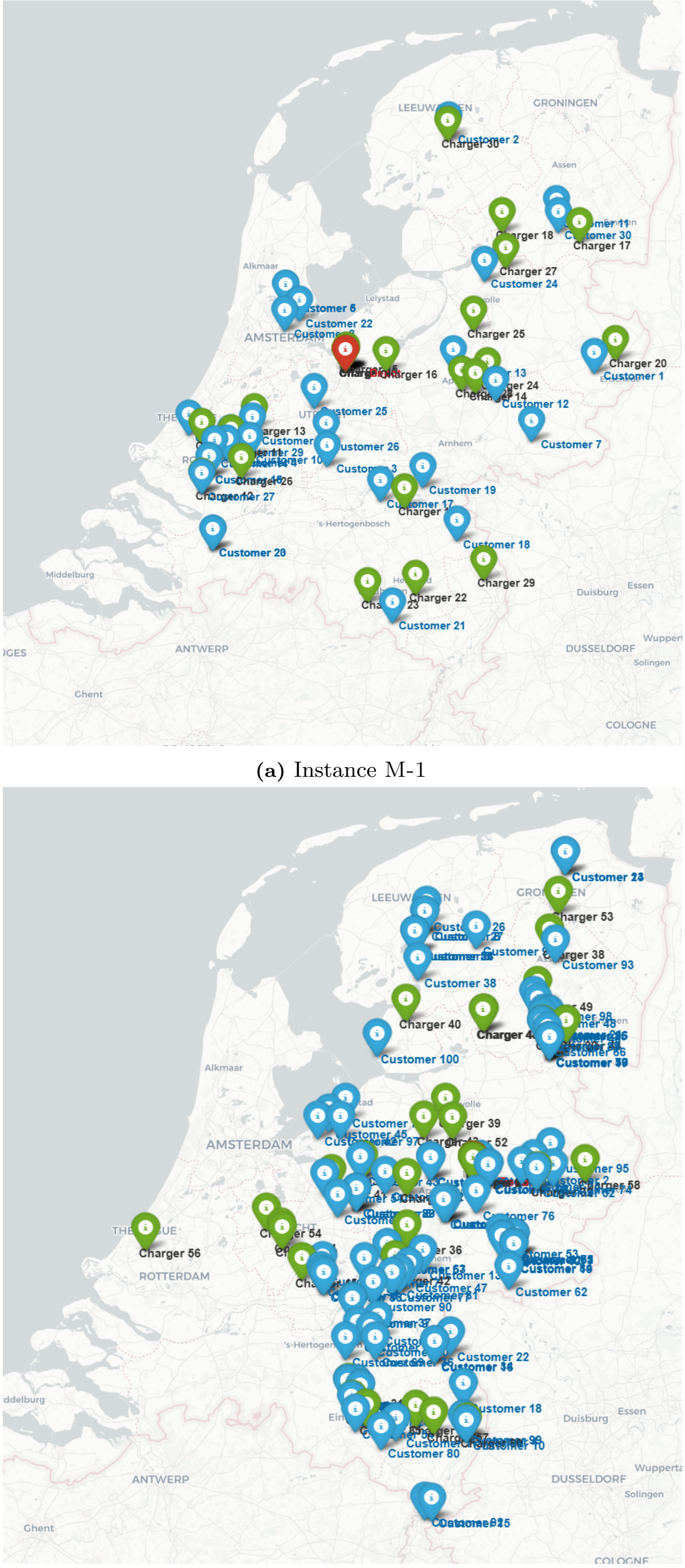


Figure 5.7: Counts across the T-DV instances for the number of times a specific ALNS operator in META-DET resulted in the discovery of an improved global solution

5.4.2. Assessing Scalability to Larger Instances

To evaluate the scalability of the ALNS metaheuristic, we conducted experiments on the two largest problem classes: Medium instances with 30 customers, 12 vehicles, and a total of 30 chargers, and Large instances featuring 100 customers, 40 vehicles, and 60 chargers. Fig. 5.8 illustrates representative examples from each class. As discussed in the previous section, EXA-DET already encountered difficulties with the Small instances, comprising just 10 customers and 3 vehicles, whereas META-DET was able to produce feasible but suboptimal solutions within a matter of minutes. For the more complex M and L instances, we observe that even the metaheuristic incurs significant computational times to converge to good solutions. Nonetheless, feasible solutions to all problem instances could be obtained within at most **10** minutes.



(a) Instance M-1

(b) Instance L-3

Figure 5.8: Geographical depiction of instances M-1 and L-3 with the locations of the depot, chargers, and customers on the map of The Netherlands

Medium Instances

We started our evaluation with exploratory tests, performing 100,000 ALNS iterations on the M instances (see to Fig. D.6). The results showed that after the 12,000-iteration limit used for the S instances, little further improvement was made to the objective values. However, we observed that many solutions still exhibited charging overlap violations, while all other constraints where satisfied. Hence, we increased the value of the charging overlap violation coefficient π_{ov} in the generalized cost function from 6 to 10 in an effort to penalize charging overlap more severely. We then performed five runs with 12,000 iterations, using the same random seeds as before. We report the runtime, the number of feasible solutions obtained, the best feasible objective value across all runs, and the maximum charging overlap observed in any infeasible run. The progression of the objective value over the iterations is illustrated in Fig. 5.9, while Table 5.10 presents the numerical results for the worst-case scenario $\boldsymbol{z}^{worst}$. For brevity, results for the nominal scenario are included in the Appendix D (see Fig. D.7 and Table D.2), although we briefly discuss them here.

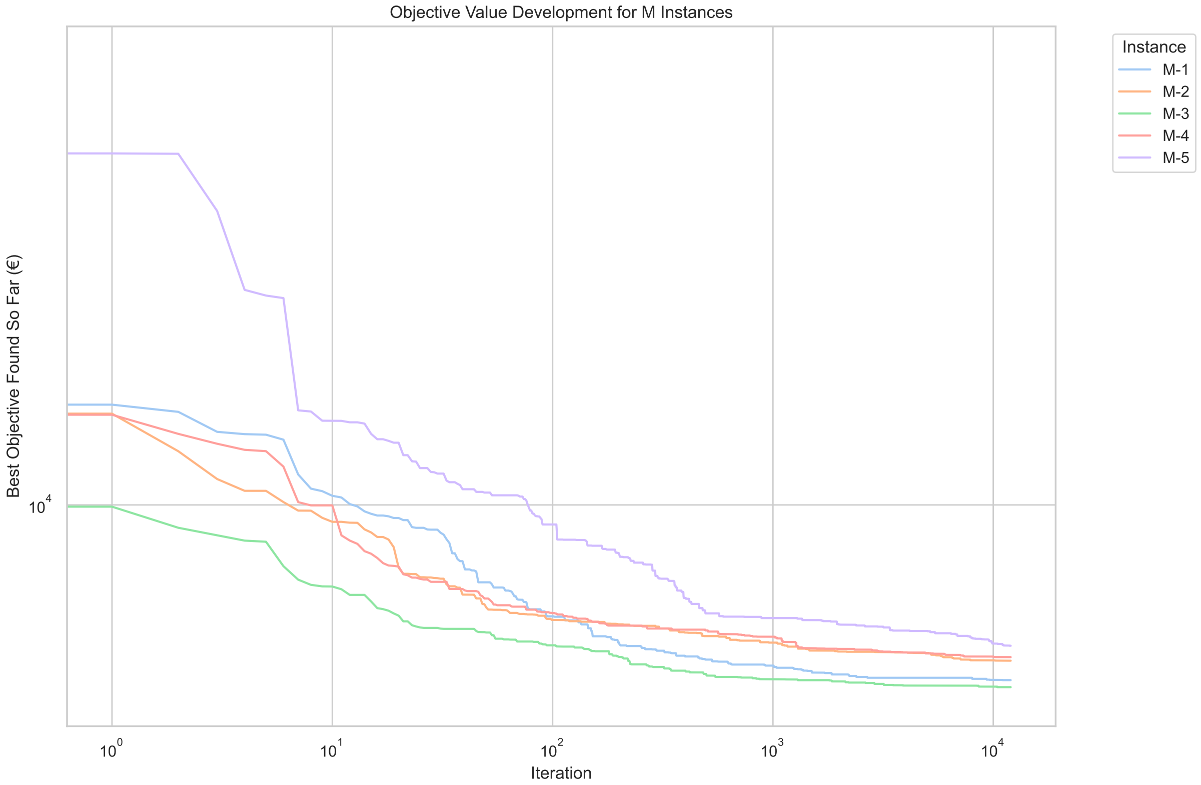


Figure 5.9: Development of the best objective value found using META-DET on M instances under the worst-case scenario $\boldsymbol{z}^{worst}$ over 12,000 iterations and five runs with varying random seeds. Both axes are represented on a logarithmic scale.

Instance	Runtime (s)	Num. Feas. Runs	Best Feas. Obj. (€)	Worst Overlap (min)
M-1	1281.07 ± 18.06	5/5	4640.11	–
M-2	1513.52 ± 21.27	4/5	5040.43	2.71
M-3	1368.32 ± 77.48	4/5	4398.66	5.04
M-4	1503.07 ± 16.23	5/5	5151.13	–
M-5	1478.71 ± 11.52	2/5	5347.04	6.46
M Overall	1428.94 ± 98.33	**20/25**	4398.66	**6.46**

Table 5.9: Performance comparison for solving the M instances under the worst-case scenario $\boldsymbol{z}^{worst}$ using the META-DET solver with a budget of 12,000 iterations over five runs with varying random seeds

Under the nominal scenario $\boldsymbol{z}^{nom}$, 18 out of 20 runs yielded feasible solutions. Specifically, feasible solutions were produced by 4 out of 5 runs for instances M-1 through M-4, while only 2 feasible solutions were found for M-5. Among the infeasible cases, the maximum observed charging overlap was 19 minutes. In contrast, under the worst-case scenario $\boldsymbol{z}^{worst}$, all runs successfully identified feasible solutions for M-1 and M-5. The results for the remaining instances remained unchanged, resulting in a total of 20 feasible solutions out of 25 runs. Notably, the maximum charging overlap among the infeasible cases decreased significantly to just 6.5 minutes.

Large Instances

For the L instances, our exploratory analysis was limited to the worst-case scenario and 12,000 iterations due to the considerable runtime, which averaged around **3 hours and 15 minutes**. However, as shown in Fig. 5.10, the objective values had not yet converged by the end of the 12,000 iterations. The numerical results are provided in Table 5.10.

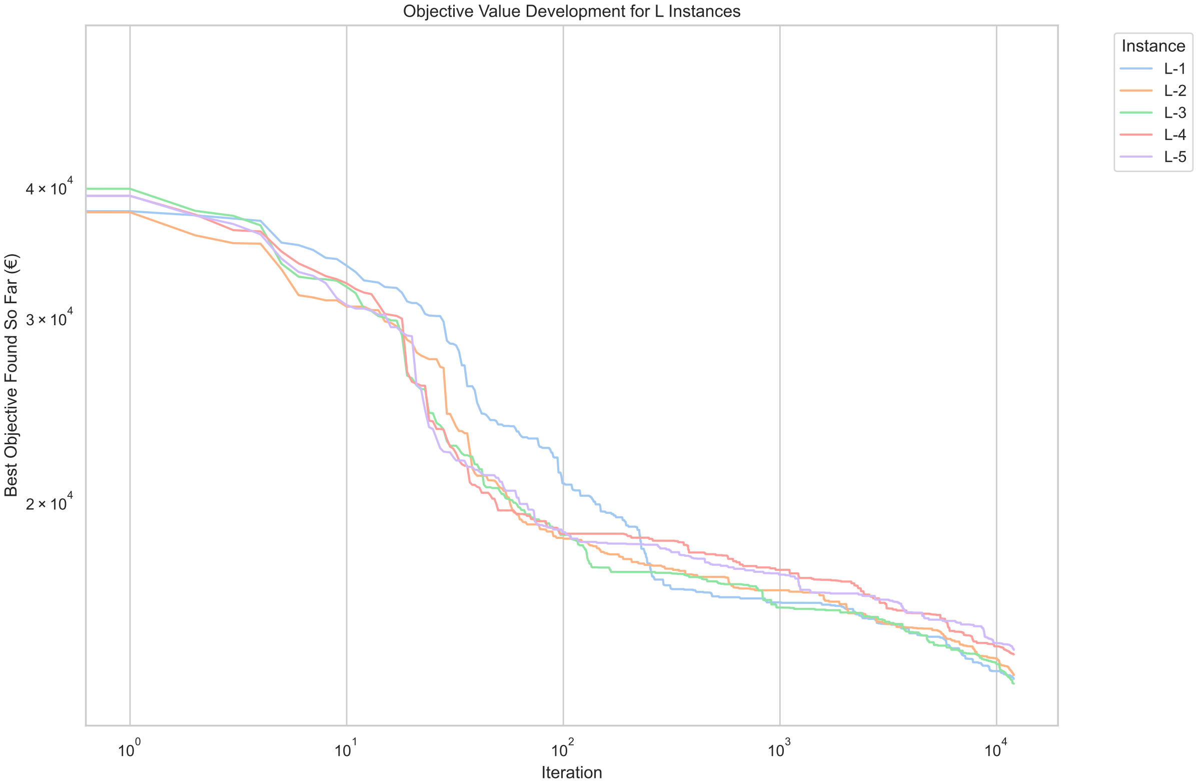


Figure 5.10: Development of the best objective value found using META-DET on L instances under the worst-case scenario $\boldsymbol{z}^{worst}$ over 12,000 iterations and five runs with varying random seeds. Both axes are represented on a logarithmic scale.

Instance	Runtime (s)	Num. Feas. Runs	Best Feas. Obj. (€)	Worst Overlap (min)
L-1	11531.73 ± 2925.99	4/5	13212.82	5.18
L-2	12519.37 ± 593.52	2/5	13885.61	4.71
L-3	11550.55 ± 2968.85	4/5	13377.73	4.85
L-4	11470.23 ± 2996.68	3/5	13944.93	32.82
L-5	11519.81 ± 3213.72	2/5	14131.91	15.53
L Overall	**11684.96 ± 2563.55**	**15/25**	13212.82	32.82

Table 5.10: Performance comparison for solving the L instances under the worst-case scenario $\boldsymbol{z}^{worst}$ using the META-DET solver with a budget of 12,000 iterations over five runs with varying random seeds

A key observation is the high variability in runtimes, with deviations of up to **45** minutes. These were largely caused by memory limitations that led to execution pauses during the final runs. Among the 25 runs, 15 yielded feasible solutions. The worst-performing solution exhibited a charging overlap exceeding 30 minutes, despite the elevated value of π_{ov}. Although no optimal benchmarks were available for direct comparison, a qualitative review of the solutions revealed that many vehicles returned to the depot with a high state of charge, suggesting suboptimal charging decisions. Furthermore, as previously noted, timing decisions were also suboptimal, leading to costly idle time.

Alternative Stopping Criterion
Due to the lengthy runtimes associated with 12,000 iterations, we also conducted experiments on the worst-case scenario using an alternative stopping criterion. Specifically, the ALNS procedure was terminated after 1,000 (M) and 100 (L) consecutive iterations without improvement to the current best objective value, aiming to obtain feasible solutions more quickly. The corresponding results are presented in Table 5.11.

Instance	**Runtime (s)**	**Num. Feas. Runs**	**Best Feas. Obj. (€)**	**Δ%**
M-1	218.74 ± 69.92	4/5	4689.44	+1.06
M-2	252.15 ± 58.64	3/5	5320.29	+5.55
M-3	214.59 ± 67.50	2/5	4548.36	+3.40
M-4	223.70 ± 73.64	4/5	5360.53	+4.07
M-5	188.65 ± 47.06	2/5	5877.95	+9.93
M Overall	**219.57 ± 62.04**	**15/25**	5159.31	**+4.80**
L-1	575.62 ± 208.84	2/5	16557.90	+25.32
L-2	487.84 ± 156.38	2/5	16985.10	+22.32
L-3	340.35 ± 202.93	3/5	17390.79	+30.00
L-4	416.76 ± 171.52	2/5	16818.66	+20.61
L-5	329.89 ± 85.32	**1/5**	17894, 14	+26.62
L Overall	**430.09 ± 182.31**	**10/25**	17129.32	**+24.97**

Table 5.11: Performance comparison for solving the M and L instances under the worst-case scenario $\boldsymbol{z}^{worst}$ using the META-DET solver over five runs with different random seeds, with respective budgets of 1,000 and 100 consecutive iterations without improvement. Δ% denotes the difference in best feasible objective value with the best known solution from the previous experiments.

The resulting runtimes decreased significantly, showing overall speedup factors of **6.5** and **27** for the M and L classes, respectively. Feasible solutions could be identified for each of the instances, with 15/25 runs yielding feasible solutions for the M class and 10/25 for the L class. Compared to the experiments using 12,000 iterations, the increase in objective values was moderate (**4.80**%) for the M instances. However, for the L instances, the objective values rose sharply by nearly **25**%, indicating that this stopping criterion was ineffective for finding cost-efficient solutions.

Discussion
While the results demonstrate that the ALNS metaheuristic is capable of generating feasible solutions for Medium and Large instances within a limited time-frame, the cost-efficiency of these solutions varied heavily and could not be assessed properly. As a result, the solutions may be highly inefficient and of limited practical value.

Moreover, the experiments conducted were largely exploratory, involving only a narrow range of ALNS configurations and parameters tested on a small set of instances. Due to time constraints, we were unable to perform comprehensive parameter tuning, such as a grid search, which might have revealed more effective hyper-parameters to further reduce charging overlap violations and improve solution quality. When we increased the penalization coefficients for charging overlap ($\pi_{ov} \in [20, 100]$) in isolation, the resulting solutions exhibited additional time window violations, ultimately worsening the overall solution quality.

5.4.3. Assessing Robustness under Budgeted Uncertainty

In our final experiment, we evaluate the integration of all methodological components that shape CERT$\mathcal{E}$ROUTE by analyzing the performance of EXA-ARO as the RMPSOLVER within the CCG framework, in combination with the OLSP algorithm for identifying violating scenarios under the locally budgeted uncertainty set $\mathcal{U}_{local}$. This evaluation is similar to the one performed in Subsection 5.3.2 but now we use the metaheuristic solution approach and consider more computationally demanding instances from the MEDIUM instance class. The results show that META-ARO is capable of producing highly robust solutions for these instances within a practical runtime, highlighting the potential of our approach.

To assess the robustness of the solutions produced by META-ARO, we sampled 100 extreme points from the corresponding locally budgeted uncertainty sets. The reduced sampling size was chosen due to the significant time required for sampling and verification, ranging from 10 to 20 minutes depending on the instance and configuration. A single run was performed per instance and configuration of the uncertainty set. As before, we employed the EXA-DET solver to find feasible second-stage solutions based on given first-stage decisions and a sampled scenario. For both Γ_ν and Γ_t, we considered the absolute budget values corresponding to 0%, 10%, 25%, 50%, and 100% of the total number of locations and/or edges attaining their maximum values. We set the stopping criterion for the ALNS procedure to a maximum of 1,000 iterations without improvement to the current best objective. The results are summarized in Table 5.12.

The average runtime across all instances and uncertainty set configurations was **7.35** minutes, with on average **2.2** RMP iterations required for convergence. The maximum number of RMP iterations observed was **8**. The low number of RMP iterations can be attributed to the effective pessimization achieved via the OSLP algorithm. As a result, the pseudo-polynomial time complexity of the ALNS algorithm did not pose a significant challenge, despite the high dimensionality of the uncertainty set ($l = 31 + 3721$). The resulting solutions demonstrated high levels of robustness, averaging **92.6**%.

Discussion

The solutions obtained for the nominal scenario ($\Gamma_\nu = 0$, $\Gamma_t = 0$) did not yield the greatest relative improvement in objective value compared to the worst-case scenario. For instance, the solutions corresponding to configurations ($\Gamma_\nu = 0$, $\Gamma_t = 372$) and ($\Gamma_\nu = 3$, $\Gamma_t = 3721$) resulted in lower average objective values across the instances. This suggests that the solutions found for the nominal scenario were sometimes suboptimal. This is not an unexpected outcome given that the metaheuristic does not guarantee optimality.

On average, more than two RMP iterations were needed to reach convergence under the budget configuration ($\Gamma_\nu = 31$, $\Gamma_t = 3721$), which corresponds to the box uncertainty set $\mathcal{U}_{\text{box}}$. This indicates that the overall worst-case scenario $\boldsymbol{z}^{\text{worst}}$ was not consistently identified in the first RMP iteration, in contrast to the results observed for EXA-RO and EXA-ARO. This difference arises from a deliberate design choice in META-ARO, where pessimization was restricted to variables associated with locations currently included in the route, while all other uncertain parameters were fixed at their nominal values (see Section 4.3). This choice was made following preliminary experiments in which the pessimization LP models grew too large to fit into memory due to the excessive number of decision variables required for the MEDIUM and LARGE instances. Consequently, the OSLP procedure lost its termination guarantees, and the overall worst-case scenario could not always be immediately captured. Nonetheless, META-ARO consistently achieved robustness against the worst-case scenario within at most three iterations.

The lowest robustness score observed was **60**%, corresponding to the configuration ($\Gamma_\nu = 31$, $\Gamma_t = 372$). For this configuration, three out of five solutions achieved 100% robustness. In the remaining two cases, META-ARO was unable to identify a feasible solution during the initial RMP iteration, causing the CCG procedure to terminate prematurely. The resulting solution, already infeasible under the nominal scenario, was tested against 100 extreme scenarios and failed in all cases. Hence, this premature convergence was due to the ALNS stopping criterion, rather than a failure of the OSLP to identify violating scenarios.

The most significant cost reductions were associated with lower values of the service duration budget Γ_ν, whereas for the smaller T-SV instances (see Subsection 5.3.2), such reductions were linked to lower traversal duration budgets Γ_t. This may be attributed to the larger fleet size and higher vehicle-to-customer ratio of the M instance class compared to the T-SV class, which results in vehicles serving geographically closer customers. Consequently, traversal durations are reduced, and the proportion of time spent at customer locations becomes more prominent in the overall route duration.

Γ_ν	Γ_t	Runtime (s)	Avg. Rel. Obj. Improv. (Δ%)	Avg. Num. RMP Iterations	Sample Robustness (%)
0	0	103.68 ± 26.29	**-5.61 ± 8.19**	1.00 ± 0.00	100.00 (z^{nom})
0	372	562.94 ± 349.05	**-11.65 ± 3.18**	2.60 ± 1.36	100.00
0	930	371.50 ± 143.44	-5.19 ± 7.38	1.60 ± 0.80	99.80
0	1860	247.71 ± 73.49	-5.38 ± 7.95	1.20 ± 0.40	99.80
0	3721	246.83 ± 73.15	-5.38 ± 7.95	1.20 ± 0.40	98.80
3	0	399.29 ± 143.43	-10.23 ± 1.81	2.60 ± 0.49	93.80
3	372	475.89 ± 259.68	-10.23 ± 3.45	2.20 ± 1.17	91.20
3	930	430.58 ± 205.03	-5.17 ± 7.23	2.00 ± 0.89	94.20
3	1860	443.13 ± 180.55	-8.80 ± 4.39	2.60 ± 1.02	92.40
3	3721	453.45 ± 186.64	**-9.68 ± 4.55**	2.40 ± 1.02	88.80
7	0	426.89 ± 259.18	-6.47 ± 8.03	2.00 ± 1.10	76.40
7	372	399.73 ± 167.50	-9.11 ± 4.03	2.20 ± 0.75	84.40
7	930	717.94 ± 478.19	-6.92 ± 2.27	3.40 ± 1.96	99.60
7	1860	348.45 ± 69.38	-3.12 ± 5.96	2.20 ± 0.75	92.40
7	3721	410.42 ± 94.64	-5.54 ± 1.70	2.60 ± 0.49	99.80
15	0	522.62 ± 191.97	-4.32 ± 6.54	2.60 ± 1.02	85.80
15	372	529.72 ± 201.59	-6.48 ± 2.05	2.60 ± 0.80	98.20
15	930	566.50 ± 214.40	-5.30 ± 3.73	2.80 ± 0.75	98.20
15	1860	599.53 ± 161.11	-4.74 ± 1.34	3.00 ± 1.10	98.20
15	3721	586.92 ± 167.04	-4.76 ± 1.36	2.80 ± 1.17	98.20
31	0	362.90 ± 70.69	-3.05 ± 1.87	2.60 ± 0.49	100.00
31	372	507.97 ± 226.62	-1.19 ± 5.03	2.20 ± 0.98	**60.00**
31	930	702.86 ± 277.24	-5.78 ± 2.78	3.40 ± 1.36	80.00
31	1860	304.29 ± 89.46	-0.74 ± 1.48	2.20 ± 0.40	100.00
31	3721	306.84 ± 89.30	0.00 ± 0.00	**2.20 ± 0.40**	100.00 (z^{worst})
Overall		**441.14 ± 245.60**	-5.79 ± 5.67	**2.33 ± 1.11**	**92.61**

Table 5.12: Comparison of the averaged runtimes, relative objective improvement, number of required RMP iterations, and sample robustness resulting from solving the instances from the M class with locally budgeted uncertainty set $\mathcal{U}_{local}$ under different uncertainty budgets Γ_ν for service duration and Γ_t for traversal duration using the META-ARO and assessing robustness using 100 randomly sampled scenarios from $ext(\mathcal{U}_{local})$ of the corresponding locally budgeted uncertainty sets. Δ% denotes the percentage difference with the worst-case objective value found for z^{worst}.

6 Conclusion and Discussion

This chapter concludes the thesis by addressing the main research question:

Main Research Question

How can the cost-efficiency of Heavy Goods Electric Vehicle (HGEV) fleet operations in road freight transport be optimized through integrated vehicle routing and charging scheduling, while ensuring robustness under time and energy consumption uncertainty, and maintaining computational tractability?

We first reflect on the work carried out in the previous chapters to address the research sub-questions and the main research question in Section 6.1. We then provide guidance on the practical application of the proposed methodology informed by our research findings in Section 6.2. Lastly, we discuss limitations of our work and outline possible avenues for future work in Section 6.3.

6.1. Addressing the Research Questions

In Chapter 2, we started our thesis by presenting a comprehensive review of the existing literature on optimization under uncertainty in the context of HGEV fleet planning, addressing the research sub-question:

Research Sub-question 1

What is the current state of research on optimization under uncertainty in HGEV fleet planning, and what gaps can be identified?

Our review highlights a solid foundation in formulating and solving vehicle routing problems, including those involving electric vehicles under energy consumption uncertainty, through methods such as stochastic optimization [7, 68], robust optimization [4, 69, 73, 113] and reinforcement learning [128].

We established that uncertainty or variability in the time domain remained underexplored, with only a few works addressing time-varying traversal durations [7, 8] and stochastic waiting times at charging stations [111]. Truly uncertain traversal durations and service durations at customers were explored [168, 169], although not in a setting with electric vehicles. The link between time and energy consumption uncertainty was only formalized before by Basso, Kulcsár, and Sanchez-Diaz [7], who precomputed estimates on traversal velocities and performed stochastic optimization using a velocity-dependent energy consumption function. However, the integration of joint time and energy consumption uncertainty in a robust optimization framework was not considered before.

The joint optimization of depot charging operations and vehicle routing, an essential step towards integration with energy management systems, was also not yet explored under uncertain operating conditions. Finally, none of the studies focusing on heavy-duty electric vehicles [7, 4, 69, 73] addressed large-scale HGEV operations in realistic European settings.

Modelling Uncertainties and Formulating an Optimization Model

In Chapter 3, we set out to close the identified research gaps, addressing the research question:

Research Sub-question 2

How can uncertainties in time and energy consumption be formally represented and incorporated into an optimization model for HGEV charging scheduling and routing?

The Heavy Goods Electric Vehicle Fleet Planning Problem under Time and Energy Consumption Uncertainty (HGEV-FPP-TECU) introduced in this work provides a formalism for the charging scheduling and routing problem at hand, incorporating a set of uncertain parameters including the energy consumption and traversal duration over links in the transport network, next to the charging rates at chargers and service and waiting durations at customers. To model uncertainties in the time domain, one can employ box and locally budgeted uncertainty sets, which use symmetric intervals around nominal values and require minimal distributional information.

Energy consumption uncertainty beyond time and mass related factors can be abstracted away from the optimization problem by introducing a linear and parameterized energy consumption function that estimates energy consumption given an HGEVs load level and its uncertain velocity over a link in the network. Its parameters represent the nominal energy consumption scenario and can be precomputed using Monte Carlo simulations and function approximation. A robustness multiplier can be applied to the predicted energy consumption to adjust protection against energy-related uncertainties in a way akin to chance-constraints in stochastic optimization.

The problem can then be formulated as a linear adaptive robust optimization model that guarantees the feasibility of routing and charging schedules across the entire uncertainty set. Our proposed model explicitly accounts for the initial depot charging session and enforces non-overlapping constraints on charging sessions. The adaptive design of this optimization model enables a two-stage decision process where routing decisions, including charger visits, are made prior to the uncertainty realization, anticipating the worst-case scenario, while charging quantity and timing related decisions remain flexible and can be refined upon route execution to prevent overly conservative and cost-inefficient solutions.

Solution Methods for the Optimization Problem

While the defined uncertainty sets can be represented by their extreme points to limit the number of constraints in the optimization model, the vast number of such points can still render the optimization problem computationally infeasible. Hence, we addressed the following research sub-question:

Research Sub-question 3

Which solution methods can be employed to robustly solve the formulated optimization problem, while ensuring computational tractability?

The Column and Constraint Generation (CCG) method can be employed to maintain computational tractability by minimizing the number of scenarios considered. It begins by solving the model for the nominal scenario and then iteratively applies a pessimization procedure to identify and solve for scenarios that render the current routing decisions infeasible. This iterative process continues until no further violating scenarios can be identified, thereby ensuring the robustness of the solution.

The novel One-Step Look-ahead Pessimization (OSLP) algorithm can be employed to facilitate and accelerate this procedure. It selectively targets only those scenarios that are provably infeasible, thereby avoiding the inclusion of unnecessary scenarios in the Restricted Master Problem (RMP). Both its maximization and minimization procedures can be formulated as linear programs, allowing for efficient resolution using off-the-shelf solvers.

The resulting CCG method was found to be prone to premature convergence, as OSLP cannot guarantee the discovery of scenarios for which constraint violations occur more than one step ahead in the routing schedule. Consequently, robustness against all scenarios within the uncertainty set cannot be assured under budgeted uncertainty. Nevertheless, our experiments demonstrated that the resulting solutions exhibited a high degree of robustness in practice.

To maintain computational tractability for large problem instances, where solving the RMP exactly becomes infeasible, a multi-scenario Adaptive Large Neighbourhood Search (ALNS) metaheuristic can be employed. The variant proposed in this work explicitly incorporates second-stage decisions to enable an efficient exploration of the solution space and ensure energy-feasible solutions.

Building on these solution techniques, the CERT$\mathcal{E}$ROUTE framework provides a comprehensive suite of solvers that can be employed for deterministic, robust, and adaptive robust optimization of the HGEV-FPP-TECU.

Experimental Evaluation of the Proposed Methodology

In Chapter 4, we detailed the process of generating realistic HGEV-FPP-TECU instances by sampling representative customer and charging locations, defining plausible vehicle and cost parameters, and bounding uncertainties in the time domain through informed assumptions and expert evaluation. We showed how key parameters of the energy consumption function can be estimated through function approximation, using Monte Carlo simulations applied to a longitudinal vehicle dynamics model. This approach accounts for both environmental and vehicle-specific uncertainties, and incorporates representative duty cycles that capture real-world HGEV velocity and acceleration patterns.

Building on this experimental framework, we conducted a series of computational experiments, presented in Chapter 5, to evaluate our proposed methodology, addressing the research sub-question:

Research Sub-question 4

How do the proposed solution methods perform when applied to realistic HGEV fleet planning scenarios, in terms of robustness, cost-efficiency, scalability and adaptability?

Our experiments revealed operational cost differences of up to **12**% when optimizing for the worst-case scenario versus the nominal case, with time-related costs accounting for more than **70**% of the total operational cost. Sensitivity analysis highlighted that uncertain service durations, due to their wide intervals, significantly impacted cost-efficiency within a single instance, while traversal durations emerged as the dominant factor across instances when considering equal variability for each uncertain parameter.

The exact deterministic solver (EXA-DET) could solve small-scale, single-vehicle instances to optimality. While feasible solutions were quickly found for two-vehicle instances, optimality remained unconfirmed within the set time limit of an hour. For instances with three vehicles and ten customers, the solver typically failed to find feasible solutions within the time limit.

In contrast, the metaheuristic deterministic solver (META-DET), based on ALNS, exhibited large optimality gaps of up to **36**% on these small instances, primarily due to suboptimal charging and timing decisions. Nevertheless, it demonstrated strong scalability, solving instances with up to 100 customers and 30 vehicles to feasibility within **10** minutes, although the high variability in objective values limits conclusions about the cost-efficiency of its solutions.

Adaptability proved to be a key advantage in robust optimization. The adaptive solver (EXA-ARO) consistently outperformed its static counterpart (EXA-RO), achieving **6** to **7**% cost reductions under the nominal scenario when considering the box uncertainty set. This came at the expense of slightly increased runtimes due to a larger number of decision variables. Despite the risk of premature convergence in the OSLP algorithm, both solvers maintained strong performance under budgeted uncertainty, achieving **99**% overall robustness against randomly sampled extreme points.

Finally, the metaheuristic adaptive solver (META-ARO) achieved **93**% robustness on medium-sized instances with 30 customers and 12 vehicles under budgeted uncertainty. It required only **2** to **3** RMP iterations on average to converge. This highlights the effectiveness of our approach, which combines the CCG procedure, the novel OSLP pessimization algorithm, and a metaheuristic RMP solver to deliver highly robust solutions while maintaining computational tractability.

Answer to the Main Research Question
In conclusion, our findings demonstrate that the cost-efficiency of HGEV fleet operations in road freight transport can be effectively optimized, while maintaining computational tractability and ensuring high levels of robustness against time and energy consumption uncertainties, through the application of the proposed CERT$\mathcal{E}$ROUTE framework. Specifically, this involves:

- **Formulating the problem** as a linear adaptive robust optimization model that integrates vehicle routing and charging scheduling and explicitly incorporates time uncertainty.
- **Representing time uncertainty** using (locally budgeted) uncertainty sets that account for uncertain service, waiting, traversal and charging durations.
- **Modelling energy consumption** using a simulation-based function that accounts for environmental and vehicle-specific uncertainties, vehicle mass, acceleration, and velocity.
- **Balancing robustness and computational efficiency** through tailored solution methods that combine:
  - CCG to limit the number of considered scenarios in the RMP,
  - OSLP to identify and incorporate provably infeasible scenarios into the RMP,
  - A suite of exact and metaheuristic solvers based on ALNS to address the RMP.
- **Reducing conservatism** via two-stage robust optimization, where routing decisions are made prior to route execution, but timing and charging decisions can be tweaked during route execution.

6.2. Guidance on Practical Application

To guide the practical application of the CERT$\mathcal{E}$ROUTE framework, we present a decision diagram (see Fig. 6.1) that synthesizes our methodology and research findings. This diagram is designed to help fleet planners select the most suitable solver and uncertainty set by considering key factors such as data availability, problem scale, robustness requirements, and operational constraints.

Further Elaboration

Our experiments revealed that most instances involving 10 customers and 3 vehicles could not be solved exactly within an hour. Therefore, we recommend using metaheuristic solvers, which demonstrated significantly better scalability, for problems of this size and beyond. However, if hardware resources permit, one can attempt to solve such instances exactly to obtain stronger optimality guarantees.

In practical scenarios where only nominal or average-case estimates are available for the uncertain parameters, deterministic optimization under the estimated scenario is the only option.

If cost-efficiency is most important, and robustness is not a priority, such that constraint violations under other scenarios are allowed, optimizing directly for the nominal scenario yields cost-efficient and pragmatic solutions.

If 100% robustness needs to be guaranteed and cost-efficiency is less important, we recommend exact deterministic optimization considering the worst-case scenario, as this method is quick and guarantees optimality under the worst-case scenario. For larger instances, the metaheuristic approach can be used although this provides no guarantees on the feasibility of the solution.

If robustness and cost-efficiency need to be balanced, we recommend using a CCG-based solver in combination with an uncertainty set that is as compact as possible to limit conservatism. If information is available on the number of uncertain parameters that can attain their worst-case value at the same time, through covariance data, we recommend to use the budgeted uncertainty set. If such data is available per parameter type, a locally budgeted uncertainty set can be constructed to further limit conservatism. If no such data is available, the box uncertainty set is the only option.

If the vehicle energy consumption and time uncertainty realizations can accurately be tracked, so that adaptive optimization during route execution is possible, we recommend to use the adaptive solver variants to produce the routing solution and the deterministic solvers during route execution to find the best decisions given the routing solution and the observed scenario at hand.

However, inaccurate measurements during route execution can compromise robustness, and constantly shifting optimal timing decisions may prevent reliable arrival time estimates, complicating customer communication and integration with reservation systems for chargers. In such cases, we recommend using the non-adaptive solver variants to produce a single solution prior to the route execution with stronger robustness guarantees, albeit potentially at the cost of reduced cost-efficiency.

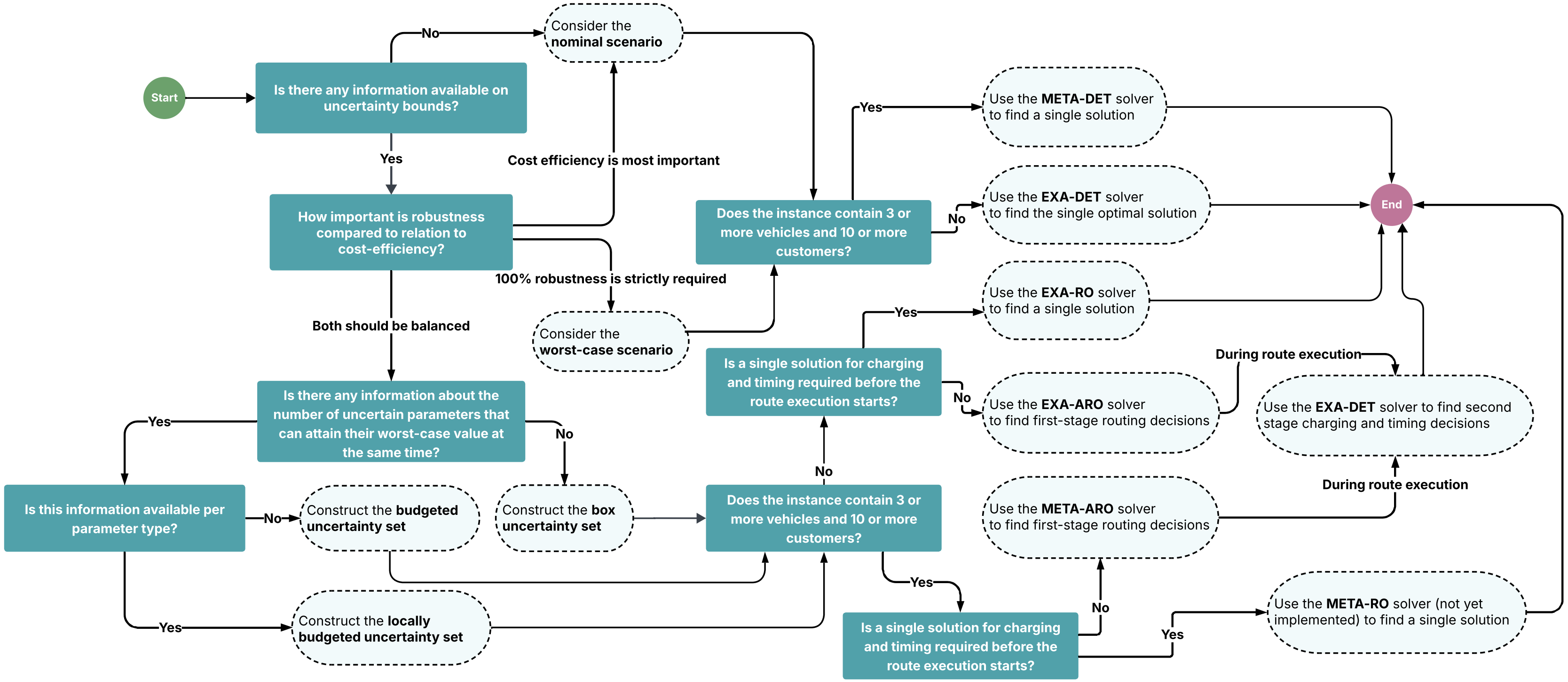

Start
Is there any information available on uncertainty bounds?
No
Consider the **nominal scenario**
Yes
Cost efficiency is most important
How important is robustness compared to relation to cost-efficiency?
100% robustness is strictly required
Consider the **worst-case scenario**
Both should be balanced
Is there any information about the number of uncertain parameters that can attain their worst-case value at the same time?
Yes
No
Is this information available per parameter type?
No
Construct the **budgeted uncertainty set**
Construct the **box uncertainty set**
Yes
Construct the **locally budgeted uncertainty set**
Does the instance contain 3 or more vehicles and 10 or more customers?
Yes
Use the **META-DET** solver to find a single solution
No
Use the **EXA-DET** solver to find the single optimal solution
End
Is a single solution for charging and timing required before the route execution starts?
Yes
Use the **EXA-RO** solver to find a single solution
No
Use the **EXA-ARO** solver to find first-stage routing decisions
During route execution
Use the **EXA-DET** solver to find second stage charging and timing decisions
During route execution
Does the instance contain 3 or more vehicles and 10 or more customers?
No
Yes
Is a single solution for charging and timing required before the route execution starts?
No
Use the **META-ARO** solver to find first-stage routing decisions
Yes
Use the **META-RO** solver (not yet implemented) to find a single solution

Figure 6.1: Decision diagram representation of the CERT$\mathcal{E}$ROUTE framework for charging scheduling and routing under time and energy consumption uncertainty

6.3. Limitations and Avenues for Future Work

In this final section of the thesis, we discuss the main limitations of our work and outline possible avenues for future research.

Modelling HGEV Fleet Operations

While we improved the realism of existing HGEV fleet planning models by explicitly incorporating depot charging scheduling and enforcing non-overlapping constraints on charging sessions, the current model's applicability remains limited beyond the considered distribution transport setting. This is due to simplifying assumptions such as single-day operations, single trips starting and ending at the same depot, and the exclusion of back-hauls or pickups along the route.

As battery and charging technologies continue to evolve, enhancing vehicle range and reducing charging durations, the simplified setting considered here may no longer present major obstacles to ensuring time and energy feasibility of the routes, even under time and energy consumption uncertainty. However, we believe that significant operational cost-efficiency (OPEX) gains can still be obtained through co-optimization with energy management systems, which presents a promising avenue for future work. Further cost-efficiency improvements can likely be achieved through the co-optimization of scarce asset management and operational planning. For instance, limiting the number of deployed chargers would introduce additional constraints on the HGEV operations, but would also reduce infrastructure requirements, ultimately lowering capital expenditures (CAPEX).

Future research could also focus on extending the model to international transport settings, where maintaining time and energy feasibility is expected to remain more challenging due to the increased uncertainty stemming from longer travel distances and a stronger reliance on public and semi-public charging networks. Such an extension would require incorporating more practical elements such as driver rest regulations and the availability of suitable parking infrastructure, both of which are essential for ensuring operational viability in long-haul transport.

Dealing with Energy Consumption Uncertainty

In this work, energy consumption was modelled as a function of the uncertainty set in the time domain, essentially decoupling it from the robust optimization framework. This approach was chosen to preserve the inherent relationship between traversal duration and energy consumption, as mediated by a vehicle's velocity. Introducing energy consumption uncertainty as an independent dimension within the box or locally budgeted uncertainty set would risk generating unrealistic scenarios. For instance, it could lead to cases where energy consumption reflects heavy congestion on an edge, while the corresponding traversal duration suggests free-flow conditions.

However, our modelling choice also comes with limitations. By not explicitly incorporating energy consumption uncertainty, beyond that arising from variable traversal durations, into the robust optimization framework, we were unable to establish any formal guarantees on true energy feasibility. Instead, we relied on Monte Carlo simulations and mean-value approximations of the energy consumption function, reducing the problem to nominal optimization with respect to energy consumption. That is, we more ore less transferred the limitation present in the work by Basso et al. [51], who accounted for stochastic energy consumption but considered nominal traversal durations, from the time domain to the energy domain. While the use of a larger robustness multiplier could offer a heuristic means to account for energy uncertainty, providing some probabilistic guarantees, it could also lead to overly conservative estimates, particularly under low-velocity conditions where energy consumption tends to be overestimated.

The limited guarantees on energy feasibility also undermine the timing feasibility, as charging durations are directly dependent on the estimated energy consumption and demand. Future research could address this limitation by exploring more expressive uncertainty models, such as ellipsoidal uncertainty sets, which can better capture the interdependence between energy consumption and traversal duration. Additionally, a promising direction would be to adopt a non-linear optimization model that directly incorporates a longitudinal vehicle dynamics model. Although this would result in a second-order cone program, modern solvers like Gurobi [66] are capable of solving such problems to optimality. Techniques such as arc segmentation, as proposed by Kim, Park, and Jeong [174], could be employed to enable robust optimization over a detailed road network, considering uncertain velocities over every edge. In such a framework, the uncertain energy-consumption parameters currently handled via simulation could also be embedded directly into the uncertainty set.

Incorporating Operational Data

Due to the limited availability of operational data, the uncertainty bounds in our experiments were established based on expert judgment and informed assumptions. Additionally, the uncertainty budgets were more or less generated by uniformly selecting points between the uncertainty bounds. Due to the absence of empirical validation, these bounds and budgets may not accurately capture the level of uncertainty typically encountered in practice.

In the presence of more operational data, future research could address this limitation by constructing data-driven uncertainty sets, leveraging methodologies such as those proposed by Li, Yorke-Smith, and Keviczky [158]. Additionally, when sufficient data is available to reliably estimate the underlying probability distributions, stochastic optimization techniques may offer a promising alternative. Such approaches can provide strong probabilistic robustness guarantees while mitigating the conservativeness typically associated with robust optimization.

Improving the Experimental Evaluation

A key limitation lies in the evaluation of our novel OSLP algorithm. We did not benchmark it against the naive pessimization method from Zeng and Zhao [155] or the bilinear pessimization approach used by Jeong et al. [113]. While we empirically demonstrated the shortcomings of the naive method, we only discussed the theoretical limitations of the bilinear approach. We initially intended to implement a bilinear approach ourselves but encountered significant modelling overhead in Pyomo. This led us to develop OSLP instead. Only later did we discover out-of-the-box two-stage optimization tools such as those proposed by Wiebe and Misener [218] and Sherman et al. [219]. Future research could utilize these tools to implement the bilinear pessimization model or investigate alternative two-stage optimization strategies that avoid premature convergence, enabling a comprehensive comparison with and potential improvements over OSLP.

Additionally, the scope of our experimental evaluation was limited by the small number of benchmark instances (25), most of which involved relatively small problem sizes ($\leq$ 10 customers). These instances, while useful for rapid experimentation, are not really representative of real-world planning scenarios. Due to time constraints, we focused primarily on these smaller instances, which could be solved to optimality using exact methods. This facilitated the large number of repeated evaluations required for the sensitivity analysis and the robustness assessments under various budgeted uncertainty sets. Although we demonstrated the scalability of our deterministic metaheuristic optimization approach on instances with up to 100 customers, the integration of the metaheuristic within the robust column and constraint generation framework was only tested on instances with up to 30 customers. In contrast, established VRP literature includes studies with up to 1000 customers [220]. However, such such works often rely on simplified models and synthetic instances that omit many real-world constraints.

Beyond instance size, the diversity of our test cases also remained limited. Our experiments focused solely on uncertainties in traversal and service durations within the time domain, while charging rates and waiting durations were treated deterministically. Additionally, although our model supports heterogeneous fleets, we only evaluated a homogeneous fleet setting. The time windows were deliberately set to be wide in order to find feasible instances more easily during instance generation. However, in practical applications, narrower windows would likely amplify the impact of charging rates and battery capacities, an aspect that warrants further investigation.

Although we considered the EVRPTW benchmark proposed by Schneider, Stenger, and Goeke [30] during the early stages of development, we did not evaluate our final metaheuristic on standard deterministic EVRPTW benchmarks. This was mainly due to the inclusion of the initial depot charging session in our model. This rendered many instances from commonly used benchmarks, including those by Schneider, Stenger, and Goeke [30] and Desaulniers et al. [50], infeasible, as the associated time windows required vehicles to depart from the depot much earlier than possible when starting with a low initial state of charge. However, future work could adapt these benchmarks to accommodate initial charging, enabling more standardized comparisons.

Further Development of the Metaheuristic Solution Approach

The ALNS metaheuristic applied in this study remains relatively untuned and rudimentary, offering limited mechanisms for refining charging and timing decisions. As a result, the approach exhibited substantial performance gaps and struggled to find optimal solutions even for small problem instances. Future work could likely improve the performance of the current algorithm by conducting systematic parameter tuning, using techniques such as grid search or learning-based methods like those proposed by Calvet et al. [129], to identify more effective configurations. The current implementation of the operators is also not yet fully optimized for computational efficiency. Performance could be improved further by reducing the time complexity of the operators, for example, by caching and incrementally updating components of the generalized cost function rather than recalculating them across the entire solution.

Enhancements to the algorithm itself could also be explored. For instance, one could incorporate more advanced charger removal operators, introduce dedicated phases for adjusting charging decisions (as in Keskin and Çatay [31]), or integrate linear programming components to efficiently optimize charging and timing decisions (as in Keskin, Laporte, and Çatay [44]). These would likely help to reduce the optimality gap. A simpler variant, using operators restricted to modifying the routing decisions, while using a corridor-based generalized cost function, could also be explored. This would more closely resemble the VNS-based approach proposed by Jeong et al. [113]. Adhering to this approach might significantly reduce implementation overhead. The current repair operator is overly complex due to the need to manage numerous edge cases with respect to the second-stage decisions, which has made the implementation susceptible to errors.

Another promising avenue for future research involves investigating alternative metaheuristic approaches for solving the RMP beyond ALNS. In particular, population-based metaheuristics present significant potential due to their inherent support for multi-objective optimization, which can help to address the trade-offs between cost-efficiency and robustness more effectively than with the current generalized cost function. Additionally, population-based metaheuristics tend to be more amenable to parallelization, thereby improving scalability. Memetic algorithms, like HGS [118], that combine the exploratory power of population-based strategies with the intensification strengths of large neighborhood search heuristics, are especially compelling.

Generating Managerial Insights using the Framework

From a business perspective, future work could investigate how the CERT$\mathcal{E}$ROUTE framework may be used to generate actionable managerial insights for Shell and its clients. This could be achieved through case studies that analyze real-world operational data and explore various optimization scenarios. For instance, the framework could be used to assess the impact of different fleet configurations (e.g., battery capacity, average energy consumption, supported charging speeds) and depot infrastructure setups (e.g., number and type of chargers) on the total expected operational cost under specific demand patterns.

Such an application would also benefit from a more comprehensive sensitivity analysis than the one conducted in this study to better capture the nuanced effects of key input parameters. Our current analysis, limited to small-scale instances, occasionally produced counterintuitive results. Notably, the Sobol indices failed to accurately reflect the true influence of some parameters on operational costs. For example, parameters like battery capacity and charging speed yielded Sobol indices close to zero, implying negligible impact, despite the fact that varying these parameters led to significant absolute changes in cost.

"Essentially, all models are wrong, but some are useful."
— George E. P. Box [221]

References

[1] Hannah Ritchie. “Cars, planes, trains: where do CO emissions from transport come from?” In: *Our World in Data* (2020). URL: https://ourworldindata.org/co2-emissions-from-transport.

[2] *Race to Zero European heavy-duty vehicle market development quarterly (January–December 2023).* [Accessed 24-04-2025]. 2024. URL: https://theicct.org/publication/race-to-zero-eu-hdv-market-development-q4-2023-mar24/.

[3] Statista. *Electric commercial vehicle fleet in the Netherlands 2018-2023, by vehicle type.* [Accessed 13-11-2024]. URL: https://www.statista.com/statistics/940793/number-of-electric-commercial-vehicles-in-the-netherlands-by-vehicle-type/.

[4] Samuel Pelletier et al. “The electric vehicle routing problem with energy consumption uncertainty”. In: *Transportation Research Part B: Methodological* 126 (2019), pp. 225–255. DOI: https://doi.org/10.1016/j.trb.2019.06.006.

[5] Alex Donkers et al. “Influence of driving style, infrastructure, weather and traffic on electric vehicle performance”. In: *Transportation Research Part D: Transport and Environment* 88 (2020), p. 102569. DOI: https://doi.org/10.1016/j.trd.2020.102569.

[6] Johannes Asamer et al. “Sensitivity analysis for energy demand estimation of electric vehicles”. In: *Transportation Research Part D: Transport and Environment* 46 (2016), pp. 182–199. DOI: https://doi.org/10.1016/j.trd.2016.03.017.

[7] Rafael Basso et al. “Electric vehicle routing problem with machine learning for energy prediction”. In: *Transportation Research Part B: Methodological* 145 (2021), pp. 24–55. DOI: https://doi.org/10.1016/j.trb.2020.12.007.

[8] Gonzalo Lera-Romero et al. “A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows”. In: *European Journal of Operational Research* 312.3 (2024), pp. 978–995. DOI: https://doi.org/10.1016/j.ejor.2023.06.037.

[9] Enrico Furnari et al. *Why most etrucks will choose overnight charging.* Oct. 2020. URL: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/why-most-etrucks-will-choose-overnight-charging.

[10] G. B. Dantzig et al. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91. DOI: 10.1287/mnsc.6.1.80.

[11] G. Clarke et al. “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”. In: *Operations Research* 12.4 (Aug. 1964), pp. 568–581. DOI: 10.1287/opre.12.4.568.

[12] Nicos Christofides. “The vehicle routing problem”. In: *Combinatorial optimization* (1979).

[13] J. K. Lenstra et al. *Complexity Of Vehicle Routing And Scheduling Problems.* Econometric Institute Archives 272191. Erasmus University Rotterdam, 1979. DOI: 10.22004/ag.econ.272191.

[14] Grigorios D. Konstantakopoulos et al. “Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification”. In: *Operational Research* (July 2022), pp. 1–30. DOI: 10.1007/s12351-020-00600-7.

[15] Çağrı Koç et al. “Thirty years of heterogeneous vehicle routing”. In: *European Journal of Operational Research* 249.1 (2016), pp. 1–21. DOI: https://doi.org/10.1016/j.ejor.2015.07.020.

[16] Xiaobo Liu et al. “A Systematic Literature Review of Vehicle Routing Problems with Time Windows”. In: *Sustainability* 15.15 (2023). DOI: 10.3390/su151512004. URL: https://www.mdpi.com/2071-1050/15/15/12004.

[17] Jairo R. Montoya-Torres et al. “A literature review on the vehicle routing problem with multiple depots”. In: *Computers & Industrial Engineering* 79 (2015), pp. 115–129. DOI: https://doi.org/10.1016/j.cie.2014.10.029.

[18] Çağrı Koç et al. “Vehicle routing with backhauls: Review and research perspectives”. In: *Computers Operations Research* 91 (2018), pp. 79–91. DOI: https://doi.org/10.1016/j.cor.2017.11.003.

[19] Victor Pillac et al. “A review of dynamic vehicle routing problems”. In: *European Journal of Operational Research* 225.1 (2013), pp. 1–11. DOI: https://doi.org/10.1016/j.ejor.2012.08.015.

[20] Rahma Lahyani et al. “Rich vehicle routing problems: From a taxonomy to a definition”. In: *European Journal of Operational Research* 241.1 (2015), pp. 1–14. DOI: https://doi.org/10.1016/j.ejor.2014.07.048.

[21] Kris Braekers et al. “The vehicle routing problem: State of the art classification and review”. In: *Computers & Industrial Engineering* 99 (2016), pp. 300–313. DOI: https://doi.org/10.1016/j.cie.2015.12.007.

[22] Sevgi Erdoğan et al. “A Green Vehicle Routing Problem”. In: *Transportation Research Part E: Logistics and Transportation Review* 48.1 (2012), pp. 100–114. DOI: https://doi.org/10.1016/j.tre.2011.08.001.

[23] Angel Alejandro Juan et al. “Electric Vehicles in Logistics and Transportation: A Survey on Emerging Environmental, Strategic, and Operational Challenges”. In: *Energies* 9.2 (2016). DOI: 10.3390/en9020086.

[24] Dimitris Margaritis et al. “Electric commercial vehicles: Practical perspectives and future research directions”. In: *Research in Transportation Business & Management* 18 (2016), pp. 4–10. DOI: https://doi.org/10.1016/j.rtbm.2016.01.005.

[25] Samuel Pelletier et al. “50th Anniversary Invited Article—Goods Distribution with Electric Vehicles: Review and Research Perspectives”. In: *Transportation Science* 50.1 (2016), pp. 3–22. DOI: 10.1287/trsc.2015.0646.

[26] Ryan G Conrad et al. “The recharging vehicle routing problem”. In: *Proceedings of the 2011 industrial engineering research conference*. Vol. 8. IISE Norcross, GA. 2011.

[27] Ilker Kucukoglu et al. “The electric vehicle routing problem and its variations: A literature review”. In: *Computers Industrial Engineering* 161 (2021), p. 107650. DOI: https://doi.org/10.1016/j.cie.2021.107650.

[28] Tomislav Erdelić et al. “A Survey on the Electric Vehicle Routing Problem: Variants and Solution Approaches”. In: *Journal of Advanced Transportation* 2019 (May 2019), pp. 1–48. DOI: 10.1155/2019/5075671.

[29] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

[30] Michael Schneider et al. “The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations”. In: *Transportation Science* 48 (Mar. 2014), pp. 500–520. DOI: 10.1287/trsc.2013.0490.

[31] Merve Keskin et al. “Partial recharge strategies for the electric vehicle routing problem with time windows”. In: *Transportation Research Part C: Emerging Technologies* 65 (2016), pp. 111–127. DOI: https://doi.org/10.1016/j.trc.2016.01.013.

[32] Maximilian Schiffer et al. “The electric location routing problem with time windows and partial recharging”. In: *European Journal of Operational Research* 260.3 (2017), pp. 995–1013. DOI: https://doi.org/10.1016/j.ejor.2017.01.011.

[33] Clair E Miller et al. “Integer programming formulation of traveling salesman problems”. In: *Journal of the ACM (JACM)* 7.4 (1960), pp. 326–329. DOI: 10.1145/321043.321046.

[34] Pedro Munari et al. *A generalized formulation for vehicle routing problems*. Sept. 2017. DOI: 10.48550/arXiv.1606.01935.

[35] W. W. Garvin et al. “Applications of Linear Programming in the Oil Industry”. In: *Management Science* 3.4 (July 1957), pp. 407–430. DOI: 10.1287/mnsc.3.4.407.

[36] Paolo Toth et al. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898718515.

[37] M. L. Balinski et al. “On an Integer Program for a Delivery Problem”. In: *Operations Research* 12.2 (Apr. 1964), pp. 300–304. DOI: 10.1287/opre.12.2.300.

[38] R. Baldacci et al. “An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation”. In: *Operations Research* 52.5 (2004), pp. 723–738. DOI: 10.1287/opre.1040.0111.

[39] Martin Desrochers et al. “Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints”. In: *Operations Research Letters* 10.1 (1991), pp. 27–36. DOI: https://doi.org/10.1016/0167-6377(91)90083-2.

[40] Francesca Rossi et al. “Chapter 4 Constraint Programming”. In: *Handbook of Knowledge Representation*. Ed. by Frank van Harmelen et al. Vol. 3. Foundations of Artificial Intelligence. 2008, pp. 181–211. DOI: https://doi.org/10.1016/S1574-6526(07)03004-0.

[41] Paul Shaw. “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems”. In: *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*. CP ’98. Springer-Verlag, 1998, pp. 417–431.

[42] Bruno Backer et al. “Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristics”. In: *J. Heuristics* 6 (Sept. 2000), pp. 501–523. DOI: 10.1023/A:1009621410177.

[43] Kyle Booth et al. “A Constraint Programming Approach to Electric Vehicle Routing with Time Windows”. In: Apr. 2019, pp. 129–145. DOI: 10.1007/978-3-030-19212-9_9.

[44] Merve Keskin et al. “Electric Vehicle Routing Problem with Time-Dependent Waiting Times at Recharging Stations”. In: *Computers & Operations Research* 107 (2019), pp. 77–94. DOI: https://doi.org/10.1016/j.cor.2019.02.014.

[45] Brian Kallehauge et al. “Vehicle Routing Problem with Time Windows”. In: *Column Generation*. Ed. by Guy Desaulniers et al. Boston, MA: Springer US, 2005, pp. 67–98. DOI: 10.1007/0-387-25486-2_3.

[46] Jun Jiang et al. “Vehicle routing problem with a heterogeneous fleet and time windows”. In: *Expert Systems with Applications* 41.8 (2014), pp. 3748–3760. DOI: https://doi.org/10.1016/j.eswa.2013.11.029.

[47] Gerhard Hiermann et al. “The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations”. In: *European Journal of Operational Research* 252.3 (2016), pp. 995–1018. DOI: https://doi.org/10.1016/j.ejor.2016.01.038.

[48] Dominik Goeke et al. “Routing a mixed fleet of electric and conventional vehicles”. In: *European Journal of Operational Research* 245.1 (2015), pp. 81–99. DOI: https://doi.org/10.1016/j.ejor.2015.01.049.

[49] Maurizio Bruglieri et al. “A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows”. In: *Electronic Notes in Discrete Mathematics* 47 (2015), pp. 221–228. DOI: https://doi.org/10.1016/j.endm.2014.11.029.

[50] Guy Desaulniers et al. “Exact Algorithms for Electric Vehicle-Routing Problems with Time Windows”. In: *Operations Research* 64.6 (Dec. 2016), pp. 1388–1405. DOI: 10.1287/opre.2016.1535.

[51] Rafael Basso et al. “Energy consumption estimation integrated into the Electric Vehicle Routing Problem”. In: *Transportation Research Part D: Transport and Environment* 69 (2019), pp. 141–167. DOI: https://doi.org/10.1016/j.trd.2019.01.006.

[52] İlker Küçükoğlu et al. “Hybrid simulated annealing and tabu search method for the electric travelling salesman problem with time windows and mixed charging rates”. In: *Expert Systems with Applications* 134 (2019), pp. 279–303. DOI: https://doi.org/10.1016/j.eswa.2019.05.037.

[53] Alberto Ceselli et al. "A Branch-and-Cut-and-Price Algorithm for the Electric Vehicle Routing Problem with Multiple Technologies". In: *SN Operations Research Forum* 2 (Jan. 2021). DOI: 10.1007/s43069-020-00052-x.

[54] Samuel Pelletier et al. "Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models". In: *Transportation Research Part B: Methodological* 103 (2017), pp. 158–187. DOI: https://doi.org/10.1016/j.trb.2017.01.020.

[55] Alejandro Montoya et al. "The electric vehicle routing problem with nonlinear charging function". In: *Transportation Research Part B: Methodological* 103 (2017), pp. 87–110. DOI: https://doi.org/10.1016/j.trb.2017.02.004.

[56] Aurélien Froger et al. "Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions". In: *Computers Operations Research* 104 (2019), pp. 256–294. DOI: https://doi.org/10.1016/j.cor.2018.12.013.

[57] Bingjie Li et al. "An Overview and Experimental Study of Learning-Based Optimization Algorithms for the Vehicle Routing Problem". In: *IEEE/CAA Journal of Automatica Sinica* 9 (July 2022), pp. 1115–1138. DOI: 10.1109/JAS.2022.105677.

[58] Paola Festa. "A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems". In: July 2014, pp. 1–20. DOI: 10.1109/ICTON.2014.6876285.

[59] George B. Dantzig. "Origins of the simplex method". In: *A History of Scientific Computing*. New York, NY, USA: Association for Computing Machinery, 1990, pp. 141–151. DOI: https://doi.org/10.1145/87252.88081.

[60] DB Yudin et al. "Informational complexity and efficient methods for the solution of convex extremal problems (in russian). Ekonomika i Matematicheskie metody, 12: 357–369, 1976". In: *English translation: Matekon* 13.2 (1976), pp. 3–25.

[61] Ralph Gomory. "Outline of an Algorithm for Integer Solutions to Linear Programs". In: *Bulletin of the American Mathematical Society* 64 (Sept. 1958), pp. 275–278. DOI: 10.1090/S0002-9904-1958-10224-4.

[62] A. H. Land et al. "An Automatic Method of Solving Discrete Programming Problems". In: *Econometrica* 28.3 (1960), pp. 497–520. URL: http://www.jstor.org/stable/1910129.

[63] Manfred Padberg et al. "A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems". In: *SIAM Review* 33.1 (1991), pp. 60–100. DOI: 10.1137/1033004.

[64] George B. Dantzig et al. "Decomposition Principle for Linear Programs". In: *Operations Research* 8.1 (1960), pp. 101–111. DOI: 10.1287/opre.8.1.101.

[65] Martin Desrochers et al. "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows". In: *Operations Research* 40.2 (1992), pp. 342–354. DOI: 10.1287/opre.40.2.342.

[66] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. [Accessed 19-06-2025]. 2025. URL: https://www.gurobi.com.

[67] IBM ILOG Cplex. "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53 (2009), p. 157.

[68] Mehmet Soysal et al. "Pickup and delivery with electric vehicles under stochastic battery depletion". In: *Computers & Industrial Engineering* 146 (2020), p. 106512. DOI: https://doi.org/10.1016/j.cie.2020.106512.

[69] Ruiting Wang et al. "Robust routing for a mixed fleet of heavy-duty trucks with pickup and delivery under energy consumption uncertainty". In: *Applied Energy* 368 (2024), p. 123407. DOI: https://doi.org/10.1016/j.apenergy.2024.123407.

[70] Herbert Kopfer et al. "Energy vehicle routing problem for differently sized and powered vehicles". In: *Journal of Business Economics* 89.7 (Sept. 2019), pp. 793–821. DOI: 10.1007/s11573-018-0910-z.

[71] Hesamoddin Tahami et al. “Exact approaches for routing capacitated electric vehicles”. In: *Transportation Research Part E: Logistics and Transportation Review* 144 (2020), p. 102126. DOI: https://doi.org/10.1016/j.tre.2020.102126.

[72] Chungmok Lee. “An exact algorithm for the electric-vehicle routing problem with nonlinear charging time”. In: *Journal of the Operational Research Society* 72.7 (2021), pp. 1461–1485. DOI: 10.1080/01605682.2020.1730250.

[73] Afsane Amiri et al. “A robust multi-objective routing problem for heavy-duty electric trucks with uncertain energy consumption”. In: *Computers & Industrial Engineering* 178 (2023), p. 109108. DOI: https://doi.org/10.1016/j.cie.2023.109108.

[74] Jean-François Cordeau et al. “New Heuristics for the Vehicle Routing Problem”. In: Dec. 2005, pp. 279–297. DOI: 10.1007/0-387-24977-X_9.

[75] Gilbert Laporte et al. “Chapter 4: Heuristics for the Vehicle Routing Problem”. In: Nov. 2014, pp. 87–116. DOI: 10.1137/1.9781611973594.ch4.

[76] Billy E. Gillett et al. “A Heuristic Algorithm for the Vehicle-Dispatch Problem”. In: *Operations Research* 22.2 (1974), pp. 340–349. DOI: 10.1287/opre.22.2.340.

[77] David L. Applegate et al. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006. URL: http://www.jstor.org/stable/j.ctt7s8xg.

[78] B. Widrow et al. “A comparison of adaptive algorithms based on the methods of steepest descent and random search”. In: *IEEE Transactions on Antennas and Propagation* 24.5 (1976), pp. 615–637. DOI: 10.1109/TAP.1976.1141414.

[79] Christian Blum et al. “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”. In: *ACM Comput. Surv.* 35 (Jan. 2001), pp. 268–308. DOI: 10.1145/937503.937505.

[80] Martin W. P. Savelsbergh. “The Vehicle Routing Problem with Time Windows: Minimizing Route Duration”. In: *ORSA Journal on Computing* 4.2 (1992), pp. 146–154. DOI: 10.1287/ijoc.4.2.146.

[81] Shen Lin. “Computer solutions of the traveling salesman problem”. In: *The Bell System Technical Journal* 44.10 (1965), pp. 2245–2269. DOI: 10.1002/j.1538-7305.1965.tb04146.x.

[82] Jean-Yves Potvin et al. “An Exchange Heuristic for Routeing Problems with Time Windows”. In: *Journal of the Operational Research Society* 46.12 (1995), pp. 1433–1446. DOI: 10.1057/jors.1995.204.

[83] Paolo Toth et al. “The Granular Tabu Search and Its Application to the Vehicle-Routing Problem”. In: *INFORMS Journal on Computing* 15.4 (2003), pp. 333–346. DOI: 10.1287/ijoc.15.4.333.24890.

[84] David Pisinger et al. “Large Neighborhood Search”. In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Cham: Springer International Publishing, 2019, pp. 99–127. DOI: 10.1007/978-3-319-91086-4_4.

[85] Fred Glover. “Future paths for integer programming and links to artificial intelligence”. In: *Computers & Operations Research* 13.5 (1986), pp. 533–549. DOI: https://doi.org/10.1016/0305-0548(86)90048-1.

[86] Hadeer Awad et al. “A Taxonomic Review of Metaheuristic Algorithms for Solving the Vehicle Routing Problem and Its Variants”. In: *Computers Industrial Engineering* (Dec. 2019), p. 106242. DOI: 10.1016/j.cie.2019.106242.

[87] N. Mladenović et al. “Variable neighborhood search”. In: *Computers & Operations Research* 24.11 (1997), pp. 1097–1100. DOI: https://doi.org/10.1016/S0305-0548(97)00031-2.

[88] Pierre Hansen et al. “Variable Neighborhood Search”. In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Cham: Springer International Publishing, 2019, pp. 57–97. DOI: 10.1007/978-3-319-91086-4_3.

[89] Paul Shaw. "A new local search algorithm providing high quality solutions to vehicle routing problems". In: *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK* 46 (1997).

[90] Stefan Ropke et al. "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows". In: *Transportation science* 40.4 (2006), pp. 455–472.

[91] Fred Glover. "Tabu search—part I". In: *ORSA Journal on computing* 1.3 (1989), pp. 190–206.

[92] Michel Gendreau et al. ""A Tabu Search Heuristic for the Vehicle Routing Problem"". In: *Mgmt Sci* 40 (Oct. 1994), pp. 1276–1290. DOI: 10.1287/mnsc.40.10.1276.

[93] Michel Gendreau et al. "Tabu Search". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Cham: Springer International Publishing, 2019, pp. 37–55. DOI: 10.1007/978-3-319-91086-4_2.

[94] Scott Kirkpatrick et al. "Optimization by simulated annealing". In: *science* 220.4598 (1983), pp. 671–680.

[95] Ibrahim Hassan Osman. "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem". In: *Annals of Operations Research* 41.4 (Dec. 1993), pp. 421–451. DOI: 10.1007/BF02023004.

[96] Daniel Delahaye et al. "Simulated Annealing: From Basics to Applications". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Cham: Springer International Publishing, 2019, pp. 1–35. DOI: 10.1007/978-3-319-91086-4_1.

[97] Charles Darwin. "Origin of the Species". In: *British Politics and the environment in the long nineteenth century*. Routledge, 2023, pp. 47–55.

[98] Thomas Bartz-Beielstein et al. "Evolutionary Algorithms". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4 (May 2014). DOI: 10.1002/widm.1124.

[99] Xinjie Yu et al. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[100] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Apr. 1992. DOI: 10.7551/mitpress/1090.001.0001.

[101] Alex Van Breedam. "An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem". In: (Jan. 1996).

[102] D.E. Goldberg. *Genetic Algorithms*. Pearson Education India, 2013. URL: https://books.google.nl/books?id=6gzS07Sv9hoC.

[103] M. Dorigo et al. "Ant system: optimization by a colony of cooperating agents". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), pp. 29–41. DOI: 10.1109/3477.484436.

[104] B. Bullnheimer et al. "An improved Ant System algorithm for the Vehicle Routing Problem". In: *Annals of Operations Research* 89.0 (Jan. 1999), pp. 319–328. DOI: 10.1023/A:1018940026670.

[105] Shuai Zhang et al. "Electric vehicle routing problem with recharging stations for minimizing energy consumption". In: *International Journal of Production Economics* 203 (2018), pp. 404–413. DOI: https://doi.org/10.1016/j.ijpe.2018.07.016.

[106] Marco Dorigo et al. "Ant Colony Optimization: Overview and Recent Advances". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Cham: Springer International Publishing, 2019, pp. 311–351. DOI: 10.1007/978-3-319-91086-4_10.

[107] Pablo Moscato et al. "A Modern Introduction to Memetic Algorithms". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau et al. Boston, MA: Springer US, 2010, pp. 141–183. DOI: 10.1007/978-1-4419-1665-5_6.

[108] Thibaut Vidal et al. "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems". In: *Operations Research* 60.3 (2012), pp. 611–624.

[109] Thibaut Vidal et al. “A unified solution framework for multi-attribute vehicle routing problems”. In: *European Journal of Operational Research* 234.3 (2014), pp. 658–673. DOI: 10.1016/j.ejor.2013.09.045.

[110] Emrah Demir et al. “An adaptive large neighborhood search heuristic for the Pollution-Routing Problem”. In: *European Journal of Operational Research* 223.2 (2012), pp. 346–359. DOI: https://doi.org/10.1016/j.ejor.2012.06.044.

[111] Merve Keskin et al. “A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations”. In: *Computers Operations Research* 125 (2021), p. 105060. DOI: https://doi.org/10.1016/j.cor.2020.105060.

[112] Dan Wang et al. “Robust Optimization for Electric Vehicle Routing Problem Considering Time Windows Under Energy Consumption Uncertainty”. In: *Applied Sciences* 15.2 (2025). DOI: 10.3390/app15020761.

[113] Jaehee Jeong et al. “Adaptive robust electric vehicle routing under energy consumption uncertainty”. In: *Transportation Research Part C: Emerging Technologies* 160 (2024), p. 104529. DOI: https://doi.org/10.1016/j.trc.2024.104529.

[114] Ángel Felipe et al. “A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges”. In: *Transportation Research Part E: Logistics and Transportation Review* 71 (2014), pp. 111–128. DOI: https://doi.org/10.1016/j.tre.2014.09.003.

[115] Huiting Mao et al. “The Electric Vehicle Routing Problem With Time Windows and Multiple Recharging Options”. In: *IEEE Access* 8 (2020), pp. 114864–114875. DOI: 10.1109/ACCESS.2020.3003000.

[116] Jianhua Xiao et al. “A diversity-enhanced memetic algorithm for solving electric vehicle routing problems with time windows and mixed backhauls”. In: *Applied Soft Computing* 134 (2023), p. 110025. DOI: https://doi.org/10.1016/j.asoc.2023.110025.

[117] Jinting Dong et al. “Dynamic electric vehicle routing problem considering mid-route recharging and new demand arrival using an improved memetic algorithm”. In: *Sustainable Energy Technologies and Assessments* 58 (2023), p. 103366. DOI: https://doi.org/10.1016/j.seta.2023.103366.

[118] Gerhard Hiermann et al. “Routing a mix of conventional, plug-in hybrid, and electric vehicles”. In: *European Journal of Operational Research* 272.1 (2019), pp. 235–248. DOI: https://doi.org/10.1016/j.ejor.2018.06.025.

[119] Thibaut Vidal et al. “Arc routing with time-dependent travel times and paths”. In: *Transportation Science* 55.3 (2021), pp. 706–724.

[120] Ruibin Bai et al. “Analytics and machine learning in vehicle routing research”. In: *International Journal of Production Research* 61 (Dec. 2021), pp. 1–27. DOI: 10.1080/00207543.2021.2013566.

[121] Aigerim Bogyrbayeva et al. “Machine Learning to Solve Vehicle Routing Problems: A Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* PP (June 2024), pp. 1–19. DOI: 10.1109/TITS.2023.3334976.

[122] JE Beasley. “Route first—Cluster second methods for vehicle routing”. In: *Omega* 11.4 (1983), pp. 403–408. DOI: https://doi.org/10.1016/0305-0483(83)90033-6.

[123] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[124] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[125] Marcelo Prates et al. “Learning to Solve NP-Complete Problems: A Graph Neural Network for Decision TSP”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 4731–4738. DOI: 10.1609/aaai.v33i01.33014731. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4399.

[126] Anton Milan et al. "Data-driven approximations to NP-hard problems". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 1453–1459.

[127] Richard S Sutton. "Reinforcement learning: An introduction". In: *A Bradford Book* (2018).

[128] Rafael Basso et al. "Dynamic stochastic electric vehicle routing with safe reinforcement learning". In: *Transportation Research Part E: Logistics and Transportation Review* 157 (2022), p. 102496. DOI: https://doi.org/10.1016/j.tre.2021.102496.

[129] Laura Calvet et al. "Combining statistical learning with metaheuristics for the Multi-Depot Vehicle Routing Problem with market segmentation". In: *Computers Industrial Engineering* 94 (2016), pp. 93–104. DOI: https://doi.org/10.1016/j.cie.2016.01.016.

[130] Angel A Juan et al. "Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem". In: *International Transactions in Operational Research* 22.4 (2015), pp. 647–667.

[131] Maria Amélia Lopes Silva et al. "A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems". In: *Expert Systems with Applications* 131 (2019), pp. 148–171. DOI: https://doi.org/10.1016/j.eswa.2019.04.056.

[132] Ali Nadi et al. "Data-driven preference-based routing and scheduling for activity-based freight transport modelling". In: *Transportation Research Part C: Emerging Technologies* 158 (2024), p. 104413. DOI: https://doi.org/10.1016/j.trc.2023.104413.

[133] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[134] Matthew Barth et al. "Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model". In: *Institute of Transportation Studies, UC Berkeley, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings* (Jan. 2005).

[135] Emrah Demir et al. "A comparative analysis of several vehicle emission models for road freight transportation". In: *Transportation Research Part D: Transport and Environment* 16.5 (2011), pp. 347–357. DOI: https://doi.org/10.1016/j.trd.2011.01.011.

[136] Felipe Rodríguez Oscar Delgado et al. *Fuel Efficiency Technologies for European Heavy-Duty Vehicles (2020-2030)*. Tech. rep. United States of America, July 2017. URL: https://theicct.org/wp-content/uploads/2021/06/EU-HDV-Tech-Potential_ICCT-white-paper_14072017_vF.pdf.

[137] Editor Engineeringtoolbox. *Air - density, specific weight and thermal expansion coefficient vs. temperature and pressure*. June 2024. URL: https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html.

[138] Giulia Sandrini et al. "Regenerative Braking Logic That Maximizes Energy Recovery Ensuring the Vehicle Stability". In: *Energies* 15 (Aug. 2022), p. 5846. DOI: 10.3390/en15165846.

[139] *CO2 Emissions from Trucks in the European Union: Analysis of the 2021 Reporting Period*. Tech. rep. United States of America, Dec. 2024. URL: https://policycommons.net/artifacts/17952569/working-paper-co-emissions-from-trucks-in-the-european-union/.

[140] Gebeyehu M. Fetene et al. "Harnessing big data for estimating the energy consumption and driving range of electric vehicles". In: *Transportation Research Part D: Transport and Environment* 54 (2017), pp. 1–11. DOI: https://doi.org/10.1016/j.trd.2017.04.013.

[141] *HERE Technologies*. [Accessed 16-01-2025]. 2022. URL: https://www.here.com/docs/.

[142] Thomas Hennig et al. "The Shuttle Radar Topography Mission". In: Jan. 2001, pp. 65–77. DOI: 10.1007/3-540-44818-7_11.

[143] *European Environment Agency, 2014. Digital elevation model over europe (eu-dem)*. [Accessed 16-01-2025]. 2014. URL: http://www.eea.europa.eu/data-and-maps/data/eu-dem.

[144] Pablo Alvarez Lopez et al. “Microscopic Traffic Simulation using SUMO”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[145] *AMBER*. [Accessed 24-04-2025]. 2022. URL: https://vms.taps.anl.gov/tools/amber/.

[146] Patrick Keyantuo et al. “Distributionally Robust and Data-Driven Solutions to Commercial Vehicle Routing Problems”. In: *IFAC-PapersOnLine* 56.2 (2023), pp. 10497–10502. DOI: https://doi.org/10.1016/j.ifacol.2023.10.1069.

[147] Soomin Woo et al. “Saving energy with eco-friendly routing of an electric vehicle fleet”. In: *Transportation Research Part E: Logistics and Transportation Review* 189 (2024), p. 103644. DOI: https://doi.org/10.1016/j.tre.2024.103644.

[148] *WideSense*. [Accessed 24-04-2025]. URL: https://www.widesense.net/.

[149] John R Birge et al. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[150] Daniel P Heyman et al. *Stochastic models in operations research: stochastic optimization*. Vol. 2. Courier Corporation, 2004.

[151] Dimitris Bertsimas et al. “Theory and Applications of Robust Optimization”. In: *SIAM Review* 53.3 (2011), pp. 464–501. DOI: 10.1137/080734510.

[152] Virginie Gabrel et al. “Recent advances in robust optimization: An overview”. In: *European Journal of Operational Research* 235.3 (2014), pp. 471–483. DOI: https://doi.org/10.1016/j.ejor.2013.09.036.

[153] Peter Kall et al. *Stochastic linear programming*. Springer, 1976.

[154] Dimitris Bertsimas et al. “Adaptive robust optimization for the security constrained unit commitment problem”. In: *IEEE transactions on power systems* 28.1 (2012), pp. 52–63.

[155] Bo Zeng et al. “Solving two-stage robust optimization problems using a column-and-constraint generation method”. In: *Operations Research Letters* 41.5 (2013), pp. 457–461. DOI: https://doi.org/10.1016/j.orl.2013.05.003.

[156] Olga Kuryatnikova. *Robust linear optimization: intro, constructing uncertainty sets, adversarial approach, polyhedral uncertainty*. Dec. 2024.

[157] Chao Shang et al. “Data-driven robust optimization based on kernel learning”. In: *Computers Chemical Engineering* 106 (2017), pp. 464–479. DOI: https://doi.org/10.1016/j.compchemeng.2017.07.004.

[158] Yun Li et al. “Machine learning enabled uncertainty set for data-driven robust optimization”. In: *Journal of Process Control* 144 (2024), p. 103339. DOI: https://doi.org/10.1016/j.jprocont.2024.103339.

[159] Chao Ning et al. “Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods”. In: *Computers Chemical Engineering* 112 (2018), pp. 190–210. DOI: https://doi.org/10.1016/j.compchemeng.2018.02.007.

[160] Marc Goerigk et al. “Data-driven robust optimization using deep neural networks”. In: *Computers Operations Research* 151 (2023), p. 106087. DOI: https://doi.org/10.1016/j.cor.2022.106087.

[161] Douglas A Reynolds et al. “Gaussian mixture models.” In: *Encyclopedia of biometrics* 741.659-663 (2009), p. 3.

[162] Aharon Ben-Tal et al. “Adjustable robust solutions of uncertain linear programs”. In: *Mathematical programming* 99.2 (2004), pp. 351–376.

[163] Michel Gendreau et al. “50th anniversary invited article—future research directions in stochastic vehicle routing”. In: *Transportation Science* 50.4 (2016), pp. 1163–1173.

[164] Jorge Oyola et al. “The stochastic vehicle routing problem, a literature review, part I: models”. In: *EURO Journal on Transportation and Logistics* 7.3 (2018), pp. 193–221.

[165] Gilbert Laporte et al. “The Vehicle Routing Problem with Stochastic Travel Times”. In: *Transportation Science* 26.3 (Aug. 1992), pp. 161–170. DOI: 10.1287/trsc.26.3.161.

[166] Ilgaz Sungur et al. “A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty”. In: *Iie Transactions* 40.5 (2008), pp. 509–523.

[167] Chrysanthos E Gounaris et al. “The robust capacitated vehicle routing problem under demand uncertainty”. In: *Operations research* 61.3 (2013), pp. 677–693.

[168] C. Hu et al. “Robust vehicle routing problem with hard time windows under demand and travel time uncertainty”. In: *Computers Operations Research* 94 (2018), pp. 139–153. DOI: https://doi.org/10.1016/j.cor.2018.02.006.

[169] Maaike Hoogeboom et al. “The robust vehicle routing problem with time window assignments”. In: *Transportation Science* 55.2 (2021), pp. 395–413.

[170] Patrick Jaillet et al. “Routing optimization under uncertainty”. In: *Operations research* 64.1 (2016), pp. 186–200.

[171] Chao Lei et al. “A two-stage robust optimization approach for the mobile facility fleet sizing and routing problem under uncertainty”. In: *Computers Operations Research* 67 (2016), pp. 75–89. DOI: https://doi.org/10.1016/j.cor.2015.09.007.

[172] Maximilian Schiffer et al. “Strategic planning of electric logistics fleet networks: A robust location-routing approach”. In: *Omega* 80 (2018), pp. 31–42. DOI: https://doi.org/10.1016/j.omega.2017.09.003.

[173] Shuai Zhang et al. “A novel location-routing problem in electric vehicle transportation with stochastic demands”. In: *Journal of Cleaner Production* 221 (2019), pp. 567–581. DOI: https://doi.org/10.1016/j.jclepro.2019.02.167.

[174] Joonrak Kim et al. “Robust optimization model for the electric vehicle routing problem under battery energy consumption uncertainty with arc segmentation”. In: *International Journal of Sustainable Transportation* 17.5 (2023), pp. 434–445. DOI: https://doi.org/10.1080/15568318.2022.2049403.

[175] Hiroshi Konno. “A cutting plane algorithm for solving bilinear programs”. In: *Mathematical Programming* 11.1 (1976), pp. 14–27. DOI: 10.1007/BF01580367.

[176] Nikolaos Damianakis et al. “Impact of Uncertainties and Price of Robustness in receding-horizon EV Smart-Charging”. In: *2024 IEEE 21st International Power Electronics and Motion Control Conference (PEMC)*. 2024, pp. 1–6. DOI: 10.1109/PEMC61721.2024.10726368.

[177] İhsan Yanıkoğlu et al. “A Survey of Adjustable Robust Optimization - (Cite as Yanıkoğlu et al. (2019). EJOR 277(3), 799-813)”. In: *European Journal of Operational Research* 277 (Sept. 2019), pp. 799–813. DOI: 10.1016/j.ejor.2018.08.031.

[178] Ali Saklaoui et al. “Coordinated Optimization of Logistics Electric Fleet and Energy Management System of Constrained Energy Hub”. In: *IEEE Access* 12 (2024), pp. 152466–152481.

[179] *Heavy goods vehicles.* [Accessed 2025-05-19]. URL: https://road-safety.transport.ec.europa.eu/eu-road-safety-policy/priorities/safe-vehicles/archive/safety-design-needs/heavy-goods-vehicles_en.

[180] Patrick S Klein et al. “Electric vehicle charge scheduling with flexible service operations”. In: *Transportation Science* 57.6 (2023), pp. 1605–1626.

[181] Pedro Lozano Ruiz et al. “Physics-based and Data-driven Modeling of Degradation Mechanisms for Lithium-Ion Batteries - A Review”. In: *IEEE Access* 13 (2025), pp. 21164–21189. DOI: 10.1109/ACCESS.2025.3535918.

[182] European Labour Authority. *Driving and Resting Time Rules.* [Accessed 16-01-2025]. 2022. URL: https://www.ela.europa.eu/sites/default/files/2022-08/Driving_and_Resting_Times_English.pdf.

[183] Marc Goerigk et al. "Robust combinatorial optimization with locally budgeted uncertainty". In: *Open Journal of Mathematical Optimization* 2 (2021), pp. 1–18.

[184] David L. Cortés-Murcia et al. "The electric vehicle routing problem with time windows, partial recharges and satellite customers". In: *Transportation Research Part E: Logistics and Transportation Review* 130 (2019), pp. 184–206. DOI: https://doi.org/10.1016/j.tre.2019.08.015.

[185] Álvaro Lorca et al. "Adaptive Robust Optimization With Dynamic Uncertainty Sets for Multi-Period Economic Dispatch Under Significant Wind". In: *IEEE Transactions on Power Systems* 30.4 (2015), pp. 1702–1713. DOI: 10.1109/TPWRS.2014.2357714.

[186] Paul Shaw. "Using constraint programming and local search methods to solve vehicle routing problems". In: *International conference on principles and practice of constraint programming.* Springer. 1998, pp. 417–431.

[187] Uğur Emeç et al. "An Adaptive Large Neighborhood Search for an E-grocery Delivery Routing Problem". In: *Computers Operations Research* 69 (2016), pp. 109–125. DOI: https://doi.org/10.1016/j.cor.2015.11.008.

[188] Jean-Yves Potvin et al. "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows". In: *European Journal of Operational Research* 66.3 (1993), pp. 331–340. DOI: https://doi.org/10.1016/0377-2217(93)90221-8.

[189] Niels A Wouda et al. "ALNS: A Python implementation of the adaptive large neighbourhood search metaheuristic". In: *Journal of Open Source Software* 8.81 (2023), p. 5028.

[190] Alberto Santini et al. "A comparison of acceptance criteria for the Adaptive Large Neighbourhood Search metaheuristic". In: *Journal of Heuristics* 24.5 (2018), pp. 783–815. DOI: 10.1007/s10732-018-9377-x.

[191] Gunter Dueck. "New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel". In: *Journal of Computational Physics* 104.1 (1993), pp. 86–92. DOI: https://doi.org/10.1006/jcph.1993.1010.

[192] Daniel Speth et al. "Synthetic European road freight transport flow data". In: *Data in Brief* 40 (2022), p. 107786. DOI: https://doi.org/10.1016/j.dib.2021.107786.

[193] Arcadis. *Synthese Zero-Emissie Verzorgingsplaatsen in Nederland.* [Accessed 24-05-2025]. Dec. 2023. URL: https://open.overheid.nl/documenten/dpc-463131590087cb23e2f9815a58c86f0f597f8a55/pdf.

[194] Wikipedia. *Lijst van verzorgingsplaatsen langs Nederlandse auto(snel)wegen.* [Accessed 24-05-2025]. 2025. URL: http://nl.wikipedia.org/w/index.php?title=Lijst%5C%20van%5C%20verzorgingsplaatsen%5C%20langs%5C%20Nederlandse%5C%20auto(snel)wegen&oldid=68701715.

[195] Eamonn Mulholland et al. *Race to Zero: European Heavy-Duty Vehicle Market Development Quarterly (January–December 2024).* Tech. rep. United States of America, Feb. 2025. URL: https://theicct.org/publication/r2z-eu-hdv-market-development-quarterly-jan-dec-2024-feb25/.

[196] Volvo Trucks. *Volvo Trucks vijfde jaar op rij marktleider elektrische vrachtwagens in Europa en Noord-Amerika.* [Accessed 23-04-2025]. Mar. 2025. URL: https://www.volvotrucks.nl/nl-nl/news/press-releases/2025/mar/volvo-trucks-vijfde-jaar-op-rij-marktleider-elektrische-vrachtwagens-in-europa-en-noord-amerika.html.

[197] Volvo Trucks. *Volvo FH Electric.* [Accessed 23-04-2025]. 2025. URL: https://www.volvotrucks.com/en-en/trucks/electric/volvo-fh-electric.html.

[198] Schmitz Cargobull. *Changes on the Schmitz Cargobull AG Supervisory Board.* [Accessed 23-04-2025]. 2024. URL: https://www.cargobull.com/se/presse/2024/2024-158-wechsel-im-ar.

[199] Schmitz Cargobull. *S.KOe COOL.* [Accessed 23-04-2025]. URL: https://www.cargobull.com/nl/producten/kofferfahrzeuge/sattelkoffer/skoe-cool.

[200] LVS Trucks. *Simon Loos neemt Volvo electric met e-PTO in gebruik.* [Accessed 23-04-2025]. URL: https://lvs.nl/nieuws/simon-loos-neemt-drie-assige-volvo-fh-electric-trekker-met-e-pto-in-gebruik.

[201] Jan Van Belle et al. "Cross-docking: State of the art". In: *Omega* 40.6 (2012), pp. 827–846. DOI: https://doi.org/10.1016/j.omega.2012.01.005.

[202] D. Liebig et al. "Charging Solutions for Battery Electric Trucks and Busses". In: *Proceedings of the 46th International Vienna Motor Symposium.* Shell Global Solutions (Germany) GmbH; SBRS GmbH. Vienna, Austria: ÖVK - Austrian Society of Automotive Engineers, 2025. DOI: 10.62626/wcr7-joex.

[203] Pascal Neis et al. "OpenRouteService.org is three times "Open": Combining OpenSource, OpenLS and OpenStreetMap". In: *GIS Research UK (GISRUK 08).* Manchester, 2008.

[204] Edsger Wybe Dijkstra. "A Note on Two Problems in Connexion with Graphs." In: *Numerische Mathematik* 1 (1959), pp. 269–271. URL: http://eudml.org/doc/131436.

[205] Mordechai Haklay et al. "Openstreetmap: User-generated street maps". In: *IEEE Pervasive computing* 7.4 (2008), pp. 12–18.

[206] *Travel Speeds.* [Accessed 24-04-2025]. URL: https://giscience.github.io/openrouteservice/technical-details/travel-speeds/.

[207] Kennisinstituut voor Mobiliteitsbeleid (KiM). *Kerncijfers Mobiliteit 2024.* [Accessed 23-04-2025]. 2024. URL: https://www.kimnet.nl/publicaties/publicaties/2024/11/18/kerncijfers-mobiliteit-2024.

[208] *Vehicle Energy Consumption calculation TOol - VECTO.* [Accessed 24-04-2025]. URL: https://climate.ec.europa.eu/eu-action/transport/road-transport-reducing-co2-emissions-vehicles/vehicle-energy-consumption-calculation-tool-vecto_en.

[209] Koninklijk Nederlands Meteorologisch Instituut (KNMI). *Daggegevens 2024.* [Accessed 23-04-2025]. 2025. URL: https://daggegevens.knmi.nl/klimatologie/daggegevens.

[210] Bruce J West et al. "The noise in natural phenomena". In: *American Scientist* 78.1 (1990), pp. 40–45.

[211] Paul E Johnson. "Where do Multivariate Normal Samples Come from?" In: *Center for Research Methods and Data Analysis* (2017).

[212] William E Hart et al. "Pyomo: modeling and solving mathematical programs in Python". In: *Mathematical Programming Computation* 3.3 (2011), pp. 219–260.

[213] I.M Sobol . "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates". In: *Mathematics and Computers in Simulation* 55.1 (2001). The Second IMACS Seminar on Monte Carlo Methods, pp. 271–280. DOI: https://doi.org/10.1016/S0378-4754(00)00270-6.

[214] Andrea Saltelli et al. *Sensitivity analysis in practice: a guide to assessing scientific models.* Vol. 1. Wiley Online Library, 2004.

[215] Jon Herman et al. "SALib: An open-source Python library for Sensitivity Analysis". In: *The Journal of Open Source Software* 2.9 (Jan. 2017). DOI: 10.21105/joss.00097.

[216] Francesca Campolongo et al. "From screening to quantitative sensitivity analysis. A unified approach". In: *Computer Physics Communications* 182.4 (2011), pp. 978–988. DOI: https://doi.org/10.1016/j.cpc.2010.12.039.

[217] Iep van der Meer. *Tweede generatie Volvo Electric.* [Accessed 19-06-2025]. URL: https://bigtruck.nl/nieuws/tweede-generatie-volvo-electric.

[218] Johannes Wiebe et al. "ROmodel: modeling robust optimization problems in Pyomo". In: *Optimization and Engineering* 23 (Dec. 2021), pp. 1–22. DOI: 10.1007/s11081-021-09703-2.

[219] Jason Sherman et al. "Recent Advances of PyROS: A Pyomo Solver for Nonconvex Two-Stage Robust Optimization in Process Systems Engineering". In: National Energy Technology Laboratory (NETL), Pittsburgh, PA, Morgantown, WV, and Albany, OR (United States). July 2024. DOI: 10.69997/sct.142058.

[220] Thibaut Vidal. "Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood". In: *Computers Operations Research* 140 (2022), p. 105643. DOI: https://doi.org/10.1016/j.cor.2021.105643.

[221] George E. P. Box and. "Science and Statistics". In: *Journal of the American Statistical Association* 71.356 (1976), pp. 791–799. DOI: 10.1080/01621459.1976.10480949.

[222] Anja Konzept et al. "Battery Electric Vehicle Efficiency Test for Various Velocities". In: *Vehicles* 4.1 (2022), pp. 60–73. DOI: 10.3390/vehicles4010004.

[223] Sokratis Mamarikas et al. "Traffic impacts on energy consumption of electric and conventional vehicles". In: *Transportation Research Part D: Transport and Environment* 105 (2022), p. 103231. DOI: https://doi.org/10.1016/j.trd.2022.103231.

[224] Ray Galvin. "Energy consumption effects of speed and acceleration in electric vehicles: Laboratory case studies and implications for drivers and policymakers". In: *Transportation Research Part D: Transport and Environment* 53 (2017), pp. 234–248. DOI: https://doi.org/10.1016/j.trd.2017.04.020.

[225] Samuel Sanford Shapiro et al. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3-4 (1965), pp. 591–611.

[226] J. P. Royston. "An Extension of Shapiro and Wilk's W Test for Normality to Large Samples". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 31.2 (Dec. 2018), pp. 115–124. DOI: 10.2307/2347973.

[227] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

A

Compact Matrix Form Notation

Definition 23 – Compact Matrix Form Notation

Vectors:
Let the first-stage decision vector be defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{r} \\ \boldsymbol{u} \end{bmatrix}, \quad \text{with dimension } n_1 = n_3 + n_4 + n_5,$$

where

- $\boldsymbol{x_1} \in \mathbb{B}^{n_3}$: edge selections x_{ij}^k, for $k \in \mathcal{K},\ i \in \mathcal{V}_0',\ j \in \mathcal{V}_{N+1}'$,
- $\boldsymbol{r} \in \mathbb{B}^{n_4}$: vehicle deployments r_k, for $k \in \mathcal{K}$,
- $\boldsymbol{u} \in \mathbb{R}^{n_5}$: load levels u_i^k, for $k \in \mathcal{K},\ i \in \mathcal{V}'$.

Let the associated first-stage cost vector be defined as

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c_1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \text{with dimension } n_1,$$

where $\boldsymbol{c_1} \in \mathbb{R}^{n_3}$: distance cost $d_{ij} c_{km}^k$, for $k \in \mathcal{K},\ i \in \mathcal{V}_0',\ j \in \mathcal{V}_{N+1}'$.

Let the candidate decision vector corresponding to scenario $s \in \mathcal{S}(\mathcal{U})$ be defined as

$$\boldsymbol{y}^s = \begin{bmatrix} \boldsymbol{\tau_1^s} \\ \boldsymbol{\tau_2^s} \\ \boldsymbol{\sigma_1^s} \\ \boldsymbol{\sigma_2^s} \\ \boldsymbol{\delta}^s \\ \boldsymbol{y_1^s} \\ \boldsymbol{y_2^s} \end{bmatrix}, \quad \text{with dimension } n_2 = 3n_4 + 2n_6 + n_7 + n_8,$$

where

- $\boldsymbol{\tau_1^s} \in \mathbb{R}^{n_6}$: location arrival times τ_i^s, for $i \in \mathcal{V}'$,
- $\boldsymbol{\tau_2^s} \in \mathbb{R}^{n_4}$: route start times τ_k^s, for $k \in \mathcal{K}$,
- $\boldsymbol{\sigma_1^s} \in \mathbb{R}^{n_6}$: location departure times σ_i^s, for $i \in \mathcal{V}'$,
- $\boldsymbol{\sigma_2^s} \in \mathbb{R}^{n_4}$: route end times τ_k^s, for $k \in \mathcal{K}$,
- $\boldsymbol{\delta}^s \in \mathbb{R}^{n_4}$: driver start times δ_k^s, for $k \in \mathcal{K}$,
- $\boldsymbol{y_1^s} \in \mathbb{R}^{n_7}$: SoCs on arrival y_i^{ks}, for $k \in \mathcal{K},\ i \in \mathcal{V}$,
- $\boldsymbol{y_2^s} \in \mathbb{R}^{n_8}$: SoCs on departure Y_f^{ks}, for $k \in \mathcal{K},\ f \in \mathcal{F}'$.

Let $\{\boldsymbol{y}^s\}$ be the shorthand notation for $\{\boldsymbol{y}^s : s \in \mathcal{S}(\mathcal{U})\}$.

Note that replacing all second-stage decision vectors $\boldsymbol{y}^s$ with a single vector $\boldsymbol{y}$ would correspond to a static robust optimization formulation.

Let the associated second-stage cost vector be defined as

$$\boldsymbol{d} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{d_1} \\ -\boldsymbol{d_1} \\ -\boldsymbol{d_2} \\ \boldsymbol{d_3} \end{bmatrix}, \quad \text{with dimension } n_2,$$

where

- $\boldsymbol{d_1} \in \mathbb{R}^{n_4}$: time cost c^k_{min}, for $k \in \mathcal{K}$,
- $\boldsymbol{d_2} \in \mathbb{R}^{n_7}$: energy cost c^i_{kWh} if $i \in \mathcal{F}'$ and 0 otherwise, for $k \in \mathcal{K},\ i \in \mathcal{V}$
- $\boldsymbol{d_3} \in \mathbb{R}^{n_8}$: energy cost c^f_{kWh}, for $k \in \mathcal{K},\ f \in \mathcal{F}'$.

Matrices:
Let the matrix $\boldsymbol{A} \in \mathbb{R}^{m_1 \times n_1}$ and the vector $\boldsymbol{b} \in \mathbb{R}^{m_1}$ denote the left-hand side and right-hand side coefficients of the first-stage constraints given by equations Eq. (3.3) to Eq. (3.13) and Eq. (3.36), when expressed in standard form.

Similarly, let the matrix $\boldsymbol{F}(\boldsymbol{z}^s) \in \mathbb{R}^{m_2 \times n_2}$ and the vector $f(\boldsymbol{z}^s) \in \mathbb{R}^{m_2}$ denote the left-hand side and right-hand side coefficients of the second-stage constraints from equations Eq. (3.14) to Eq. (3.18) and Eq. (3.30) to Eq. (3.35), in standard form for each scenario $s \in \mathcal{S}(\mathcal{U})$. The notation emphasizes that $\boldsymbol{F}(\boldsymbol{z}^s)$ and $f(\boldsymbol{z}^s)$ are affine functions of the uncertainty vector $\boldsymbol{z}^s \in \mathcal{U}$.

Let the matrices $\boldsymbol{G}(\boldsymbol{z}^s) \in \mathbb{R}^{m_3 \times n_1}$ and $\boldsymbol{H}(\boldsymbol{z}^s) \in \mathbb{R}^{m_3 \times n_2}$, along with the vector $\boldsymbol{j}(\boldsymbol{z}^s) \in \mathbb{R}^{m_3}$, denote the coefficients associated with the first-stage decision variables, second-stage decision variables, and the right-hand side, respectively, of the coupling constraints given in equations Eq. (3.19) to Eq. (3.25), in standard form for each scenario $s \in \mathcal{S}(\mathcal{U})$.

Finally, let the matrix $\boldsymbol{L}(\boldsymbol{z}^s) \in \mathbb{R}^{m_4 \times n_1}$ and the vector $l(\boldsymbol{z}^s) \in \mathbb{R}^{m_4}$ denote the coefficients of the left-hand side and right-hand side, respectively, of the redundant first-stage constraints from Eq. (3.27) to Eq. (3.29), again in standard form for each scenario $s \in \mathcal{S}(\mathcal{U})$.

B

Premature Convergence of the OSLP Algorithm

Counterexample 1 – Premature Convergence of the OSLP Algorithm

Claim:
Whenever a violating scenario $q \in \mathcal{S}(\mathcal{U})$ exists for the current first-stage decisions $\boldsymbol{x}$, so that $\Omega(\boldsymbol{x}, \boldsymbol{z}^q) = \emptyset$, the OSLP Algorithm will identify it.

Counterexample:
Consider a simplified instance of the HGEV-FPP-TECU problem with

- $\mathcal{U} = \mathcal{U}_{budget}$,
- $\mathcal{V} = \{a, b, c, d\}$,
- $\mathcal{K} = \{k\}$,
- k is a single vehicle,
- a and d are the entrance and exit of the depot, respectively,
- b is a depot charger,
- c is the only customer to be served.

Let the current routing solution $\boldsymbol{x}$ correspond to the route $a \rightarrow b \rightarrow c \rightarrow d$, which is in fact the only feasible route.

Let the traversal durations are uncertain within the intervals

$$t^s_{ab} \in [0, 0], \quad t^s_{bc} \in [3, 7], \quad t^s_{cd} \in [4, 6], \, \forall s \in \mathcal{S}(\mathcal{U}_{budget}).$$

Let the uncertainty budget be $\Gamma = 1$ and let the time windows $[e_i, \, l_i]$ be

$$[0, 12] \text{ for } i \in \{a, b, c\}, \text{ and } [4, 11] \text{ for } i \in \{d\}.$$

Let all other uncertain parameters be deterministic, with for all $s \in \mathcal{S}(\mathcal{U}_{budget})$:

- $e_a = \nu^s_a = \nu^s_b = \nu^s_c = \nu^s_d = 0$,
- $g^{ks}_b = 0$, so that vehicle k can depart from b with a full battery at time 0,
- Q_k sufficiently large to complete the route.

Let the nominal scenario $\boldsymbol{z}^{nom}$ be the last scenario added to the RMP, resulting in solution $(\boldsymbol{x}, \{\ldots, \boldsymbol{y}^{nom}\})$ with,

$$\tau^{\text{nom}}_a = \sigma^{\text{nom}}_a = \tau^{\text{nom}}_b = \tau^{\text{nom}}_b = 0,$$

$$t^{\text{nom}}_{bc} = 5, \quad t^{\text{nom}}_{cd} = 5,$$

$$\tau^{\text{nom}}_c = \sigma^{\text{nom}}_c = 0 + 5 = 5, \quad \tau^{\text{nom}}_d = 5 + 5 = 10.$$

The OSLP now maximizes the sum of one-step-ahead violations under the uncertainty budget Γ. Since the budget is enforced via constraints linear in the the uncertainty, the scenario that maximizes constraint violation lies at an extreme point of the uncertainty set, so that it is part of

$\text{ext}(\mathcal{U}_{\text{budget}})$.

That is, the scenario that maximizes constraint violation must be among

- q_1: $t_{bc}^{q_1} = 7$, $t_{cd}^{q_1} = 5$,
- q_2: $t_{bc}^{q_2} = 5$, $t_{cd}^{q_2} = 6$.

In both cases, the OSLP does not detect a one-step-ahead time violation, as

$$\begin{aligned} \sigma_b^{\text{nom}} + t_{bc}^{q_1} &= 0 + 7 = 7 \leq 12, \quad & \sigma_c^{\text{nom}} + t_{cd}^{q_1} &= 5 + 5 = 10 \leq 11, \\ \sigma_b^{\text{nom}} + t_{bc}^{q_2} &= 0 + 5 = 5 \leq 12, \quad & \sigma_c^{\text{nom}} + t_{cd}^{q_2} &= 5 + 6 = 11 \leq 11. \end{aligned}$$

However, scenario q_1 leads to a violation in the original problem, as

$$\tau_d^{q_1} \geq t_{bc}^{q_1} + t_{cd}^{q_1} = 7 + 5 = 12 > 11.$$

Furthermore, under q_1, the charging nor timing decisions can be improved any further to fix this violation, as $\sigma_b^{nom} = 0$, denoting the earliest possible time to leave location b.

That is $\Omega(\boldsymbol{x}, \boldsymbol{z}^{q_1}) = \emptyset$.

Despite this, the OSLP cannot distinguish between q_1 and q_2, as both are extreme points and both result in the same maximum objective value during OSLP. If q_1 is selected, which is just as likely as choosing q_2, this leads the algorithm to incorrectly conclude that no violating scenario exists, resulting in premature convergence.

C

Energy Consumption Modelling

C.0.1. Further Details on the Duty Cycles

The duty cycles employed in our energy consumption simulations were adopted from the Vehicle Energy Consumption calculation TOol (VECTO) [208], the official tool of the European Comission for regulating and standardizing CO_2 emissions and fuel consumption of HGVs. This tool, which is also based on a longitudinal energy consumption model, comes with duty cycles that are representative for different types of HGV operations and track the velocity and position over a traversed trajectory. VECTO makes a distinction between UrbanDelivery, InterUrban, SubUrban, RegionalDelivery and LongHaul operations. Where the first three are associated with lower average velocities and frequent deceleration, the regional profile mixes such sections with sections between cities where the velocity reaches higher values as the vehicle drives on regional or motorways. During the LongHaul cycle, the vehicle is mainly located on the motorway, showing a high average velocity and little acceleration and deceleration. In case the vehicle stopped at a certain position, for example, to wait at a traffic light, the stop duration in seconds was recorded. The resulting profiles are displayed in Fig. C.1.

On a critical note, the accelerations found in the profiles do not always appear to be physically sound. For example, the LongHaul profile starts with a velocity of 0 km/h to record a velocity of 83 km/h after having traversed only a single meter. This is highly unlikely, given that the maximum acceleration for HGEVs is estimated to be around $2\,\mathrm{m/s^2}$ [51]. Additionally, the included road gradients that range from $-10.93°$ to $13.69°$ might not be fully representative for the flat Dutch landscape.

As our energy simulation ran in the time domain, we converted the profiles by considering the difference in time between any two measurements based on the average velocity and the distance between the two. We enforced the stops by introducing additional points in time. We then linearly interpolated the velocities over the timesteps that did not directly correspond to a measurement in the distance domain to obtain a velocity value per second of the simulation, and calculated the accelerations accordingly. The interpolation also aimed to mitigate unrealistically high decelerations and accelerations occurring immediately before or after stops. The resulting duty cycles are illustrated in Fig. 4.6. Notably, in the regional delivery cycle plot, a significant stop occurs just after 5000 seconds. The subsequent linear segment reflects the interpolated velocity during the stop, effectively reducing the acceleration magnitude when the vehicle resumes its trip. It is also worth noting that the duty cycles differed in length, with most spanning approximately 100 km, while the SubUrban cycle covered only 23.5 km. Additionally, the characteristic average velocities tended to be slightly lower when analyzed in the time domain compared to the distance domain. This discrepancy arose from the interpolation process and the inclusion of additional stop points, which reduced the average velocity.

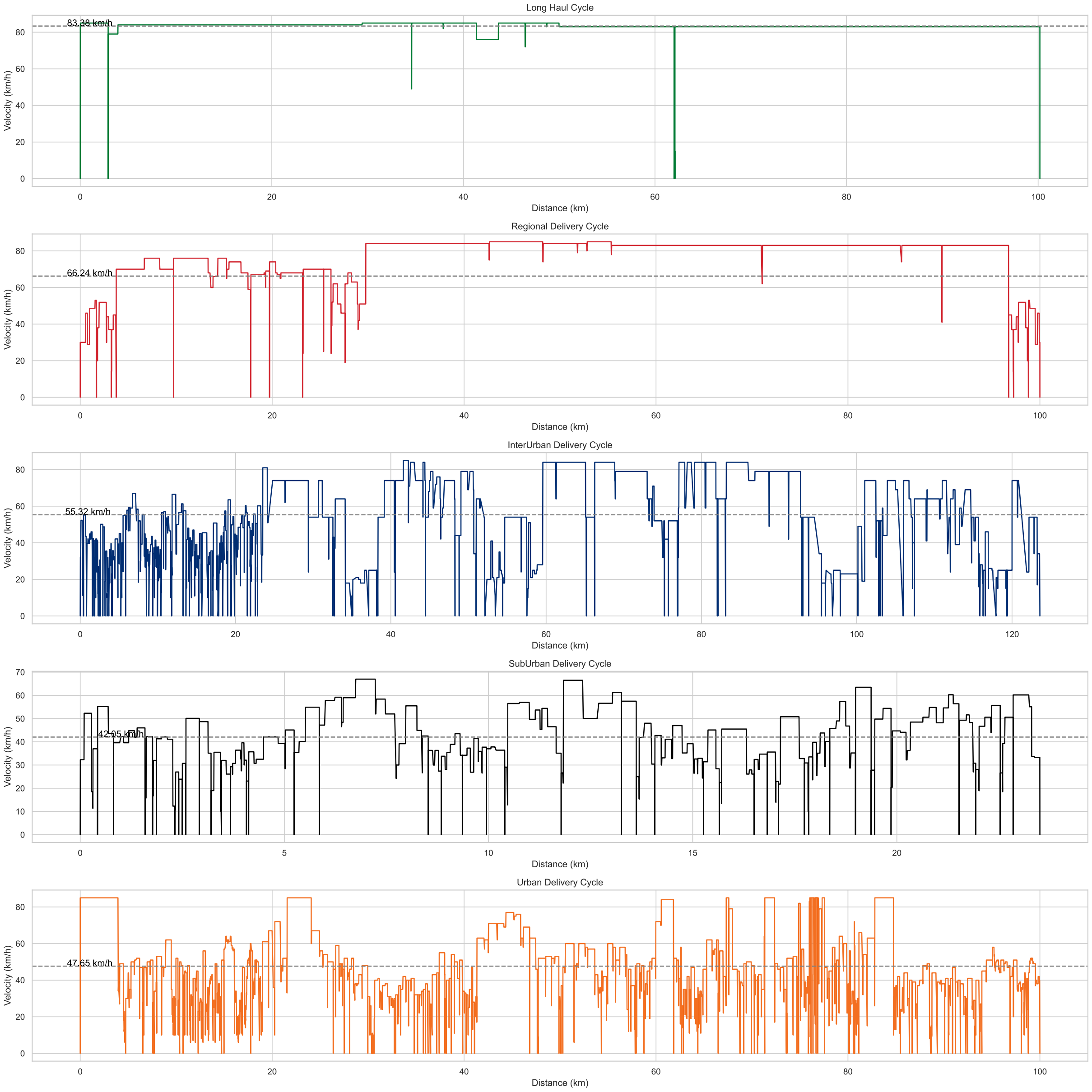


Figure C.1: Original duty cycles for LongHaul, RegionalDelivery, InterUrban, SubUrban and UrbanDelivery operations, showing the recorded velocity and traversed distance over a 100 km trajectory. The dashed line indicates the characteristic average velocity for the profile in the distance domain.

C.0.2. Generating Synthetic Duty Cycles

Note that a single average velocity value can correspond to numerous possible velocity profiles, each resulting in a different perspective on the energy consumption. The VECTO duty cycles provide only one such perspective for a limited set of average velocities, so that intermediate velocities are treated as a mix of the closest two available profiles. In an attempt to address this limitation, we generated synthetic duty cycles for a given average velocity using an MILP model. This model acted like a controller and imposed constraints on maximum velocity and acceleration, ensured consistency with the target average velocity, and incorporated stochastic elements, such as acceleration and deceleration events, to emulate real-world traffic dynamics. At higher average velocities, the model was more constrained, as it had to maintain near-maximum velocities to complete the trajectory within the allotted time. In contrast, lower average velocities offered greater flexibility in velocity assignment across time steps, occasionally resulting in irregular or random patterns. When the profiles corresponding to different average velocities were evaluated through MC simulation, they produced very similar energy consumption distributions, thereby offering limited discriminatory capacity with respect to the average velocity. Therefore, we chose to continue using only the VECTO profiles, as they exhibited more variations in energy consumption in response to changes in average velocity, aligning more closely with our expectations and reality.

C.0.3. Analyzing the Simulation Results

As discussed in Subsection 2.4.1, velocity is the only quadratic term in the longitudinal energy model, making it an important factor at higher velocities. Consequently, one might intuitively expect energy consumption to increase with average velocity due to greater aerodynamic drag. However, Table 4.5 reveals higher energy consumption values for the low-velocity urban driving profiles compared to the higher-velocity regional and long-haul profiles. While this observation aligns with expectations for HGVs with combustion engines, which lack regenerative braking, it contrasts with findings for battery electric passenger vehicles, such as those reported by Konzept et al. [222] and Mamarikas et al. [223], where lower average velocities typically correspond to lower energy consumption due to effective energy recuperation. However, the study by Galvin [224] found that the energy consumption of passenger BEVs follows a parabolic trend: it decreases up to around 60 km/h, after which it begins to rise again due to increasing aerodynamic drag. We hypothesize that for HGEVs, this inflection occurs at higher velocities as a result of their greater aerodynamic efficiency relative to their mass, which explains why we observe a continuous decrease in energy consumption over the simulated velocity range.

Looking at the longitudinal dynamics model, the higher mass of HGEVs compared to passenger BEVs increases the influence of the acceleration component in the traction force. In the LongHaul cycle, the vehicle's traction force component is primarily influenced by aerodynamic drag due to the high but steady velocity. In contrast, for the UrbanDelivery cycle, where aerodynamic drag is limited but accelerations are more common, the acceleration component is most dominant. Fig. C.2 illustrates this dynamic by comparing the cumulative energy demanded by the engine (excluding auxiliary loads) with the energy recovered through regenerative braking. Despite lower aerodynamic losses, the frequent acceleration events in the UrbanDelivery cycle result in a net increase in energy demand. However, this cycle also offers more opportunities for regenerative breaking. When we sum both effects and include the auxiliary power consumption, the cumulative energy consumption curves for both cycles (as shown in Fig. C.3) exhibit similar slopes. Yet, because the UrbanDelivery cycle takes approximately 2.5 times longer to complete than the LongHaul cycle, its total energy demand is proportionally larger.

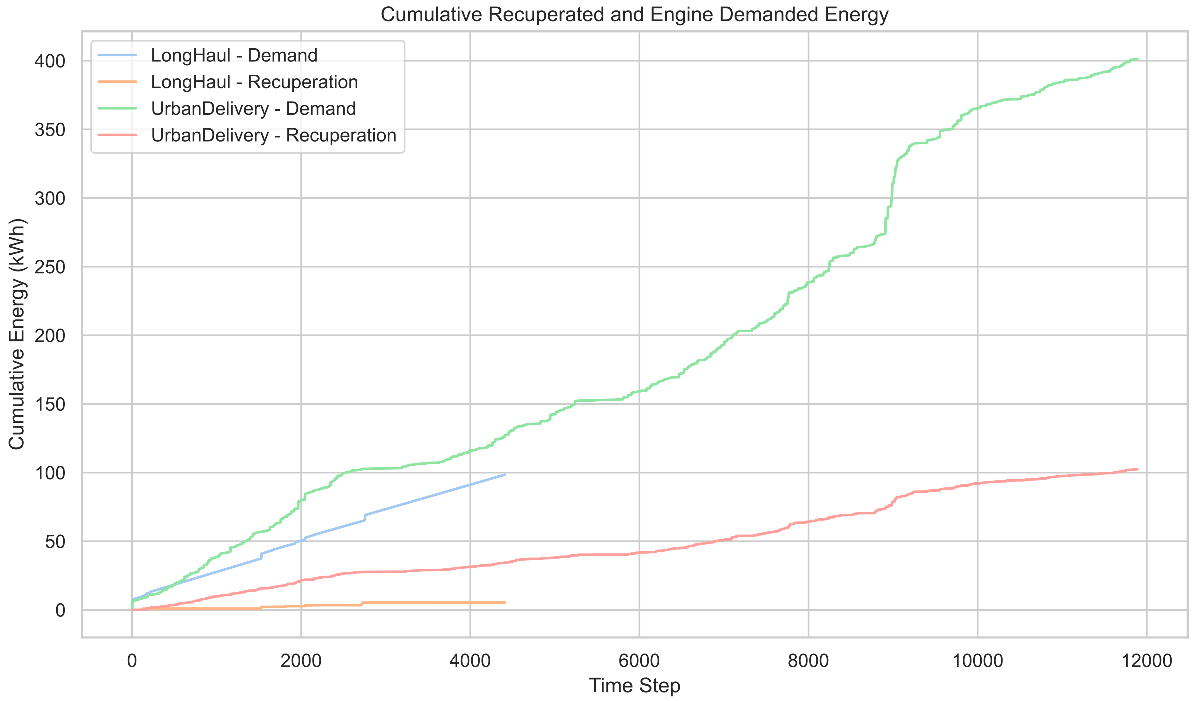


Figure C.2: Cumulative recuperated energy and energy demanded by the engine for the LongHaul and UrbanDelivery cycle over time under simulation with a vehicle mass level of 40,000 kg and mean values for the uncertain parameters

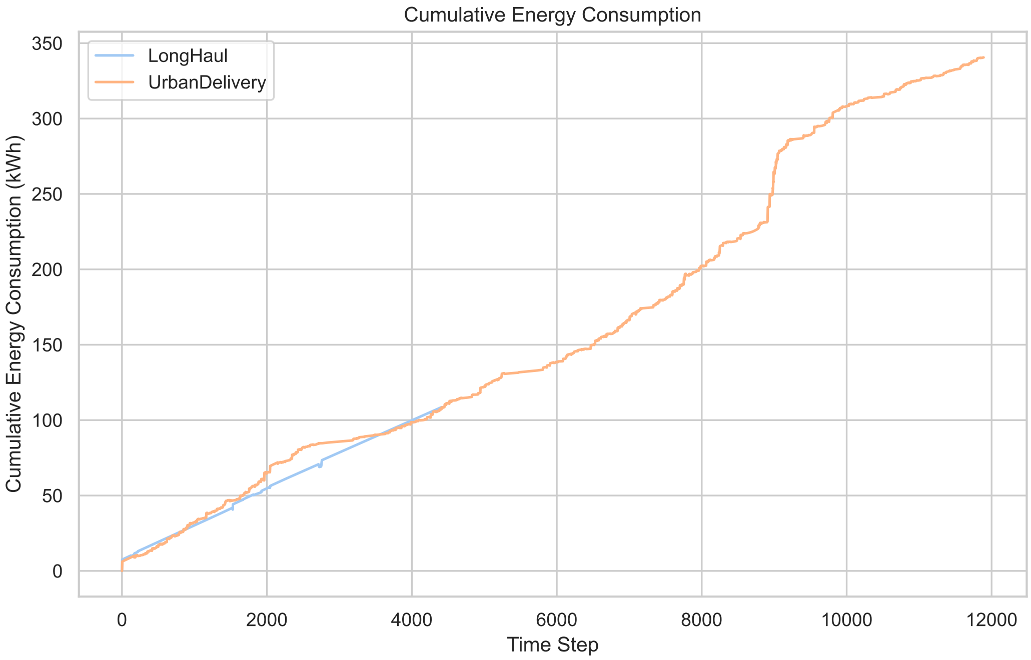


Figure C.3: Cumulative energy consumption for the LongHaul and UrbanDelivery cycle over time under simulation with a vehicle mass level of 40,000 kg and mean values for the uncertain parameters

Assessing Normality

Fig. C.5 presents a histogram and boxplot for the LongHaul duty cycle at 40 tonnes. The simulated energy consumption values exhibit a distribution that visually resembles a normal distribution. Notably, the distribution is centered around a mean of 1.1 kWh/km, which aligns exactly with the reference value reported by Volvo [197]. To statistically assess normality, we applied the Shapiro-Wilk test [225] across all combinations of duty cycle and mass level. The resulting test statistics were consistently **0.99** out of 1.00, indicating a strong similarity to a normal distribution. However, the corresponding p-values were all $\mathbf{0} < 0.05$, which technically leads to the rejection of the null hypothesis of normality. It is important to note that with large sample sizes, normality tests like Shapiro-Wilk tend to detect even negligible deviations from normality, often resulting in rejection of the hypothesis despite the data being practically normal [226]. Therefore, based on visual inspection and the high test statistics, we conclude that the data can be considered normally distributed. This allows us to reason about Confidence Intervals (CI), in the sense that approximately 95% of the data resides between two standard deviations.

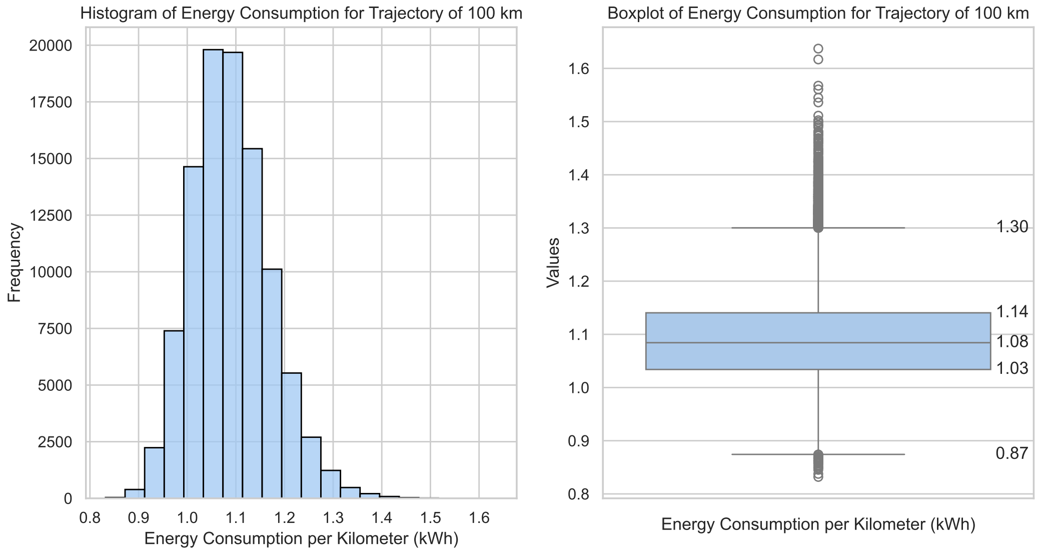


Figure C.4: Histogram and boxplot of the resulting energy consumption values per kilometre resulting from a MC simulation on the LongHaul duty cycle with a mass level of 40,000 kg

Relation with Vehicle Mass

Fig. C.5 illustrates how energy consumption increases across all duty cycles as the mass increases. Notably, the three profiles that include urban conditions exhibit both higher variability and steeper slopes compared to the regional and long-haul profiles. This confirms that energy consumption in urban settings is more sensitive to changes in mass. The flatter slopes and lower standard deviations observed in the regional and long-haul cycles indicate more consistent energy consumption, likely due to the steadier driving conditions with fever acceleration and deceleration. Consequently, in the UrbanDelivery cycle, the energy consumption of the simulated HGEV is nearly twice that observed in the LongHaul profile when operating under curb mass. This difference becomes even more pronounced under full load conditions, where the energy consumption in urban settings reaches approximately 3.5 times that of the long-haul equivalent.

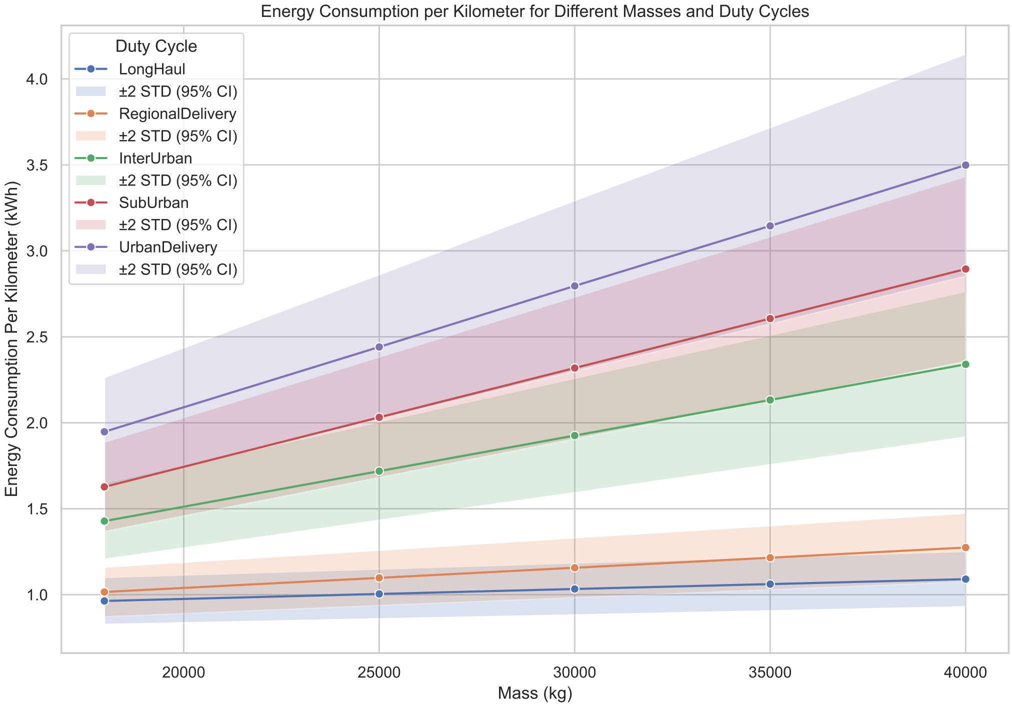


Figure C.5: Energy consumption per kilometre for different duty cycles and mass levels resulting from the MC simulation

C.0.4. Further Analysis of the Energy Function

To find appropriate values of the edge-independent parameters, we fitted a linear regression model using the Scikit-Learn library [227] with the target variable being the mean energy consumption per kilometre and the feature variables being the total vehicle mass (coefficient α^k), the inverse velocity (coefficient γ^k), and their product (coefficient κ^k). The model was fitted on output data from the MC simulations, which was split into training and testing sets using an 80% to 20% ratio, yielding five random test points. The resulting function is shown graphically in Fig. C.6 to Fig. C.8 and the resulting parameter values are listed in Table C.1.

Parameter	Description	Value
β^k	Intercept	0.963
α^k	Mass coefficient	-3.970×10^{-5}
γ^k	Inverse velocity coefficient	-1.125×10^{-1}
κ^k	Interaction coefficient	5.511×10^{-5}

Table C.1: Estimated Parameters

Note that, unlike in strictly additive linear models, the coefficients here cannot be directly interpreted as measures of individual influence, as the interaction term means the relationship is conditional.

Evaluation of the Resulting Function

The insights drawn from Fig. C.5 support the inclusion of the term κ^k, which scales both the mass term $u_j^k + m_c^k$ and the inverse velocity term t_{ij}^s. After all, the slope of energy consumption per kilometre appears to vary depending on the combination of mass and average velocity (and thus inverse velocity).

To evaluate the function fit, we calculated the Mean Squared Error (MSE) and the coefficient of determination (R^2) on the test set, yielding values of **0.02** and **0.95**, respectively. These values indicate that the function approximation represents the means of the simulated data well.

Graphically, we observe in Fig. C.7 that the function tends to bend around the data points for the regional delivery and suburban profiles, while matching the other profiles less accurately. This behavior is also evident in Fig. C.9 and Fig. C.10, which compare the energy consumption values obtained from the MC simulations with those predicted by the approximated function across various duty cycles and mass levels.

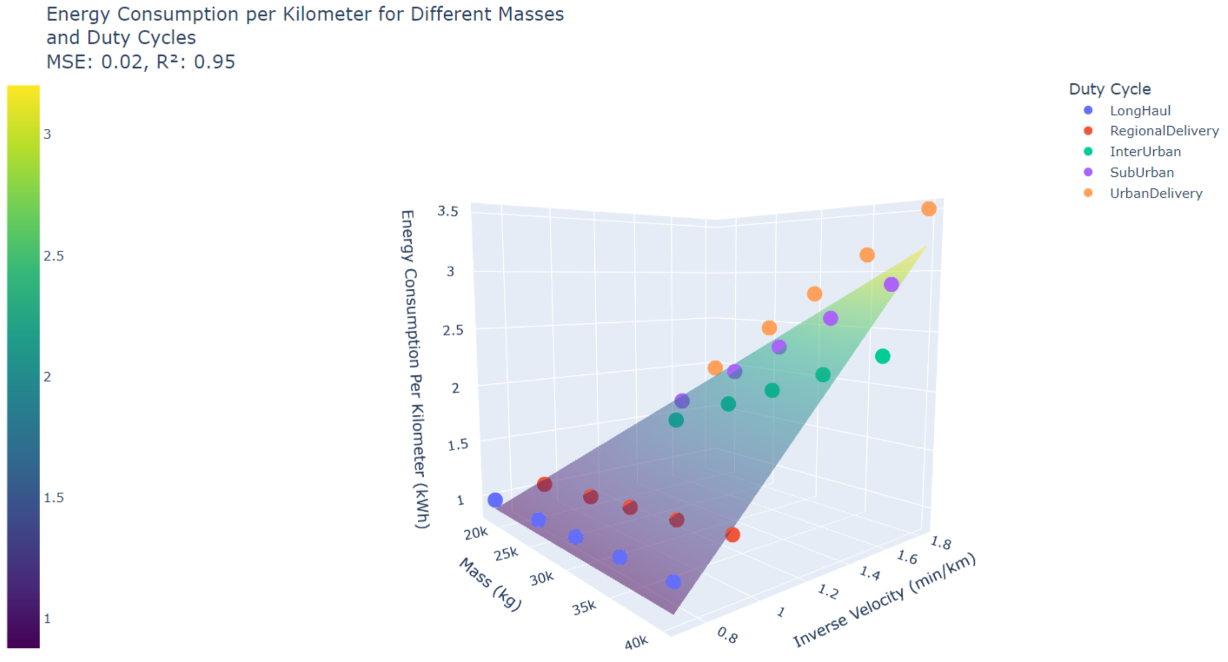


Figure C.6: Perspective 1 on the approximated energy consumption function, showing the average energy consumption per kilometre given the simulated inverse velocities and mass levels

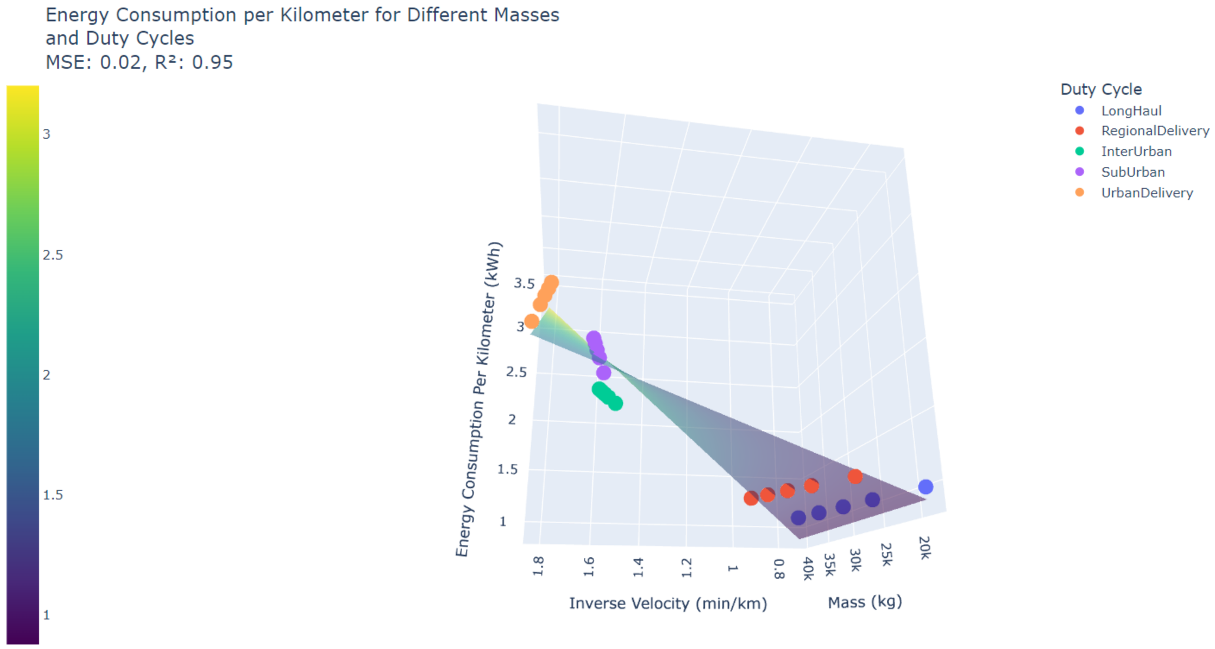


Figure C.7: Perspective 2 on the approximated energy consumption function, showing the average energy consumption per kilometre given the simulated inverse velocities and mass levels

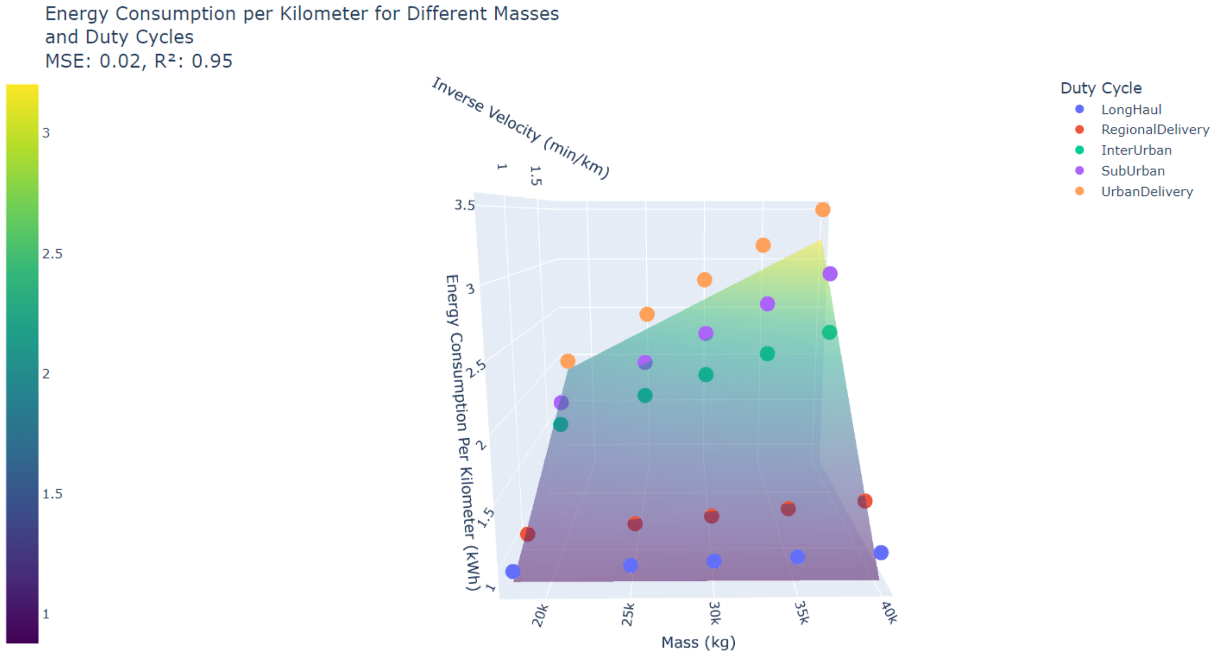


Figure C.8: Perspective 3 on the approximated energy consumption function, showing the average energy consumption per kilometre given the simulated inverse velocities and mass levels

Introducing a Robustness Multiplier

From a duty cycle perspective, Fig. C.9 shows that the function approximates the energy consumption for the LongHaul cycle as being entirely independent of the load level. This assumption is physically inaccurate and may lead to underestimations of energy consumption, particularly at high average velocities in combination with high load levels. For the UrbanDelivery cycle, the function also tends to underestimate energy consumption compared to the mean values from the MC simulations. In contrast, for the InterUrban cycle, the predicted energy consumption is very robust across load levels and aligns well with the upper bounds of the 95% confidence interval resulting from the MC simulations. The predictions for the RegionalDelivery cycle are slightly higher than the simulation mean values, while those for the SubUrban profile closely match the mean. Fig. C.10 further illustrates that energy consumption is underestimated at both ends of the velocity spectrum compared to the mean simulation values, although the predicted values generally remain within the 95% confidence bounds for each load level. In the mid-range velocities, the function tends to overestimate consumption, aligning with the worst-case values from the simulations.

Given that HGEVs are expected to operate at average velocities falling between those of the RegionalDelivery and InterUrban cycle (see Fig. 4.5), the function offers a reasonably conservative estimate of nominal energy consumption for a given edge distance, load level, and traversal time. Nonetheless, more risk-averse fleet operators may wish to apply a robustness multiplier $\psi^k > 1$ to the estimated energy values to ensure higher levels of robustness. Fig. C.9 and Fig. C.10 present results for several robustness coefficients $\psi^k \in [1.1, 1.2, 1.45]$. Choosing $\psi^k = 1.1$ corresponds to a mildly conservative strategy, ensuring that predicted energy values for velocities within the RegionalDelivery to InterUrban range always exceed the upper bound of the 95% confidence interval. This roughly imposes a chance constraint that ensures battery depletion risk remains below 5%, assuming average velocities remain within the simulated range $[37.65, 64.85]$ km/h. A coefficient of $\psi^k = 1.3$ represents a more robust approach, covering all profiles except LongHaul. To extend robustness to high average velocities of over 80 km/h, a coefficient of $\psi^k = 1.45$ or greater is required. However, such a high value significantly overestimates energy consumption at lower velocities, thereby inflating projected operational costs.

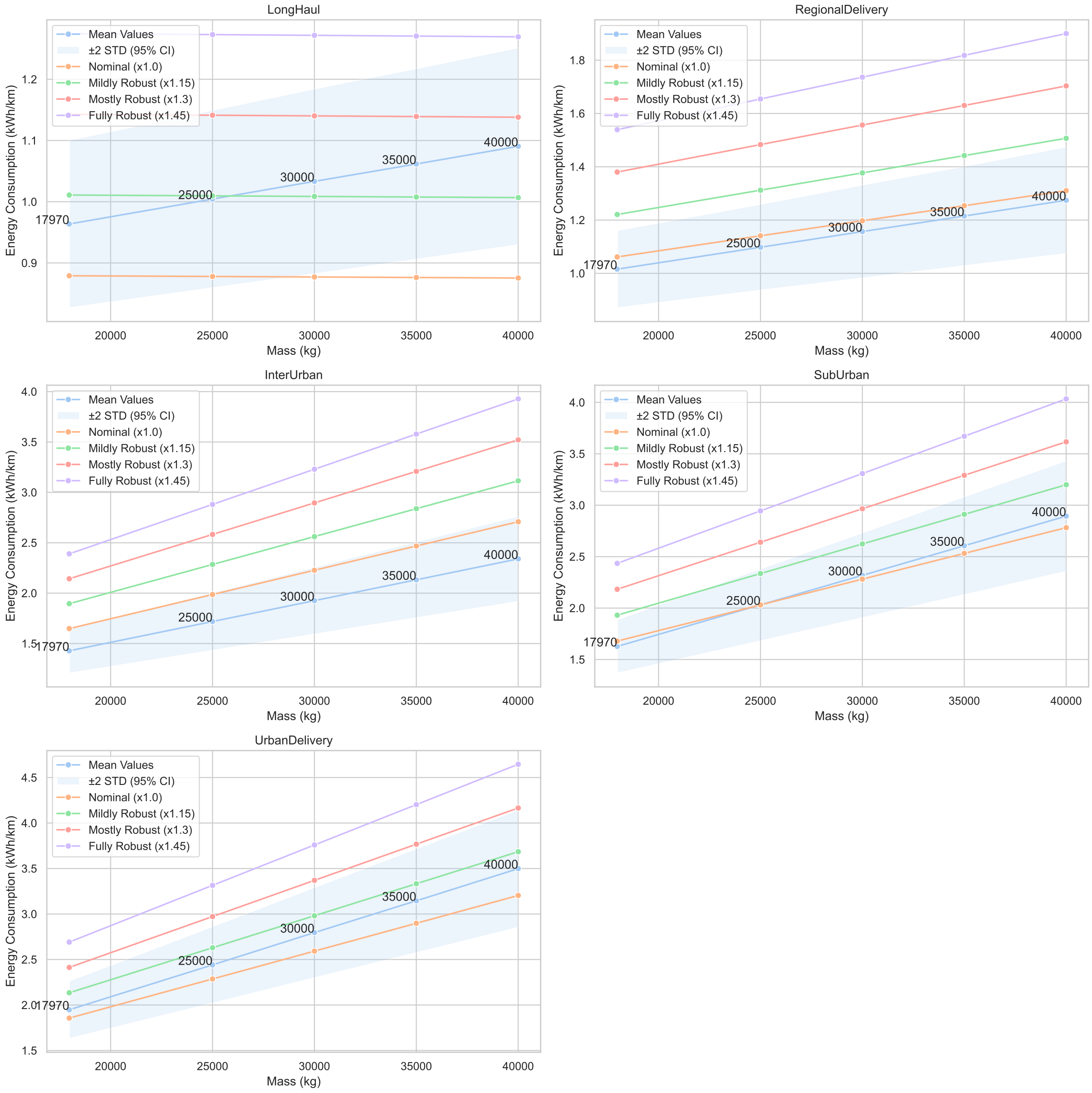


Figure C.9: Simulated and approximated energy consumption per duty cycle for all mass levels, illustrating the mean values and 95% confidence intervals from the Monte Carlo simulations, alongside the approximated function values for various robustness coefficients ψ^k

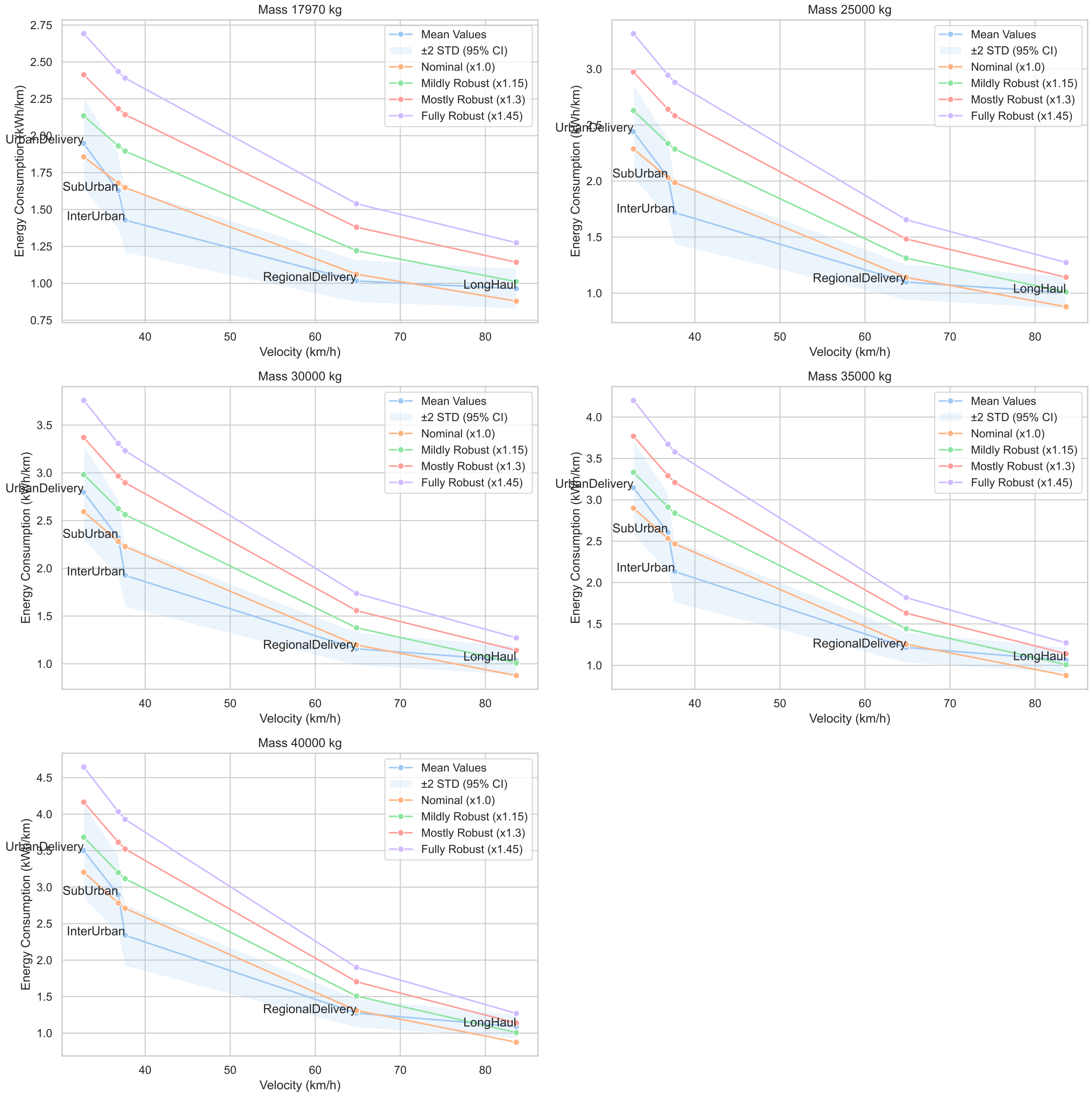


Figure C.10: Simulated and approximated energy consumption per mass level for all duty cycles, illustrating the mean values and 95% confidence intervals from the Monte Carlo simulations, alongside the approximated function values for various robustness coefficients ψ^k

D

Additional Figures and Tables

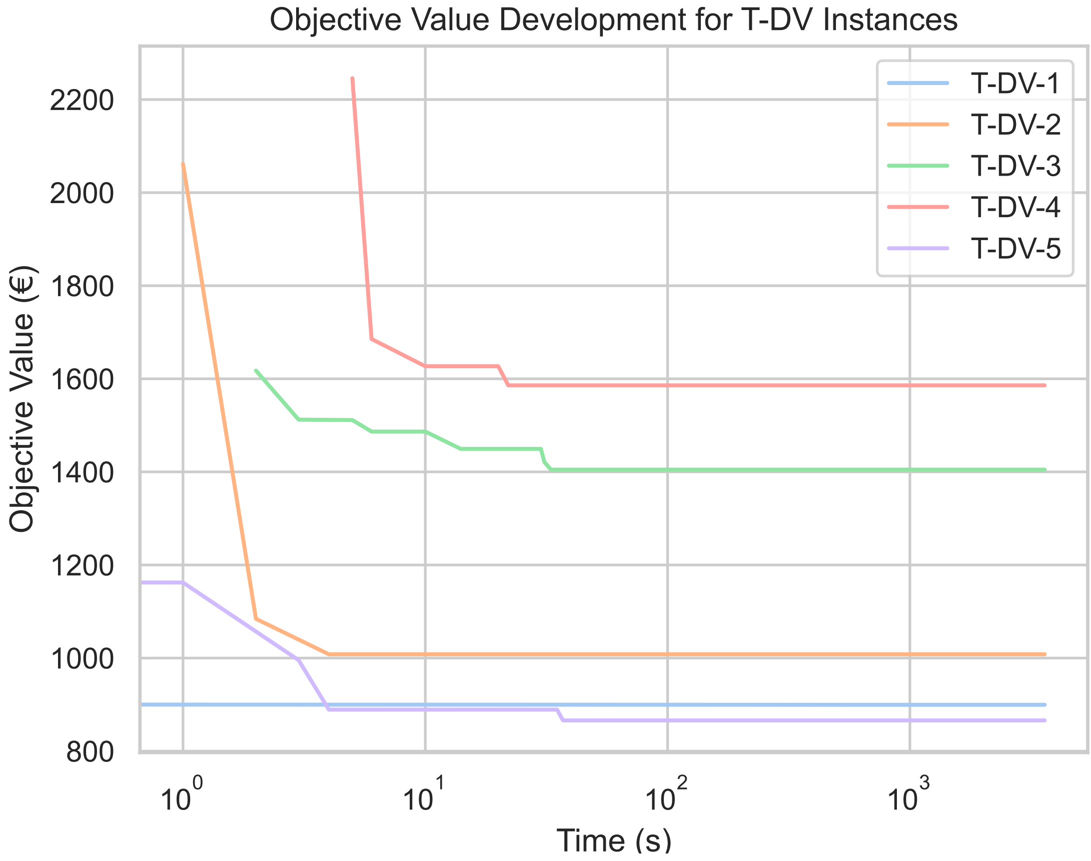


Figure D.1: Development of the objective value for the T-DV instances given the worst-case scenario $\boldsymbol{z}^{worst}$ solved using the EXA-DET over the course of an hour. Time is represented on a logarithmic scale

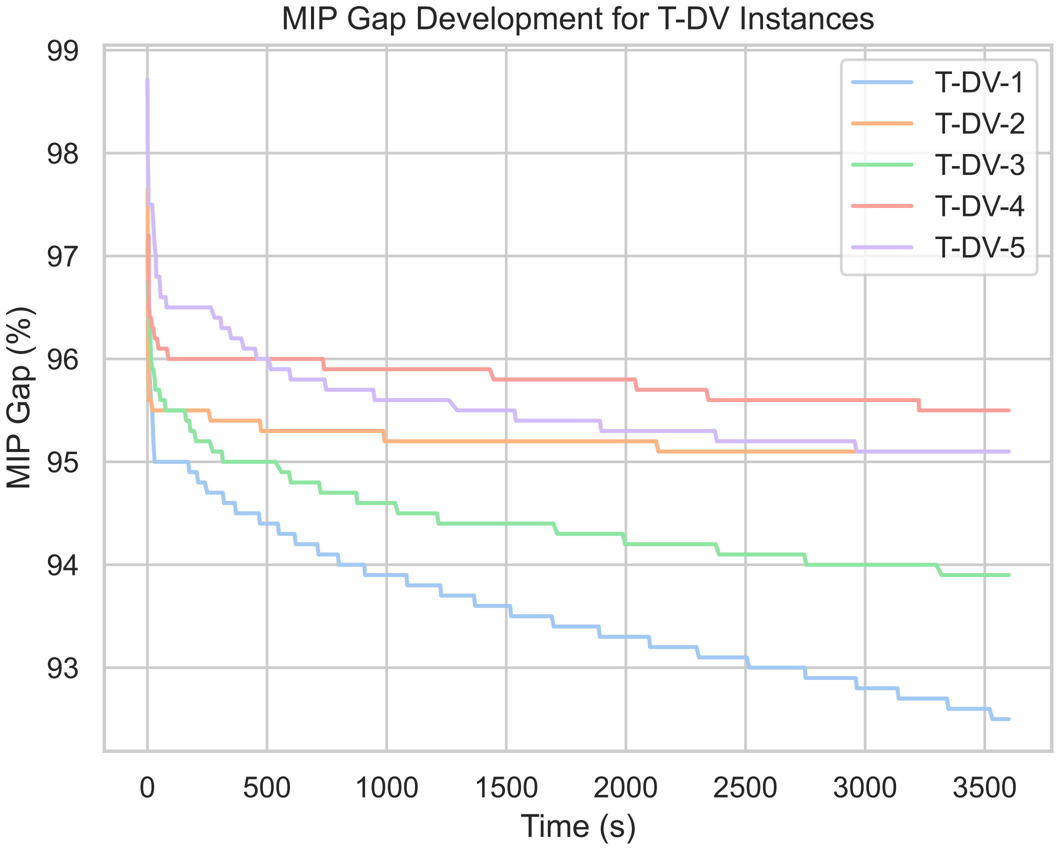
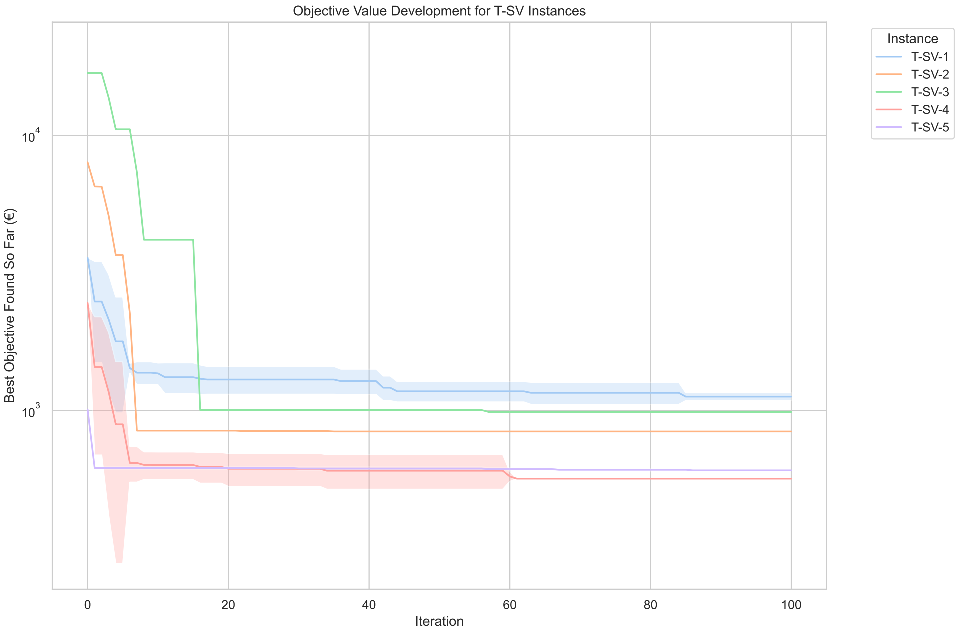
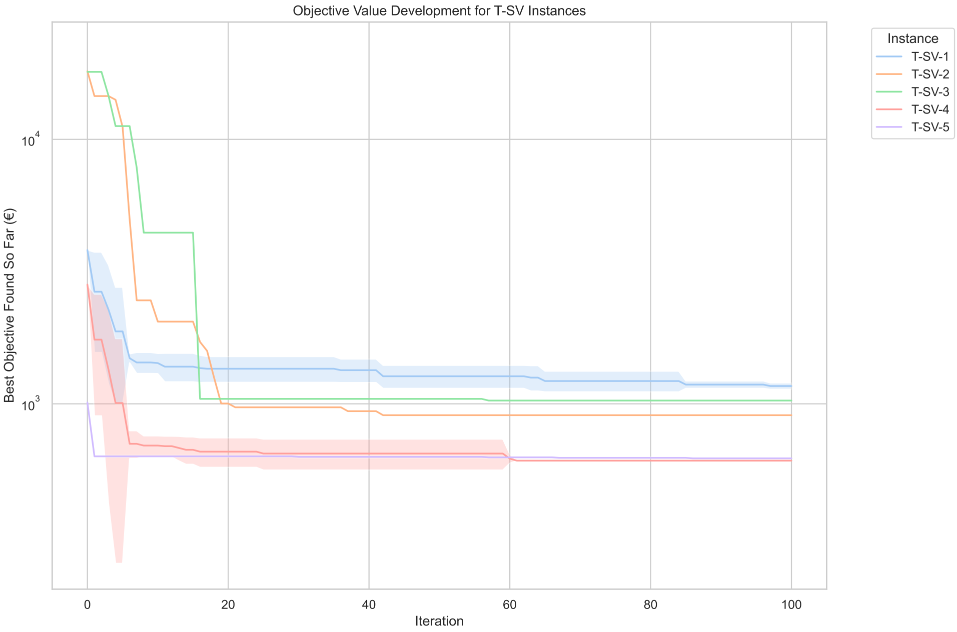


Figure D.2: Development of the MIP gap for the T-DV instances given the worst-case scenario $\boldsymbol{z}^{worst}$ solved using the EXA-DET over the course of an hour

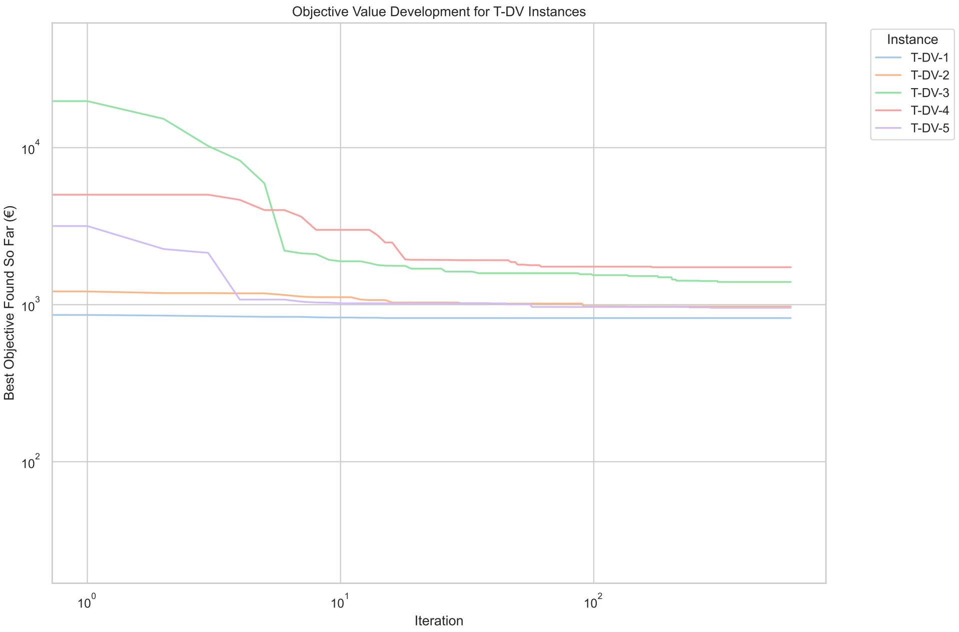


(a) Nominal Scenario

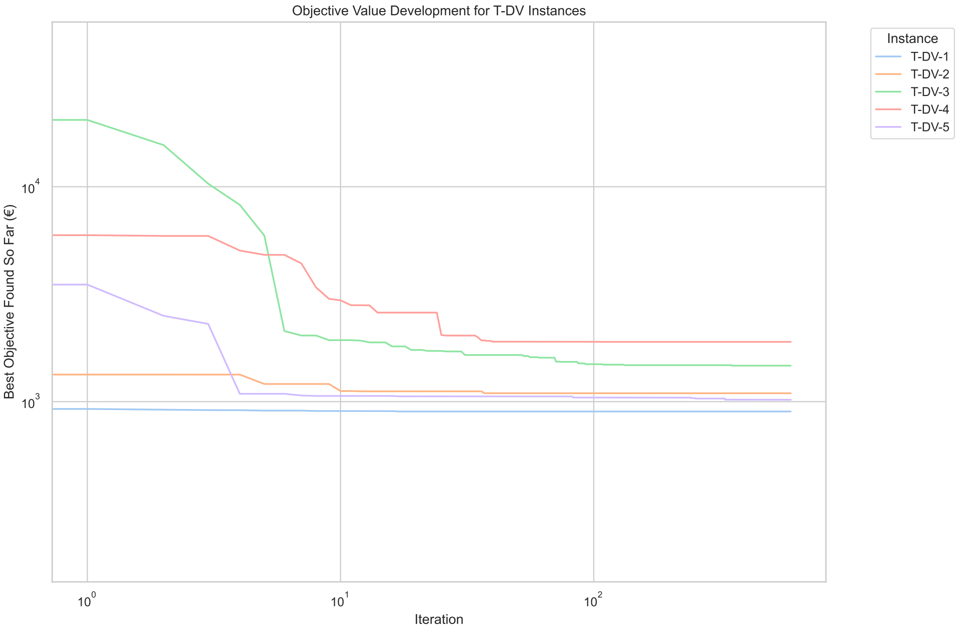


(b) Worst-case Scenario

Figure D.3: Development of the best objective value found using the META-DET solver on instances of the class T-SV over 100 iterations and five runs with varying random seeds. The objective value is represented on a logarithmic scale.

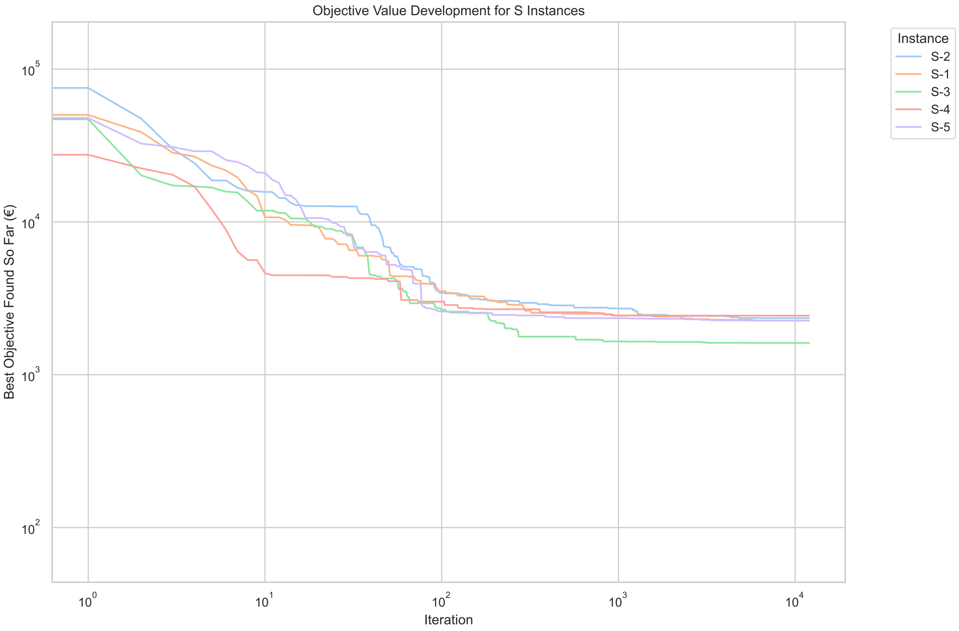


(a) Nominal Scenario

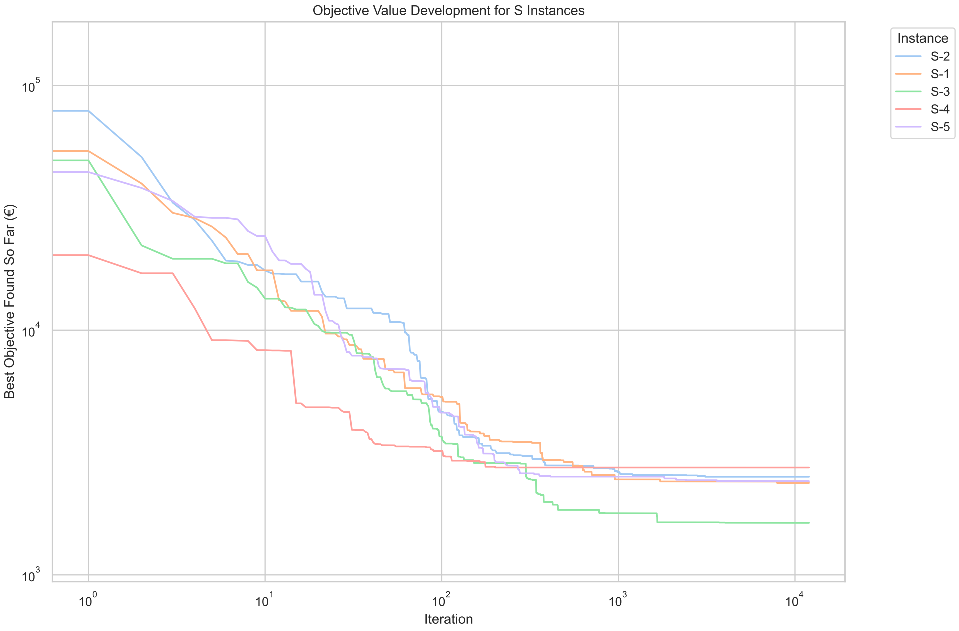


(b) Worst-case Scenario

Figure D.4: Development of the best objective value found using the META-DET solver on instances of the class T-DV over 600 iterations and five runs with varying random seeds. Both axes are represented on a logarithmic scale.



(a) Nominal Scenario



(b) Worst-case Scenario

Figure D.5: Development of the best objective value found META-DET on instances of the class S over 12,000 iterations and five runs with varying random seeds. Both axes are represented on a logarithmic scale.

Instance	**Runtime (s)**		**Avg. Obj. (€)**		
	EXA-DET	META-DET	EXA-DET	META-DET	Δ%
T-SV-1	0.508 ± 0.045	1.197 ± 0.552	993.35	1124.81 ± 33.09	9.85
T-SV-2	0.344 ± 0.019	0.732 ± 0.084	806.45	840.32 ± 0.00	4.20
T-SV-3	0.220 ± 0.022	0.441 ± 0.557	943.14	989.27 ± 0.00	4.89
T-SV-4	1.048 ± 0.066	0.688 ± 0.335	566.30	566.30 ± 0.00	**0.00**
T-SV-5	0.516 ± 0.049	0.463 ± 0.261	447.26	606.99 ± 0.00	**35.71**
T-SV Overall	0.526 ± 0.292	0.704 ± 0.485	751.30	825.54 ± 215.71	**10.93**
T-DV-1	0.800 ± 0.447	0.587 ± 0.076	821.88	821.88 ± 0.00	**0.00**
T-DV-2	1.000 ± 0.000	0.587 ± 0.076	933.67	974.30 ± 26.73	2.92
T-DV-3	4.400 ± 0.548	3.779 ± 1.464	1323.04	1396.02 ± 0.00	5.52
T-DV-4	4.400 ± 0.548	1.502 ± 0.529	1506.71	1731.90 ± 25.30	12.89
T-DV-5	1.000 ± 0.000	2.208 ± 1.664	797.41	955.40 ± 43.19	16.27
T-DV Overall	$\mathbf{2.120 \pm 1.965}$	$\mathbf{1.921 \pm 1.497}$	1076.54	1175.90 ± 339.23	**7.52**
S-1	-	119.224 ± 83.218	-	2261.55 ± 53.50	-
S-2	85.403 ± 11.150	149.279 ± 52.700	2092.04	2349.44 ± 90.39	**9.14**
S-3	52.807 ± 11.952	76.831 ± 54.378	1438.36	1615.13 ± 21.59	**10.73**
S-4	-	16.106 ± 11.833	-	$\mathbf{2436.24 \pm 229.39}$	-
S-5	-	136.983 ± 59.280	-	2262.97 ± 65.39	-
S Overall	-	$\mathbf{99.685 \pm 74.905}$	-	2185.07 ± 314.72	-

Table D.1: Comparison of runtimes and objective values for the nominal scenario $\boldsymbol{z}^{nom}$ across T-SV, T-DV, and S instances solved by EXA-DET and META-DET over five runs with varying random seeds and a one-hour timeout. Δ% indicates the performance gap between the best objective value from META-DET and the value found by EXA-DET

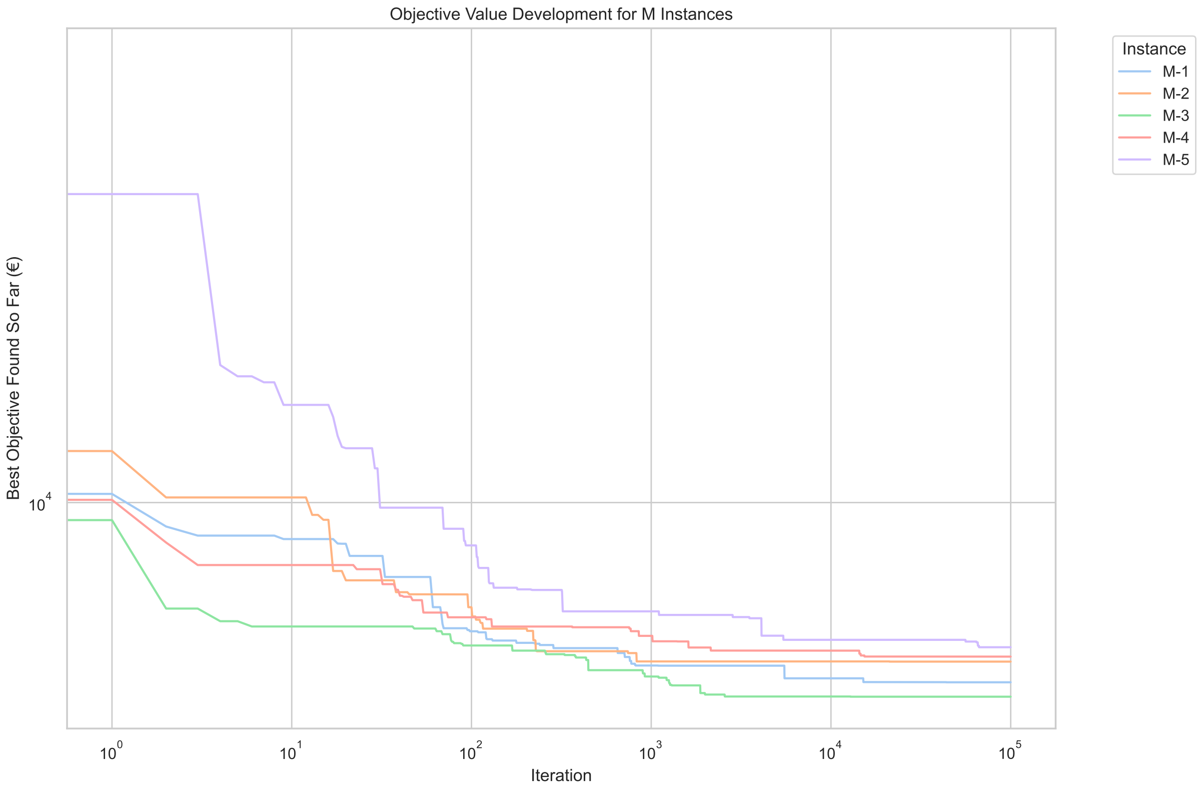


Figure D.6: Development of the best objective value found using META-DET on instances of the class M over 100,000 iterations. Both axes are represented on a logarithmic scale.

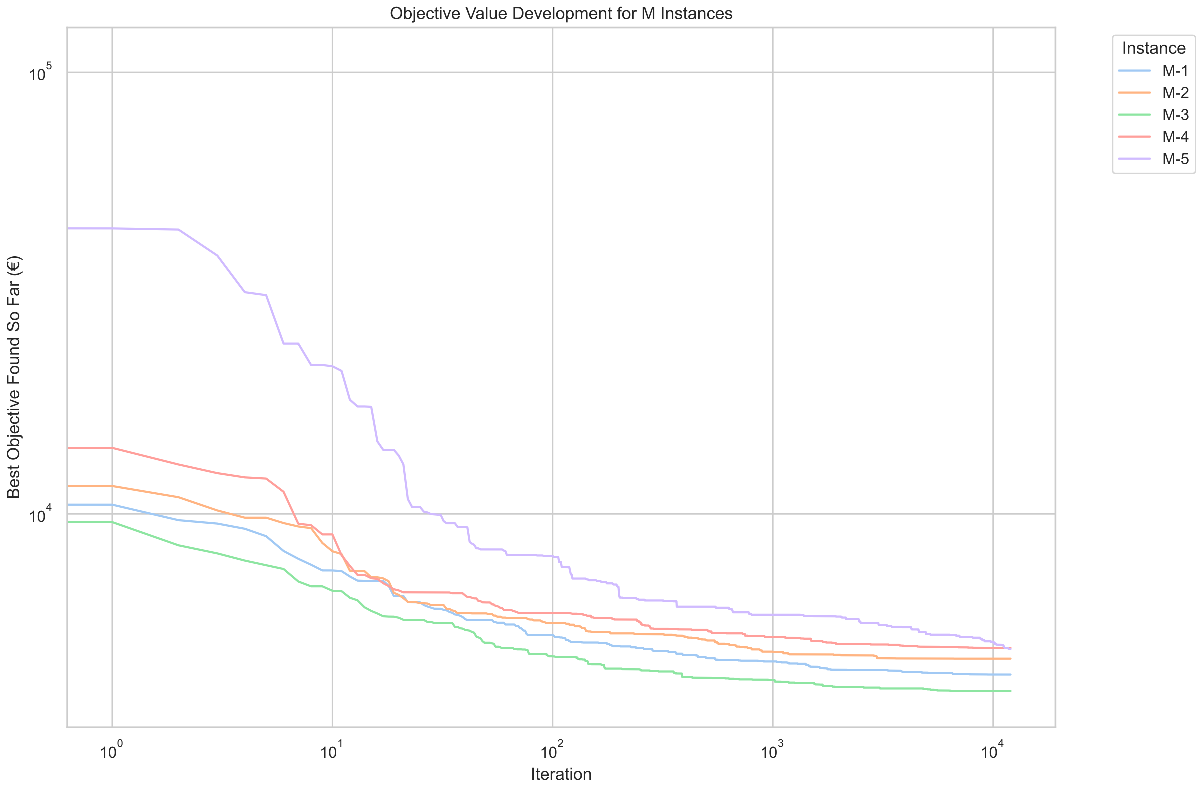


Figure D.7: Development of the best objective value found using META-DET on M instances under the worst-case scenario $\boldsymbol{z}^{nom}$ over 12,000 iterations and five runs with varying random seeds. Both axes are represented on a logarithmic scale.

Instance	Runtime (s)	Num. Feas. Runs	Best Feas. Obj. (€)	Worst Overlap (min)
M-1	1153.94 ± 47.91	4/5	4279.04	4.42
M-2	1328.64 ± 73.09	4/5	4543.83	18.95
M-3	1255.96 ± 43.86	4/5	3840.18	11.38
M-4	1410.08 ± 7.50	4/5	4897.31	0.23
M-5	1369.76 ± 11.18	2/5	4970.77	7.60
Overall	1300.92 ± 102.01	**18/25**	3840.18	**18.95**

Table D.2: Performance comparison for solving the M instances under the nominal scenario $\boldsymbol{z}^{nom}$ using the META-DET solver with a budget of 12,000 iterations over five runs with varying random seeds