Integrated Optimization of Charging Infrastructure for Battery-Electric Trucks Using Tour-Based Freight Data

By

Mingyan Jin

September 2025

TU Delft

MSc thesis in Transport, Infrastructure, and Logistics

Integrated Optimization of Charging Infrastructure for Battery-Electric Trucks Using Tour-Based Freight Data

Author: Mingyan Jin
Student Number: 6050972

Supervisors:
Prof. dr. ir. Lóri Tavasszy
Dr. ir. Alessandro Bombelli

Faculty of Civil Engineering and Geosciences
Delft University of Technology

September 2025

Abstract

The decarbonization of heavy-duty road freight transport requires overcoming the dual challenge of limited driving range and insufficient charging infrastructure for battery-electric trucks (BETs). This thesis develops a nationwide bi-level optimization framework that integrates the deployment of static charging stations (SCS), electrified road systems (ERS), and truck-level battery assignment, leveraging large-scale tour-based freight data from the Netherlands. By operating explicitly at the tour level rather than the trip level, the framework captures cumulative energy feasibility across multiple linked trips, thereby providing a more realistic representation of freight operations.

The model is solved using a Genetic Algorithm (GA), capable of evaluating more than 1.5 million tours and 3.5 million trips, and validated against exact MILP solutions on small instances. The results show that the optimization raises feasibility from 58% to 89.9%, reducing infeasible tours to 10.1%, while overall fitness improves by 34.7%. ERS emerges as the backbone of electrification, with 12,792 km deployed (€25.7 billion, 65.6% of CAPEX), while 251 SCS facilities (€50.2 million, <1% of CAPEX) provide low-cost redundancy at regional hubs. Battery allocation is highly heterogeneous: 25% of trucks operate on 90 kWh, 40% on 600 kWh, and the remainder on intermediate sizes, yielding an average of 357 kWh—closely aligned with ElaadNL's benchmark of 289.5 kWh/day. This heterogeneity reduces battery CAPEX by approximately 19% compared to a uniform-capacity baseline. Operating expenditures (OPEX) are dominated by ERS charging, while penalty costs for infeasible tours remain substantial, averaging €362 per unserved tour.

These findings demonstrate that nationwide electrification is feasible under a layered strategy: ERS as the long-haul backbone, SCS as regional redundancy, and heterogeneous batteries as cost optimizers. The study advances the literature by moving from trip-level to tour-level modelling at unprecedented scale, explicitly quantifying infeasibility, and decomposing system costs into CAPEX, OPEX, and penalties. The results provide actionable insights for policymakers and industry stakeholders seeking cost-effective and operationally viable pathways for freight decarbonization in the Netherlands and beyond.

Keywords: Freight electrification; Battery-electric trucks; Bi-level optimization; Genetic Algorithm; Electric Road Systems (ERS); Static Charging Stations (SCS); Tour-level modelling; Infrastructure planning; Infeasibility penalty; Nationwide optimisation

Contents

1 Introduction

1.1 Research Background

Road freight transport is the backbone of modern logistics systems, ensuring timely and efficient delivery of goods across local, national, and international supply chains. However, this indispensable sector is also a major contributor to global greenhouse gas (GHG) emissions and air pollution. More than a quarter of global transport emissions are due to road freight: medium and heavy trucks alone emitted more than 1.7 billion tons of CO2 in 2021 [45]. Trucks account for approximately 29.4% of global transport-related emissions, ranking second only to passenger cars. Within international trade-related transport, road freight contributes around 53% of total CO_2 emissions, a share expected to increase further if no decisive mitigation strategies are implemented. In the European Union (EU), heavy-duty vehicles (HDVs) are responsible for more than a quarter of GHG emissions from road transport—and account for over 6% of total EU GHG emissions [15].

The decarbonization of freight transport is thus emerging as a critical policy and research agenda, with battery-electric trucks (BETs) considered one of the most promising solutions. Yet, the transition towards electric freight fleets is hindered by two interrelated challenges: limited driving range due to battery constraints and the lack of adequate charging infrastructure. While larger batteries can extend range, they substantially increase vehicle weight, cost, and energy demand—undesirable trade-offs for freight operations.

Two complementary infrastructure technologies have received increasing attention: static charging stations (SCS), where trucks charge during dwell times, and electric road systems (ERS), which allow dynamic charging while driving. Each technology has its advantages and limitations—SCS are cheaper to deploy but cause downtime and require large batteries, while ERS reduce battery needs and enable seamless operations but demand high upfront investments. Therefore, an integrated strategy that combines both technologies is essential to balance infrastructure cost, operational efficiency, and technological feasibility.

Despite the growing literature, most existing studies remain limited in two important ways. First, previous study often examine SCS or ERS in isolation, rather than in an integrated framework that reflects real-world deployment needs. However, as freight electrification requires both widespread accessibility for regional operations and continuous energy supply for long-haul corridors, the coordinated deployment of SCS and ERS is increasingly recognized as essential. For instance, Sun et al. (2020) proposed an integrated model for optimally planning both static and dynamic charging infrastructure, highlighting the value of coordinated deployment [40]. Moreover, Rogstadius et al. (2025) demonstrated through simulations that ERS inclusion can significantly accelerate freight electrification and reduce long-term GHG emissions, especially when static charging alone falls short[33].

Second, many rely on synthetic or trip-based data, which fail to capture the operational continuity and path dependencies of freight movements. In reality, freight operations are organised around tours—multi-stop chains covering complete missions—where energy demand and feasible charging opportunities depend on the full sequence of trips, not isolated OD pairs. Neglecting this dimension risks misallocating infrastructure and underestimating system costs. A recent review by Alam and Guo (2023) emphasizes the need for dedicated research on charging station planning and fleet operations tailored to electric freight vehicles, given their unique logistics patterns [2].

In summary, the challenges of freight decarbonization and the limitations of existing approaches call for a new modelling framework that can integrate infrastructure investment with operational feasibility. The next sections therefore set out the research objectives (Section 1.3) and research questions (Section 1.4) that guide this thesis.

1.2 Objectives

The overarching objective of this study is to develop a comprehensive optimization framework for the strategic deployment of charging infrastructure tailored to battery-electric trucks (BETs). Specifically, the study seeks to determine the optimal spatial distribution and configuration of SCS and ERS such that the energy demand of freight transport operations is reliably satisfied while the total system cost—comprising infrastructure investment, operational expenditure, and vehicle-related costs—is minimised.

To achieve this, the thesis pursues several interrelated objectives:

1. Model development: Establish a bi-level optimization model that jointly considers strategic siting and investment decisions at the upper level, and operational feasibility of freight tours (including energy balances, charging events, and battery constraints) at the lower level. This enables integrated planning that explicitly accounts for the interaction between infrastructure deployment and vehicle operations.
2. Data-driven design: Incorporate large-scale, real-world truck tour data as the primary model input. Unlike trip-based approaches, tour data preserve the sequential structure of freight movements, including dwell times and stop dependencies, thereby ensuring that energy demand and charging opportunities are realistically captured.
3. Technology integration: Evaluate the complementary roles of SCS and ERS within a single optimization framework. By exploring different deployment scenarios, the model aims to identify under what conditions static charging, dynamic charging, or a combination thereof is most cost-effective, scalable, and operationally feasible.
4. Methodological innovation: Implement a genetic algorithm tailored to the large-scale nature of the problem, ensuring computational tractability while preserving solution quality. Benchmark the Genetic Algorithm against exact MILP solutions in smaller cases to validate correctness.
5. Policy and planning insights: Provide empirically grounded evidence for policymakers, infrastructure planners, and industry stakeholders on how integrated charging strategies can accelerate freight decarbonization. The findings will contribute to identifying cost-effective infrastructure investment pathways that balance environmental benefits with economic and operational viability.

In sum, this research aims not only to propose a novel methodological framework but also to bridge the gap between theoretical optimization studies and real-world freight system planning. By explicitly combining SCS and ERS within a nationwide, tour-based optimization framework, the thesis aspires to inform both academic discourse and practical decision-making on the electrification of heavy-duty road freight.

1.3 Research questions

This research is driven by the following main research question:

> *How can a bi-level optimization framework, integrating SCS and ERS, be designed and applied to real-world truck tour data in order to minimise total system cost while ensuring operational feasibility of battery-electric freight transport?*

To address the main research question, the following subquestions are formulated:

1. What technological, economic, and operational factors need to be incorporated when modelling SCS and ERS deployment for heavy-duty battery-electric trucks?
2. Which optimization modelling approach (e.g., bi-level MILP, heuristic methods) is most suitable for nationwide infrastructure planning, and how should the parameters, decision variables, and constraints be structured to reflect both strategic siting and operational energy feasibility?
3. How can large-scale truck tour data be transformed and utilised as model input, and how does this differ from the use of conventional trip-based data?
4. What are the impacts of using tour-based data compared to trip-based data on optimization outcomes, infrastructure deployment patterns, and system cost estimation?

1.4 Approach

To address the main research question and its sub-questions, this study adopts a sequential, mixed-method research design that integrates literature synthesis, data preparation, model formulation, computational optimization, and result analysis. First, a systematic literature review identifies technological, economic, and operational factors relevant to planning SCS and ERS, and surveys modelling paradigms (bi-level MILP, heuristics, and simulation), thereby informing assumptions, decision variables, and constraints. Second, large-scale truck trajectory data are processed into tour- and trip-based inputs through tour identification, Dutch VAM zonal aggregation, and shortest-path reconstruction (542 nodes, 3,252 links). This step preserves operational continuity while providing tractable corridor-level inputs for optimization. Third, a bi-level optimization model is formulated: the upper level determines strategic siting of SCS, ERS deployment, and truck-level battery assignment, while the lower level enforces tour-level energy feasibility (SOC balance, static/dynamic charging, and penalty terms), explicitly linking infrastructure investment with vehicle operations. Fourth, a tailored genetic algorithm is designed for the national-scale problem (binary chromosome for SCS/ERS and battery classes), with corridor-aware crossover/mutation and layered fitness evaluation. Small instances are additionally solved via an exact MILP to verify correctness and solution quality. Fifth, results analysis quantifies system costs, infrastructure deployment, feasibility, and battery utilization, with sensitivity checks. Finally, the findings are synthesised into policy-relevant insights on cost-effective electrification pathways, highlighting when SCS, ERS, or hybrid strategies are preferable and how nationwide freight decarbonisation can be accelerated under realistic constraints.

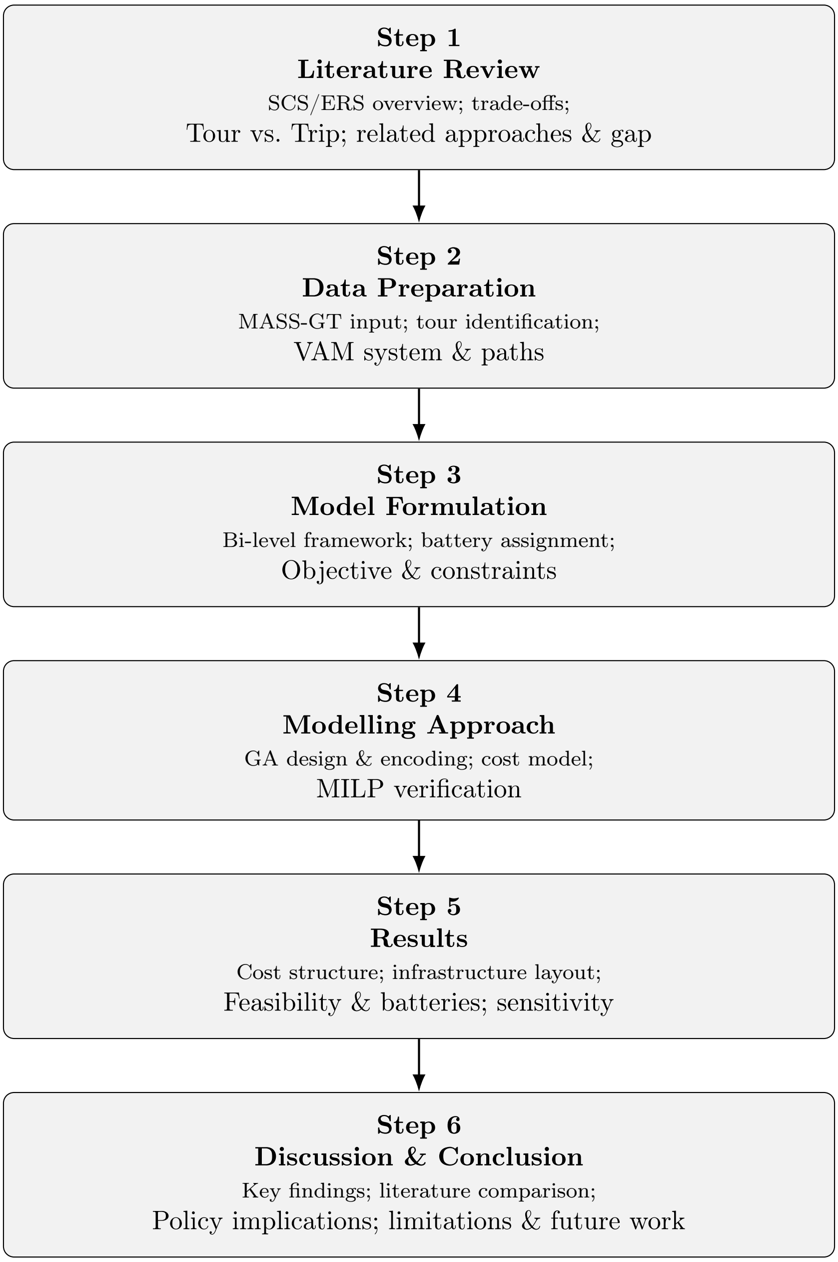


Figure 1: Overall research approach flowchart

As shown in Figure 1, the overall workflow proceeds sequentially through literature review, data preparation, model formulation, modelling approach, result analysis, and discussion with policy implications.

2 Literature Review

2.1 Overview of Static and Dynamic Charging Technologies in Heavy-Duty Road Freight

Research and practice have shown that charging technologies primarily fall into two categories: static charging SCS and ERS. SCS typically relies on high-power fast-charging systems but has stringent requirements for grid connectivity and on-site infrastructure. In contrast, ERS technologies (including overhead contact lines, ground rails, and wireless induction) can provide continuous power supply during vehicle operation, reducing reliance on battery capacity and the need for charging stops.

In terms of technological maturity, Piedel et al. (2024) noted in their review that static conductive technologies (such as pantographs) are relatively mature, while dynamic conductive technologies (such as overhead contact lines) have significant development potential; wireless (inductive) charging remains in the experimental stage, with efficiency and cost challenges persisting [30].

Earlier studies, such as Sun (2020), proposed an integrated planning model considering the interdependent reliance between transportation and power systems, exploring the optimal strategy for the joint deployment of SCS and ERS. This perspective aligns closely with the "joint layout optimization" objective in this study [40].

Furthermore, Rogstadius et al. (2025) used agent-based simulation to analyze the impact of policy incentives for ERS on the electrification of heavy-duty freight in a European context, finding that when electric road usage reaches a certain level, it serves as a 'no-regret investment' with low risk and high benefits for carbon reduction [33].

2.2 Electric Road System

Electric Road Systems (ERS) are commonly grouped into three technology families: overhead conductive systems using catenary contact lines (OCL), ground-level conductive rails (in-road or on-road), and inductive (wireless) power transfer. Multiple technical and policy assessments consistently find that OCL is the near-term, system-level least-cost option for high-volume truck corridors, provided sufficient traffic density, while ground-level conductive and inductive systems offer different trade-offs in visual impact, vehicle compatibility, and installation complexity [39].

Overhead conductive systems, such as the Siemens e-Highway trials [38], are among the most mature ERS technologies and enable high power delivery suitable for heavy-duty trucks, shown in figure 2. However, studies highlight significant drawbacks: high installation and maintenance costs, safety concerns (e.g., collisions, hindrance to emergency services), and limited accessibility for non-commercial vehicles [46].

Ground-level conductive rails—either embedded in or affixed to the road surface (e.g., Elways, Elonroad systems)—offer strong power transfer capabilities and broader vehicle compatibility [47]. Measurement-based analyses like the Swedish eRoadArlanda pilot report end-to-end charging efficiencies in the range of 82–89%, noting that rectifier placement and weather-induced leakage are key efficiency constraints [17].

Inductive (wireless) systems utilize dynamic wireless power transfer through coils embedded beneath the road surface [4][46]. While offering installation flexibility and eliminating overhead infrastructure, they typically deliver lower power (25–40 kW) and exhibit lower energy transfer efficiency [48][23]. Pilot studies (e.g., Electreon) show energy collection around 64%, but emphasize high implementation complexity and blockage of underlying infrastructure [23][46].

Comparative technoeconomic assessments (e.g., Sweden's national electric road program) estimate initial infrastructure costs per two-lane kilometer as approximately: overhead wires USD 1.1 million, in-road rail ŨSD 0.7 million, and inductive coils USD 2.2 million [48]. In broader lifecycle cost analysis, conductive rails appear most cost-effective in reducing battery size and societal energy system costs; inductive solutions, though feasible, impose higher infrastructure burdens [48][36].

In summary, overhead systems offer high power and maturity but at elevated cost and limited accessibility; ground-level conductive systems balance performance and applicability with moderate cost and better efficiency; wireless inductive systems are flexible but currently hindered by lower efficiency and higher infrastructure complexity. Such trade-offs suggest the necessity for detailed comparative analysis in freight-specific scenarios, an area under-explored in current literature.

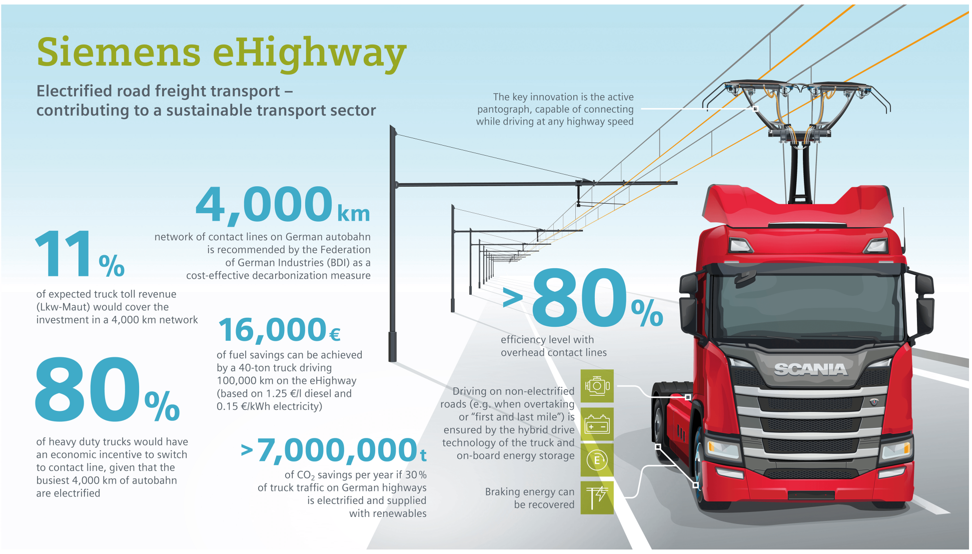


Figure 2: Electrified road freight traffic – the eHighway by Siemens [38]

2.3 Comparative Perspectives on ERS and Conventional Charging

In recent years, several studies have directly compared static charging stations (SCS) and electric road systems (ERS) to evaluate trade-offs in infrastructure cost, battery requirements, and system-wide performance. Rogstadius et al. (2025) modeled a 6,000 km ERS network and showed that, compare to traditional charging network, battery consumption per vehicle could be reduced by about 40%, while cost-minimizing battery capacity could drop by as much as 70% when dynamic charging directly powers propulsion [33]. Similarly, Shoman et al. (2022), using real-world driving data from Sweden, found that combining ERS with home charging reduced the required battery range by 62–71%, and that cost savings from smaller batteries could outweigh the additional infrastructure costs of ERS [36]. From an economic perspective, a Dutch study (Decisio, 2022) concluded that ERS becomes cost-effective only under high daily freight distances (above 180–300 km), identifying a breakeven point that links infrastructure investment to traffic density. At the system level [9], Olovsson et al. (2021) compared the electricity impacts of SCS and ERS in Sweden and Germany, showing that large-scale ERS deployment raises peak grid load and requires complementary renewable and storage capacity, while optimized static charging can smooth demand and better align with renewable generation [29].

Overall, the literature indicates that ERS can significantly reduce battery-related costs and improve operational efficiency, but only under traffic conditions that justify the infrastructure investment, and with additional measures to mitigate grid stress. In contrast, SCS is less capital-intensive and more grid-friendly, though it relies on larger vehicle batteries. A clear research gap remains in freight-specific applications, particularly in evaluating how hybrid deployment of SCS and ERS could balance infrastructure cost, operational efficiency, and system integration.

2.4 Tour data versus Trip data

The development of tour-based freight models can be traced back to the late 1990s. One of the earliest frameworks was the GoodTrip model by Boerkamps and van Binsbergen (1999), which simulated multi-stop urban goods distribution by linking consumer, wholesale, and warehouse nodes into closed vehicle chains. This work demonstrated how alternative urban delivery policies affect vehicle-kilometres travelled and emissions [3].

In the 2000s and early 2010s, further theoretical advances were made to analytically capture tour formation. Holguín-Veras and colleagues introduced entropy-maximization formulations for tour-flow estimation, arguing that closed tours could be inferred at the aggregate level under principles of consistency and conservation, thereby laying the foundation for analytical approaches to tour modelling [34].

More recently, Gonzalez-Calderon and Holguín-Veras (2019) extended this line of research by proposing an entropy-based freight tour synthesis method. Their study showed how tours can be reconstructed from traffic count data, and highlighted that the sampling design strongly affects the accuracy of synthesized tours [16].

A 2017 synthesis by Oak Ridge National Laboratory (ORNL) further compared trip- and tour-based frameworks, concluding that tour-based models better capture stop sequences, dwell times, and driver constraints than the conventional four-step trip-based approach, albeit at the cost of greater reliance on high-frequency GPS data and establishment surveys [25].

A detailed comparison of trip-based and tour-based freight modeling approaches is provided in Table 1 below.

Table 1: Comparison of Trip-based Four-Step and Tour-based Freight Models

Dimension	**Trip-based Four-Step Model**	**Tour-based Model**
Basic unit	OD trip	Vehicle tour (sequence of stops within a duty cycle)
Data requirement	Freight flows / zone-level production–attraction	GPS traces or operation logs at stop level
Applicable scenarios	Long-haul OD flows; strategic national forecasting	Urban freight distribution; regional multi-stop deliveries; reverse logistics (returns, recycling, and backhauls)
Pros & Cons	Simple and fast to implement, but ignores tour-chaining and stop sequence decisions	Captures stop sequence, dwell time, and driver constraints, but requires richer data and more parameters

Unlike trip-based data, tour data preserve the full sequence of stops, dwell times, and distances that a single truck covers within one duty cycle (typically a working day). This is essential for charging-infrastructure planning for three reasons:

1. Complete energy budget – Battery depletion and re-charging needs accumulate over the entire mission. Only a tour representation captures the true end-to-end distance and the timing of consecutive legs, preventing under- or over-estimation of required on-route energy supply.

2. Feasible charging windows – Stop locations and dwell durations embedded in a tour reveal when a truck is physically able to plug in (static DCFC) or merge onto an ERS segment. Trip-only data lack this temporal linkage and can misplace chargers at nodes that trucks never visit in sequence.

3. Infrastructure interoperability – Tours expose corridor overlap among multiple carriers: different trips that appear unrelated at the OD level may in fact share the same highway segments once chained in a tour. This information helps prioritise ERS deployment on high-utilization corridors and avoids redundant station placement.

Consequently, the optimization model relies on tour-level inputs to ensure that every complete mission can be executed without range anxiety while minimising total infrastructure cost. Trip-based inputs would break this continuity and risk infeasible or sub-optimal charging layouts.

2.5 Related Modelling approach

In the field of transportation and energy facility siting, bi-level optimization models are widely used due to their structural and decision-making flexibility. These models typically include:

Upper level: Strategic siting and investment decisions to determine facility layout plans.

Lower level: Operational scheduling and route feasibility analysis to evaluate operational performance and system response after strategy implementation.

This modeling structure has been widely applied in facility location and routing problems (Location–Routing Problem, LRP) and electric vehicle charging infrastructure location problems (EV Charging Station Location Problem, EVCSLP). It enables strategic decision-making and operational analysis to be closely integrated, thereby enhancing the overall optimization of the system [49].

Liao et al. (2024) developed a bi-objective optimization model to determine ERS deployment strategies across highway networks in four European countries. Their framework minimizes both infrastructure investment and transport operating costs, and the case studies quantify how ERS deployment affects battery capacity requirements and vehicle life-cycle performance under different network configurations [24].

A study based on an integrated simulation model evaluated the battery and charging requirements of medium and heavy-duty electric trucks in the M180 ERS demonstration section in the UK [8]. The model incorporates driving paths, terrain, and charging infrastructure to dynamically simulate vehicle energy consumption and battery status. By simulating various transportation scenarios, three vehicle configurations—small battery type, medium battery type, and range extension type—along with corresponding charging strategies were proposed, offering a technical foundation for vehicle selection and infrastructure planning within the ERS system.

Seilabi et al. (2025) proposed a two-layer framework in the context of sustainable transportation: the upper layer optimizes the location and capacity allocation of charging stations; the lower layer simulates users' travel and energy choice decisions, with the aim of reducing CO_2 emissions. The solution is based on genetic algorithms, and the model's potential for application in real-world networks is verified [10].

Kunawong et al. (2025) established a complete two-layer MILP model for long-distance bus networks: the upper layer determines the location and number of charging stations, while the lower layer optimizes the configuration of off-site chargers and charging scheduling to maximize the operational efficiency of the system. Sensitivity analysis was conducted on this model using real-world cases in Thailand [22].

An evaluation based on the MOSTACHI model (Model for Optimization and Simulation of Traffic and Charging Infrastructure) assessed the role of ERS in the decarbonization of heavy transport in Europe. The findings suggest that with policy mandates or economic incentives, ERS can achieve high utilization rates, thereby reducing system costs and accelerating the electrification process, while also enabling significant reductions in greenhouse gas emissions by 2030 [32].

Inez (2024) formulates the nationwide planning of charging infrastructure as a bi-level mixed-integer linear program (MILP). At the upper level, strategic decisions are made on the siting of static charging stations (SCS) and the deployment of electric road system (ERS) segments, with the objective of minimizing annualized investment (CAPEX) subject to budget and technical constraints. The lower level represents the operational feasibility of OD trips given the chosen infrastructure, evaluating operational expenditures (OPEX) including driver time, electricity tariffs, and tolls. Feasibility is ensured through explicit state-of-charge (SOC) trajectories, battery capacity bounds, and charging opportunity constraints. The two layers are coupled by requiring that only layouts that yield feasible and cost-efficient operations at the lower level are acceptable at the upper level. The model is implemented in Python and solved using the commercial solver Gurobi, which provides exact MILP solutions after appropriate linearization of nonlinear terms. While the thesis discusses heuristic alternatives for large-scale instances, the methodological emphasis is on the exact MILP construction and solution process, including initialization with shortest-path calculations, infrastructure allocation at the upper level, operational verification at the lower level, and iterative convergence. [19]

As summarized in Table 2, prior studies demonstrate the value of bi-level and heuristic optimization for electrification planning, but rely mostly on trip-based data or focus on single technologies in isolation. This thesis builds on Inez (2024) by extending the framework to tour-based data, introducing penalty terms for infeasible tours, and scaling the heuristic to solve nationwide problems with over 1.5 million tours.

Table 2: Overview of modelling approaches in freight electrification

Study	Focus / Data Basis	Methodological Approach	Key Findings / Limitations
Liao et al. (2024)	European corridors	Bi-objective MILP	ERS reduces battery size (370→90 kWh); large-scale cost trade-off
De et al. (2023)	UK M180 demo	Integrated simulation	Proposed three truck configs; context-specific insights
Seilabi et al. (2025)	Sustainable EV networks	GA-based bi-level	Verified GA feasibility, but limited to passenger EVs
Kunawong et al. (2025)	Long-distance bus networks	Two-layer MILP	Charger siting + scheduling; validated with Thailand data
Rogstadius et al. (2023)	Europe-wide freight	MOSTACHI system model	ERS reduces GHG; policy support crucial for utilization
Inez (2024)	Netherlands freight (synthetic trips)	Bi-level MILP (solved with Gurobi)	Formulated integrated SCS–ERS optimization; validated with exact solver, but scalability limited by trip-based OD data
This Thesis (2025)	Netherlands freight (1.5M tours, MASS-GT)	Bi-level MILP + GA heuristic	First nationwide tour-based optimization of hybrid SCS–ERS; scalable and empirically grounded

2.6 Summary of Literature Review

The literature highlights that freight electrification relies on two main infrastructure paradigms: static charging stations (SCS) and electric road systems (ERS). Section 2.1 introduced these technologies, noting their complementary strengths and weaknesses. While SCS are cheaper and widely deployable, they require dwell times and larger batteries. ERS, by contrast, reduce battery size requirements and support seamless long-haul operations, but involve high capital costs and grid integration challenges. Section 2.2 compared ERS technologies, showing that overhead conductive systems are the most mature, ground-level conductive rails offer a balance of efficiency and cost, and inductive charging remains experimental with significant efficiency barriers.

Section 2.3 examined comparative perspectives on SCS and ERS, revealing that ERS can substantially reduce vehicle battery sizes and operational costs under dense traffic conditions, whereas SCS remain cost-effective and grid-friendly in most regional contexts. The evidence consistently suggests that hybrid strategies, combining SCS for widespread accessibility and ERS for corridor continuity, hold the greatest promise.

Section 2.4 highlighted the methodological importance of adopting tour-based rather than trip-based models. Unlike trip data, tour data capture chained stops, dwell times, and daily duty cycles, enabling realistic assessment of energy demand and feasible charging opportunities. Recent advances in Freight Tour Synthesis, entropy-based demand models, and simulation-based calibration show that tour data are increasingly adopted in both research and practice, including public-sector freight models in the US and the Netherlands.

Section 2.5 reviewed related modelling approaches, focusing on bi-level frameworks that integrate strategic siting with operational feasibility. Studies by Liao et al. (2024), De Saxe et al. (2023), Seilabi et al. (2025), Kunawong et al. (2025), and Rogstadius et al. (2023) demonstrate the value of combining infrastructure planning with vehicle operations, yet most rely on synthetic or trip-based data, or focus on a single technology. Inez (2024) advanced this line of work by formulating a nationwide bi-level MILP for SCS–ERS planning in the Netherlands, solved with Gurobi, but her reliance on OD-trip data limited operational realism and scalability.

In summary, the literature converges on three insights: (i) both SCS and ERS are necessary and complementary for freight electrification; (ii) tour-based data provide a more realistic foundation for infrastructure planning than trip-based inputs; and (iii) bi-level optimization offers a suitable methodological framework, but has so far been applied only at limited scales or with synthetic data. These gaps motivate the present thesis, which develops the first nationwide tour-based bi-level optimization model for hybrid SCS–ERS deployment in the Netherlands, solved with a scalable GA heuristic and validated against exact MILP benchmarks.

Beyond academic contributions, the study also generates practical insights of direct relevance to policymakers and infrastructure planners. The results provide quantitative evidence on cost structures, deployment patterns, and feasibility distributions across regions, highlighting the complementary roles of SCS and ERS and the potential of infrastructure deployment to reduce reliance on oversized batteries. These findings align with the objectives of the EU Green Deal, the Fit for 55 package, and the forthcoming Dutch truck-kilometre charge, offering timely input for shaping investment strategies and policy interventions.

3 Data

3.1 Data Sources and Introduction

The data used in this study originates from the MASS-GT project, an agent-based freight transport model that generates micro-level trip records through automated reporting from Transportation Management Systems (TMS). Compared with survey-based data collection, MASS-GT offers far higher density and completeness of observations. The definition and formation mechanism of tours follow the descriptive model of Thoen et al. (2020), which incrementally allocates shipments using a random utility framework and clearly specifies the start and end boundaries of each tour, including empty return legs [41].

The dataset provides a rich set of attributes at multiple levels. Each record contains hierarchical identifiers for carriers, tours, and trips, along with zonal references under different Dutch systems (NRM, BG, VAM). Temporal information (departure and arrival times), spatial coordinates (Rijksdriehoekstelsel, RD), commodity classifications (NSTR codes), vehicle types, shipment weights, and CO_2 emissions are also included. Table 3 summarizes the full range of available fields. For modelling purposes, however, only a subset is required: tour and trip identifiers to preserve sequential order, zonal and coordinate fields for embedding into the VAM network, time attributes for detecting charging windows, and distance and emission measures for energy and penalty calculations. These variables provide the essential inputs for constructing the corridor-based optimization framework developed in Chapter 4.

Table 3: Tour dataset variables and their descriptions

Field	Description
carrier_id	Identifier for carrier
tour_id	Tour identifier
trip_id	Trip identifier
origin_nrm	Origin zone NRM
destination_nrm	Destination zone NRM
origin_bg	Origin region BG
destination_bg	Destination region BG
origin_vam	Origin zone VAM
destination_vam	Destination zone VAM
x_rd_origin__meter	X-coordinate of origin (RD meters)
x_rd_destination__meter	X-coordinate of destination (RD meters)
y_rd_origin__meter	Y-coordinate of origin (RD meters)
y_rd_destination__meter	Y-coordinate of destination (RD meters)
vehicle_type_lwm	Vehicle type (light/medium/heavy)
nstr	NSTR commodity classification
commodity	Commodity description
logistic_segment	Logistic segment type
n_shipments	Number of shipments
dc_id	Distribution center identifier
tour_weight__ton	Total weight of tour (tons)
trip_weight__ton	Weight of trip (tons)
tour_deptime__hour	Departure time of tour (hour)
trip_deptime__hour	Departure time of trip (hour)
trip_arrtime__hour	Arrival time of trip (hour)
combustion_type	Vehicle combustion type
external_zone_bg	External zone identifier (BG system)
container	Container indicator
co2__gram	CO_2 emissions (grams)
co2_nl__gram	CO_2 emissions within NL (grams)
distance__kmeter	Trip distance (km)

distance_nl__kmeter	Distance within NL (km)

For the purpose of modelling the optimization tour-based model, only a subset of the available fields in the tour dataset is required.

First, tour identifiers (`carrier_id`, `tour_id`, `trip_id`) are essential to reconstruct the sequence of trips that form each complete tour, thereby preserving the stop order and path dependency of freight operations.

Second, spatial information is needed to embed the tours into the network. We rely on zonal representations (`origin_vam`, `destination_vam`) and corresponding RD coordinates (`x_rd_origin_meter`, `y_rd_origin_meter`, etc.), which are aggregated into the VAM corridor network. This enables shared infrastructure planning at the corridor level rather than at individual depots.

Third, temporal attributes (`tour_deptime__hour`, `trip_deptime__hour`, `trip_arrtime__hour`) allow us to capture feasible charging windows along a tour, and to approximate tour duration for energy budgeting.

Fourth, distance and energy-related fields (`distance__kmeter`, `distance_nl__kmeter`, `co2__gram`) provide the basis for estimating energy consumption, emission levels, and the corresponding cost components. Energy demand in the model is derived from distance multiplied by a uniform consumption rate, while CO_2 emissions are used for penalty terms when tours are not served by electric trucks.

Finally, vehicle and commodity information (`vehicle_type_lwm`, `nstr`, `commodity`, `tour_weight__ton`) are included to ensure that the dataset aligns with realistic truck operations, even though the optimization model itself abstracts away from detailed vehicle heterogeneity.

Together, these variables supply the core inputs for constructing the tour-based energy feasibility checks and the investment–operation trade-off embedded in the bi-level optimization model.

3.2 Tour Data and Identification

Definitions. A *trip* is a single origin–destination movement of a truck between two stops. A *tour* is an ordered chain of trips executed by the same vehicle within one duty cycle (typically a working day), possibly including deadheading legs (empty moves) and returning to the depot.

Unique identifier and ordering. We construct each tour by grouping records with the same `carrier_id` and `tour_id` and sorting constituent trips by `trip_deptime__hour` (ascending). The concatenation $<$ *carrier_id* $>$ _<tour_id> serves as a unique tour key (e.g., `981_285`), which we use throughout the paper for traceability across preprocessing, modelling, and results.

Path geometry disclaimer. The raw dataset provides only trip endpoints (origins and destinations) but no intermediate path geometry. Hence, Figure 3 visualises each leg as a straight line between its endpoints for illustration. The actual corridor-level routes used by the model are reconstructed later on the VAM network via shortest paths.

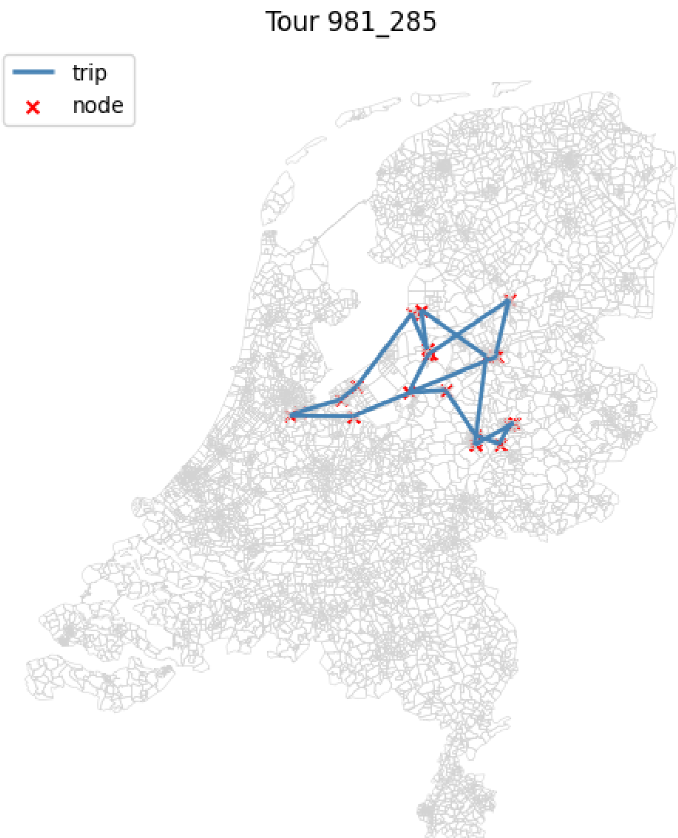


Figure 3: An example tour demo

Sanity checks and cleaning rules. Before using tours as model inputs, we apply the following checks:

- Field completeness: required fields in Table 4 must be present and non-null.
- Temporal order: for each trip, `trip_deptime__hour` < `trip_arrtime__hour`; tour trips are strictly ordered by departure time.
- Leg continuity: after zonal aggregation, the destination zone of leg k should match the origin zone of leg $k+1$; otherwise records are flagged and corrected or removed.
- Distance plausibility: remove zero/negative distances; winsorise extreme outliers (e.g., top 0.1%) if caused by geocoding errors.
- External zones: trips involving external BG/VAM zones are either excluded or truncated at the national boundary for consistency with the Dutch network analysis (document the chosen policy).

Link to model inputs. For each tour $t \in T$, we retain an ordered set of segments K_t with their origin/destination VAM zones, segment distances, and timestamps. These become the basis for shortest-path reconstruction, energy accounting, charging-window detection, and ultimately the operational feasibility constraints.

Table 4: Minimum fields required to construct tours and feed the model

Field	Type	Role
carrier_id, tour_id, trip_id	keys	Grouping (tour), ordering (trip), traceability
origin_vam, destination_vam	categorical	Zonal embedding for corridor network
distance__kmeter	numeric	Segment length for energy and cost
trip_deptime__hour, trip_arrtime__hour	numeric	Temporal ordering, charging windows
x_rd_origin__meter, y_rd_origin__meter	numeric	(Optional) spatial QA / map rendering
x_rd_destination__meter, y_rd_destination__meter	numeric	(Optional) spatial QA / map rendering

3.3 Truck Identification and Tour Aggregation

Motivation

The raw trajectory logs are available at the *trip/tour* level and contain no native truck identifier. In reality, however, a single physical vehicle executes *multiple* tours over the planning horizon. If we naïvely treated each tour as a distinct vehicle, the upper level would have to purchase a battery for *every* tour, which massively inflates the number of vehicles and leads to an unrealistic, order-of-magnitude overestimation of battery CAPEX. It would also distort infrastructure siting (SCS/ERS) and bias the optimiser toward outsourcing, because buying one battery per tour is far more expensive than the per-tour outsourcing penalty.

To align the data with operational reality and with our model design in Chapter 4 (battery assignment is made *per truck*, fixed across all its tours), we therefore construct a consistent vehicle identifier `truck_id`. Tours are first grouped by carrier and then partitioned into plausible vehicle bundles so that each bundle represents the set of tours that could reasonably be served by one truck. This enables (i) realistic battery investment—one battery per truck rather than per tour, (ii) consistent lower-level evaluation where all tours of the same vehicle inherit the same battery capacity and initial SOC, and (iii) a fair comparison between investing in batteries/infrastructure versus outsourcing under the same accounting boundary.

Overall procedure

The pipeline comprises four steps:

- S1. Carrier-anchored grouping: group all tours by `carrier_id` so that tours from different carriers are never mixed.
- S2. Feature-based clustering within each carrier: within each carrier group, compute feature vectors for all tours (features listed in Table 5) and cluster tours by feature similarity. Each cluster represents a set of tours that could plausibly be served by the same physical vehicle.
- S3. Cluster-to-vehicle mapping: assign a globally unique `truck_id` to each cluster and label all tours in that cluster with this `truck_id`, establishing a hierarchy carrier → truck_id → tours.
- S4. Post-hoc consistency checks: refine the initial assignment by enforcing temporal non-overlap and geographic continuity rules (see below). Conflicting tours are reassigned to the nearest feasible `truck_id` within the same carrier.

Features used for clustering

We adopt operationally meaningful features so that tours attracted into the same cluster are realistically executable by a single vehicle. Table 5 summarises the features and their roles; no calibrated parameter values are required for understanding the logic.

Table 5: Features for carrier-anchored tour clustering and their roles in truck_id generation

Feature	**Construction**	**Rationale**
Carrier identity	`carrier_id`	Hard boundary: tours from different carriers are never assigned to the same vehicle.
Vehicle class	`vehicle_type_lwm`	Keeps tours requiring the same vehicle specification together; avoids infeasible reassignments across classes.
Departure time window	`tour_deptime__hour`	Encourages tours with similar duty periods to be grouped, improving schedule realism and avoiding systematic time conflicts.
Origin & destination zones	`origin_vam`, `destination_vam`	Promotes spatial cohesion so that successive tours of the same truck are geographically plausible.
Deadhead proximity	`distance__kmeter`	Favors clusters where inter-tour repositioning is short, improving operational continuity.
Tour duration proxy	`trip_arrtime__hour`	Groups tours with comparable lengths to balance per-vehicle workload.
Load category (if present)	categorical load band	Prevents mixing tours with incompatible capacity requirements under one vehicle.

Consistency rules (post-processing)

After initial clustering, we enforce two minimal feasibility screens within each `truck_id`:

- Temporal non-overlap: tours assigned to the same `truck_id` must not overlap in time. Overlaps trigger a local reassignment within the carrier.
- Geographic continuity: the end of a tour and the start of its successor (under the same `truck_id`) must be reachable within the available turnover interval; otherwise the successor tour is reassigned.

These light-weight screens stabilise the mapping without requiring detailed scheduling.

Link to battery assignment

The resulting mapping provides a one-to-many relationship from `truck_id` to tours. In all subsequent modelling, each `truck_id` is bound to exactly one battery capacity class (upper-level decision), and *all* tours of that `truck_id` inherit the same battery configuration during operational evaluation.

Empirical distribution of tours per truck

Figure 4 depicts the empirical distribution of the number of tours associated with each `truck_id`. Most vehicles undertake only a few tours, with a long tail of higher-utilisation vehicles. This pattern supports our clustering design: the bulk of vehicles form compact tour bundles, while the tail captures intensive-use trucks that are particularly informative for battery sizing and infrastructure coverage.

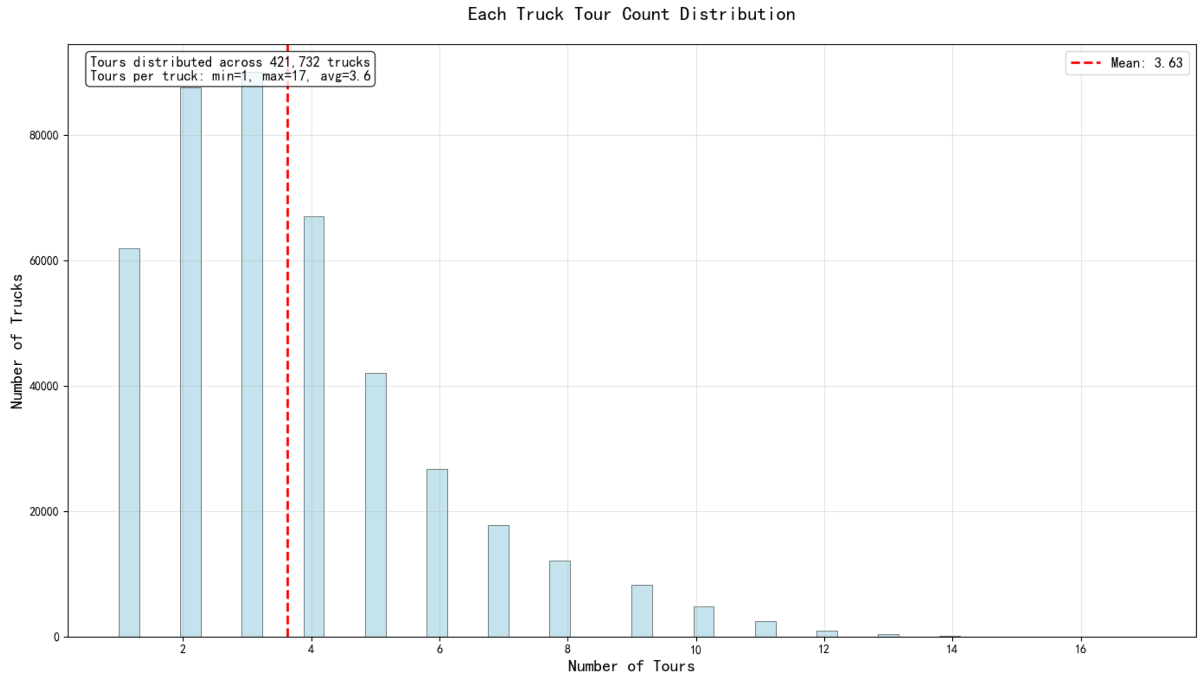


Figure 4: Tours-per-truck distribution

3.4 NL Zonal System

The zonal representation adopted in this study builds on the Dutch national freight modelling framework. BasGoed (the Basic Freight Model) is the official strategic freight transport model of the Netherlands and has been widely used for forecasting freight volumes by road, rail, and inland waterways [?]. Within the broader Dutch modelling system, the National/Regional Models (NRM) are mainly designed for passenger flows, but share compatible input data and modelling structures with BasGoed [21]. For freight-specific applications, the VrachtAutoMatrix (VAM) provides a dedicated OD matrix for medium- and heavy-duty trucks, serving as an important data element in BasGoed. The VAM is generated from the Basic Freight File (BBGV) and is designed to support freight demand modelling at a national scale. Since the 2014 revision, light-duty vehicles (bestelauto's, $<$1.5 tons) have been excluded from the VAM to enhance consistency and applicability for freight analysis [7]. Given its freight orientation and nationwide coverage, the VAM zonal system is adopted in this study as the spatial framework for data aggregation and corridor-level analysis.

After organizing the detailed trip data into tours, the dataset can be conceptualized as a directed graph of nodes (origin/destination points) and arcs (trip segments) with fixed tour-level demands. However, such node-level granularity is too fine for practical planning purposes. Selecting a specific OD node (e.g. a particular warehouse or depot) as a charging location would likely only serve that individual tour, since other tours — even those starting nearby — do not share that exact node in their routes (i.e. there is a lack of shared routing at the node level). Moreover, the trip-level data contain no explicit path information: two tours with geographically close OD points might in reality travel along the same highway corridor, but this overlap remains invisible when considering only the disaggregate nodes and trips. This is particularly problematic for planning shared corridor infrastructure such as ERS, where identifying and exploiting common routes is crucial. To address these issues, we aggregate the flows to a higher spatial level using the 542 VAM zones. The geo-data frame also provides the area of each zone, which in RD coordinates corresponds to a cropping window of approximately 150,000–250,000 m east longitude and 450,000–550,000 m north latitude. Figure 5 shows the spatial distribution of the VAM zones.

In this zonal representation, each trip is assigned to an origin and destination VAM zone, and all freight flow between a given pair of zones is assumed to follow a common representative route on the network (essentially capturing the main corridor between those regions). Furthermore, we assume that if a charging station or ERS segment is present within a zone, then all trucks passing through that zone can potentially access it – reflecting the idea that the zone's network is covered by the infrastructure. We adopt the VAM zoning scheme in lieu of finer-grained zones (such as the detailed NRM zones) because it offers a far more tractable model size and better corridor-level generalizability. By reducing the problem to a manageable number of zone-to-zone flows, the model remains computationally feasible, and the key shared-corridor patterns are preserved. In summary, the VAM-based region aggregation provides a practical balance between detail and abstraction, ensuring that major freight corridors and overlapping routes are represented for infrastructure planning while avoiding the complexity and limited insight of modelling thousands of individual trip nodes.

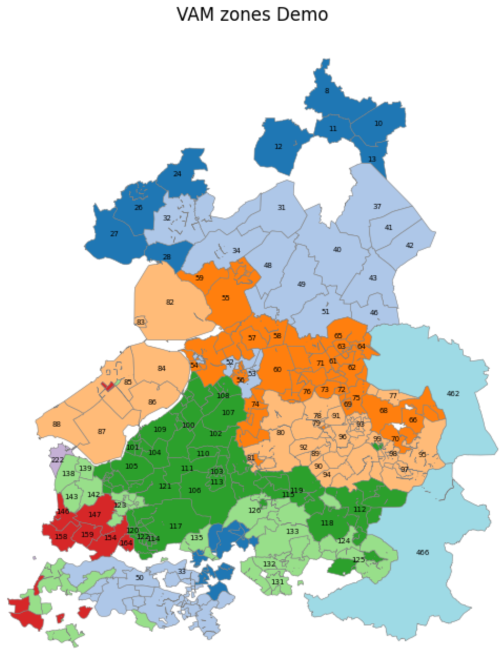


Figure 5: VAM Zone Demo

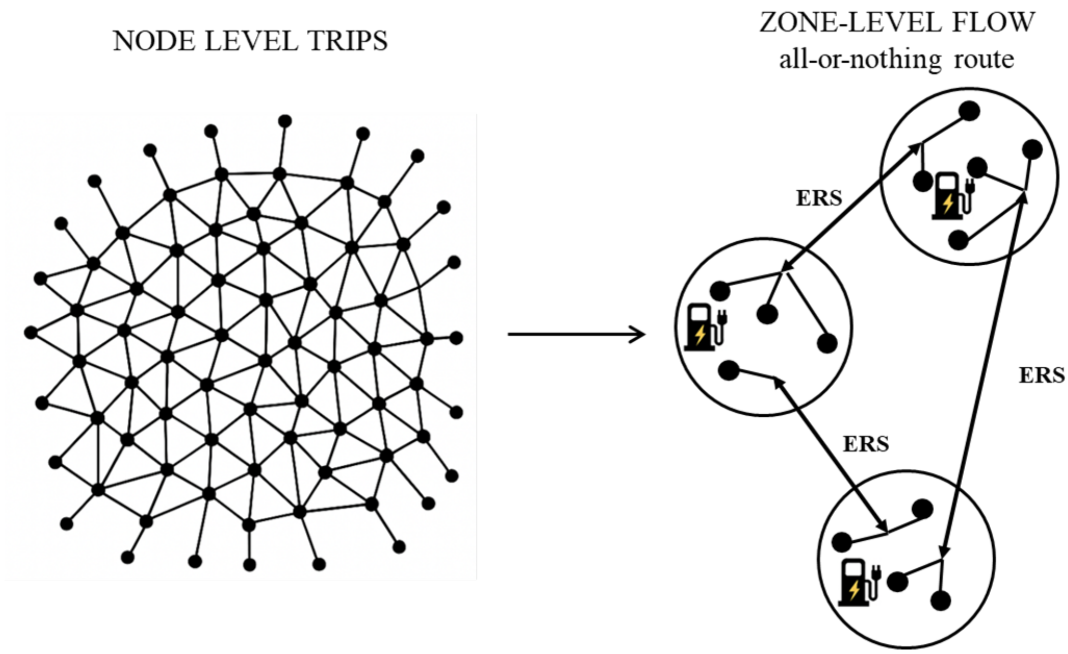


Figure 6: Node-level trips vs. zone-level flows

As illustrated in Figure 6, the original freight dataset comprises thousands of node-level trips (left), where every origin–destination pair is represented explicitly in a dense graph. For modelling purposes, these granular flows are aggregated into a handful of VAM zones (right). Within each zone, individual stops are retained (small black dots), but all traffic between two zones is loaded onto a single all-or-nothing corridor (bold line). Dynamic charging infrastructure (ERS) is then positioned directly on this corridor, while any static charging station located inside a zone is assumed to serve every tour that traverses that zone. The schematic thus highlights how detailed trip data are condensed into corridor-level flows without losing the key interaction between freight demand and charging coverage.

3.5 Data Processing

3.5.1 VAM Corridor Network Construction (Delaunay Triangulation)

To obtain a sparse yet well-connected inter-zonal corridor graph consistent with infrastructure siting, we construct links between VAM zone centroids using a Delaunay triangulation of all nodes (RD New, EPSG:28992). The triangulation is computed over the planar coordinates of the 542 VAM nodes, and unique undirected edges are extracted from triangle sides to form the network. This yields a connected graph with 1613 links and 542 nodes; link lengths are the Euclidean distances between zone centroids and serve as the basis for energy and toll calculations in the model. Candidate SCS are defined on nodes, and candidate ERS are defined on links.

Implementation details.

- Input and triangulation: Let N be the set of VAM nodes with coordinates (x_n, y_n). We compute a Delaunay triangulation on $\{(x_n, y_n) : n \in N\}$ and extract all triangle sides as adjacency relations. This choice maximises the minimum angle and provides natural geometric neighbours at $O(|N| \log |N|)$ complexity.
- Edge set and IDs: Each undirected edge is stored once using a canonical ID `u|v` with $u < v$ to avoid duplicates. The resulting link table `vam_link.csv` contains `from_node`, `to_node`, and `distance_km`; these distances are the corridor lengths used by the solver.
- Length distribution: Links are predominantly short-to-medium range ($<$ 10 km and 10–30 km buckets dominate), with a small tail of long edges ($\geq$ 100 km) connecting coastal/outlying zones and preserving single-component connectivity; see Fig. 7.
- Rationale: Compared with k-NN or radius graphs, Delaunay naturally encodes proximate neighbours, avoids unrealistic crisscrossing, and retains full connectivity with a compact number of links—well suited for corridor-level planning where ERS is deployed on links and SCS on nodes.

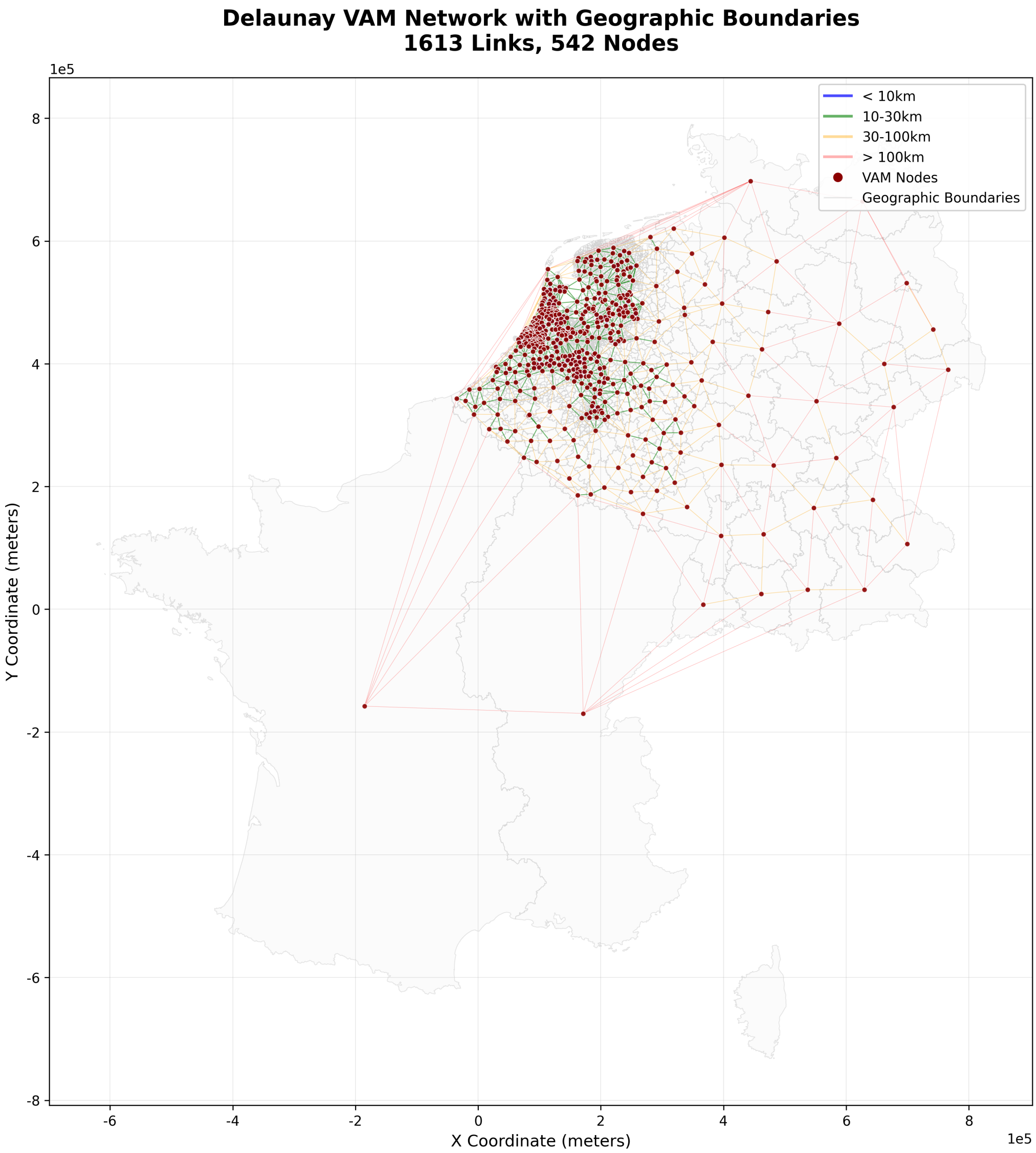


Figure 7: Delaunay-based VAM network

3.5.2 Shortest-path reconstruction on the corridor graph

In the raw truck–tour dataset, many trips connect origins and destinations that are *not* adjacent VAM zones. To represent such movements on the corridor network defined in Section 3.5.1, each trip is re-routed along the network *shortest path* between its origin and destination zone centroids. Paths are computed with Dijkstra's algorithm using link lengths (`distance_km`) as weights, yielding an ordered list of nodes and inter-zonal links for every trip. For a tour, the trip-level paths are concatenated and consecutive duplicates are removed, resulting in a clean zone-by-zone representation that is compatible with node-based SCS siting and link-based ERS deployment.

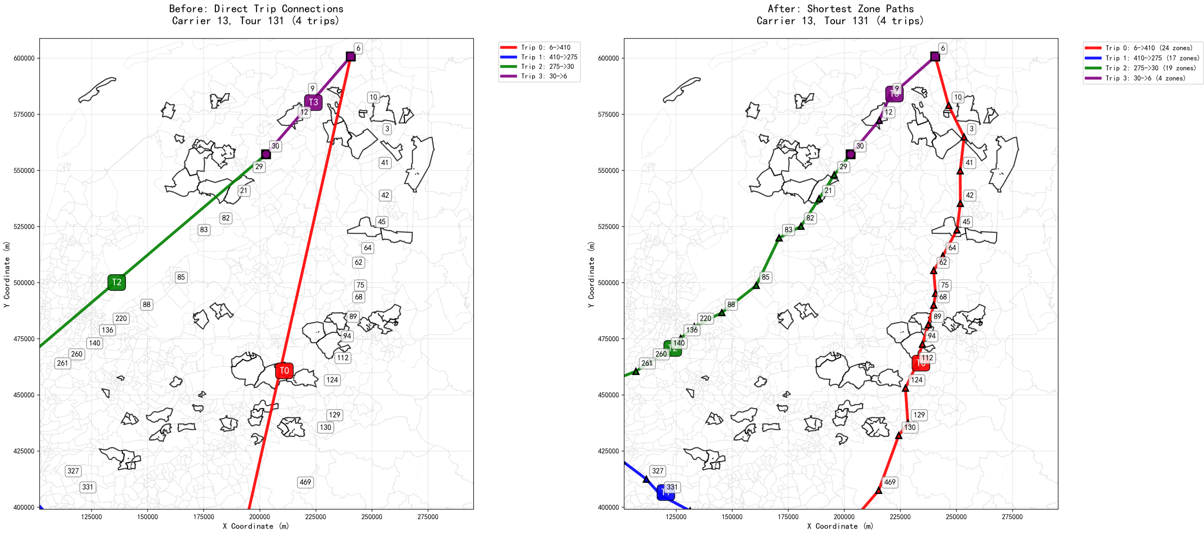


Figure 8: Tour zone-path example (carrier 13, tour 131)

A comparison between reconstructed VAM path lengths and recorded trip distances shows the expected pattern: deviations can be noticeable for short movements (intra-zonal or single-hop inter-zonal), while multi-zone trips are much closer in aggregate. To avoid metric inconsistency between operations and infrastructure placement, the optimization model *consistently uses the reconstructed VAM path length* (sum of link lengths along the shortest path) for energy and toll calculations. The original distances are retained only for diagnostics and descriptive statistics. This modelling choice keeps corridor costs aligned with the graph where decisions are made—SCS at nodes and ERS on inter-zonal links—and ensures that route energy, charging opportunities, and infrastructure usage are evaluated on a common, graph-consistent basis.

3.6 Conclusion

This chapter transformed raw trajectory logs into graph-consistent inputs for the nationwide optimisation. First, we constructed vehicle identities by grouping tours within carriers and partitioning them via feature-based clustering, followed by temporal and geographic consistency checks; the resulting `truck_id` provides the key index through which all tours of a vehicle can later share one battery configuration (Section 3.3). Second, we built a corridor network over VAM zone centroids using Delaunay triangulation, storing unique undirected links with canonical IDs and lengths; nodes act as SCS candidates and links as ERS candidates (Section 3.5.1). Third, we re-routed every trip along shortest paths (Dijkstra) on this network and concatenated trip paths into tour-level zone sequences (Section 3.5.2). To keep infrastructure decisions and operations on a common metric, all energy and toll calculations in subsequent chapters use the reconstructed VAM path lengths; original distances are retained only for diagnostics.

The data pipeline outputs the following artefacts for the model layer:

- a tour table with zone-by-zone paths and segment lists aligned to corridor links;
- a link table with canonical edge IDs and inter-zonal lengths for ERS decisions and cost/energy evaluation;
- a node table of VAM zones for SCS siting;
- a truck catalogue (`truck_id` $\rightarrow$ tours) enabling per-vehicle battery assignment.

Together, these artefacts provide a consistent handoff to Chapter 4: the upper level selects SCS/ERS and assigns one battery class per `truck_id`, while the lower level evaluates tour feasibility and operating costs along the reconstructed corridor paths.

4 Model Formulation

4.1 Modelling Framework and Rationale

Building on the data pipeline in Chapter 3—specifically the Delaunay-based VAM corridor graph (nodes as SCS candidates, links as ERS candidates; Section 3.5.1) and the tour paths reconstructed as shortest paths on that graph (Section 3.5.2)—this study formulates a bi-level model that couples long-term investment with tour-level operations. All energy and corridor-fee calculations in the model use the *reconstructed VAM path lengths*, ensuring a common metric between operations and infrastructure decisions.

At the strategic (upper) level, planners jointly decide: (i) where to build SCS on VAM nodes, (ii) which inter-zonal links to equip with ERS, and (iii) which battery-capacity class to assign to each truck. These are capital decisions contributing to CAPEX (facilities and batteries). Battery assignment is *per truck* and remains fixed across all of that vehicle's tours, avoiding the unrealistic assumption of per-tour battery switching.

At the operational (lower) level, each tour is evaluated along its reconstructed VAM path. Energy feasibility is enforced via segment-by-segment SOC balance. ERS provides *continuous* in-link charging (power–time approximation on equipped links), while SCS charging at nodes is *conditional*: it occurs only when the current SOC is insufficient to traverse the next segment and is limited by the battery headroom. Feasible tours contribute OPEX (electricity, time, corridor-related fees); infeasible tours are outsourced or diesel-executed and incur a penalty capturing fuel, CO_2, and distance-based surcharges. The lower level does not decide battery size; it tests feasibility under the upper-level configuration. Initial SOC is set to a fraction α of the assigned capacity.

A single-level model would miss these interactions: pure facility location can yield operationally infeasible layouts, while purely operational models cannot capture long-term investment trade-offs. The bi-level framework resolves this by coupling the layers through feasibility indicators (e.g., s_t for served, $r_t = 1 - s_t$ for unserved) and a composite objective combining *CAPEX (SCS, ERS, batteries)*, *OPEX* for served tours, and *penalties* for unserved tours (with weight Ω that prioritises feasibility). Figure 9 illustrates the structure.

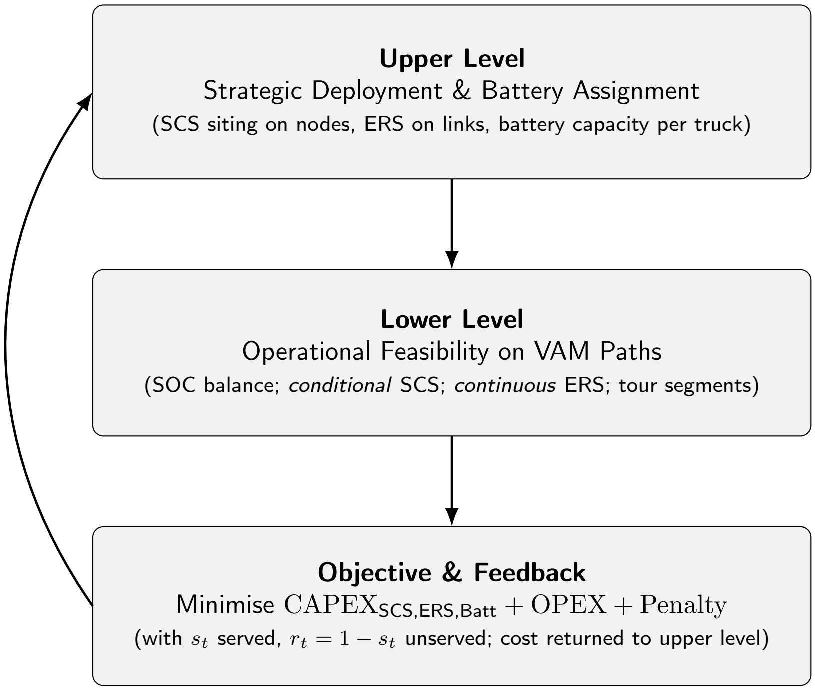


Figure 9: Bi-level optimisation framework aligned to the VAM corridor graph: upper-level SCS/ERS siting and per-truck battery assignment; lower-level tour feasibility on reconstructed VAM paths.

To operationalise this framework, each tour is classified as feasible or infeasible under the given infrastructure and assigned battery. A tour is feasible when its SOC never drops below zero and required charging can be met using available SCS/ERS along the VAM path; it contributes OPEX (electricity, time, and corridor fees). Otherwise the tour is unserved and contributes a penalty. Figure 10 summarises this logic using the same graph-consistent inputs.

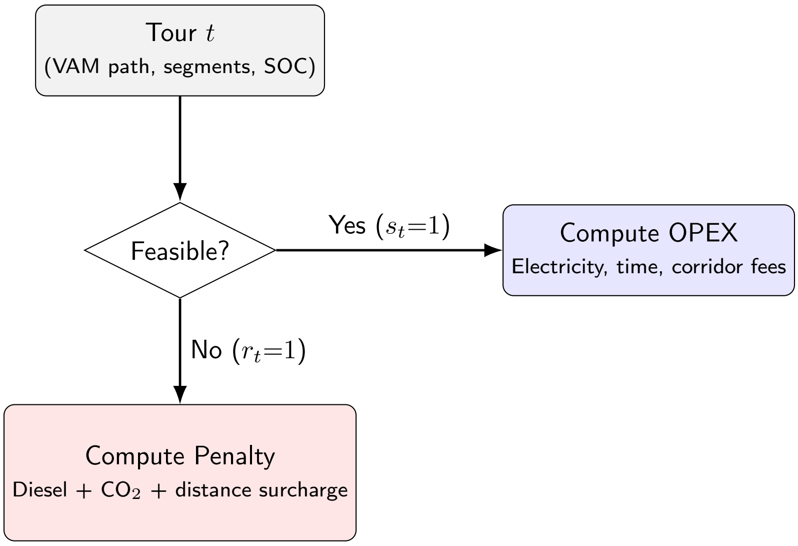


Figure 10: Feasibility–cost logic on the VAM path: served tours add OPEX; unserved tours add a penalty.

Service-first mechanism via r_t and Ω.

We introduce a binary indicator r_t for each tour to mark infeasibility/outsourcing ($r_t = 1$) versus service by a BE-HDT ($r_t = 0$, with $s_t = 1 - r_t$). In practice, the monetary penalty used for outsourcing a tour (diesel fuel, CO_2 cost, and a distance surcharge) is typically much lower than the capital expenditure required to purchase larger batteries or deploy additional infrastructure. If left unchecked, an optimiser would therefore tend to outsource rather than invest, yielding many infeasible tours. To counter this, a large weight Ω multiplies the count of unserved tours in the objective, acting as a modelling device that imposes a strong *service-first* preference: the search is discouraged from declaring tours infeasible and is nudged to invest in SCS/ERS or larger batteries instead. When $r_t = 1$, the lower-level energy-balance constraints are relaxed via big-M gating so that infeasible tours do not bind the physics while they still contribute a penalty term.

Importantly, Ω is not a real economic cost and is excluded from reported cost totals; it is used only to guide the optimisation away from excessive outsourcing. The precise placement of r_t, s_t, Ω, and the penalty term within the objective will be given later in Section 4.3.4.

4.2 Assumptions

Before presenting the mathematical formulation, several modelling assumptions are introduced to keep the bi-level optimisation tractable and aligned with the data pipeline (Chapter 3). Table 6 updates the original list to reflect the implemented logic: battery decisions are made at the *truck level* in the upper layer; tour operations are evaluated on reconstructed VAM paths with conditional SCS charging and continuous ERS charging; feasibility is coupled to the objective via an explicit unserved-tour mechanism.

Table 6: Modelling assumptions used in the bi-level optimisation framework

Area	**Assumption**	**Rationale / Role**
Energy consumption	Vehicles consume energy at a *constant, exogenously specified* rate per kilometre.	Simplifies demand as distance $\times$ rate; avoids dependence on payload, speed, or weather.
Initial battery state	Each tour departs with initial SOC equal to a fixed fraction α of its assigned battery capacity Q.	Provides a uniform but flexible starting SOC consistent with the code interface.

Continued on next page

Area	Assumption	Rationale / Role
Battery decision scope (upper level)	Battery capacity is assigned *per truck* (one class per vehicle) and counted as *CAPEX*; tours do not choose or switch batteries.	Matches fleet procurement; prevents per-tour double-counting of battery costs; clarifies upper–lower responsibilities.
Truck–tour mapping	Every tour t is linked to a unique truck $\kappa(t)$ (from Chapter 3), inheriting that truck's battery capacity and initial SOC.	Enforces vehicle-level consistency across all tours of the same truck.
Charging opportunities	Vehicles may recharge *only on their predetermined VAM path*: at SCS *nodes* and ERS *links* encountered along the route; no detours.	Keeps routing fixed and graph-consistent with infrastructure siting; avoids operational rerouting decisions.
Static (SCS) charging policy	SCS charging at a node is *conditional*: it occurs only if current SOC is insufficient to traverse the next segment, and is limited by battery headroom.	Reflects the implemented decision rule; avoids unnecessary dwell while preserving feasibility.
Dynamic (ERS) charging model	ERS provides *continuous* in-link charging modelled by a power–time approximation on equipped links, subject to deployment and battery headroom.	Tractable representation of in-motion charging; aligns with segment travel time on the corridor graph.
Charging-facility capacity	No explicit queuing or grid-capacity limits are modelled; facilities are treated as *always available*.	Excludes micro-congestion so the optimisation focuses on strategic siting and scale.
Distance / cost metric (graph-consistent)	Energy use, time, and corridor-related fees are computed from the *reconstructed VAM shortest-path lengths*; original trip distances are for diagnostics only.	Ensures a single metric for both operations and infrastructure placement (nodes=SCS, links=ERS).
Feasibility and penalty coupling	Each tour has indicators s_t (served) and r_t=1−s_t (unserved). Unserved tours incur a penalty; a weight Ω prioritises feasibility.	Makes the service gap explicit and tunable in the upper objective.
Big-M gating for unserved tours	When s_t=0, energy-balance constraints are relaxed via big-M terms.	Technically decouples infeasible tours from lower-level physics while keeping them in the objective.
Demand & routing	OD demand, the set of tours, and their VAM-path routes are *given and deterministic.*	Clarifies that stochastic demand and dynamic rerouting are out of scope.
Travel time and efficiencies	Segment travel times and charging efficiencies are treated as *constant parameters*; a default average speed is used where segment speeds are unavailable.	Provides a stable distance–time–energy mapping for ERS/SCS calculations.
Electricity & toll prices	Electricity tariffs and corridor/toll rates are *time-invariant averages.*	Simplifies OPEX; avoids time-of-use pricing and dynamic tolling.
Battery degradation	Battery ageing and replacement are *not modelled*; capacity and efficiencies are constant over the planning horizon.	Keeps the model focused on near/medium-term infrastructure decisions.
Fleet heterogeneity (operational)	Trucks are homogeneous in consumption rate and charging behaviour; heterogeneity arises only through *upper-level* battery-capacity assignment.	Reduces state space while retaining the key investment dimension.
Policy and behaviour	User behaviour, charging preferences, and policy incentives are *not explicitly modelled.*	Maintains tractability; such factors can be explored in scenarios/simulations.

4.3 Mathematical Model Formulation

To operationalise the bi-level optimisation framework introduced in Section 4.1, this section formalises the model in detail. We first define the sets and indices used to represent VAM nodes and links, trucks, battery classes, tours, and the ordered segments of each tour along its reconstructed VAM path. We then introduce parameters and decision variables.

The upper-level objective combines *CAPEX* (SCS siting, ERS deployment, and per-truck battery investment), *OPEX* aggregated over served tours, and *penalties* for unserved tours. The lower level does not select batteries; it enforces tour-level energy feasibility via SOC balance with *conditional* SCS charging at nodes and *continuous* ERS charging on equipped links, and returns the corresponding OPEX. The two layers are coupled by service indicators (s_t served, $r_t = 1 - s_t$ unserved) and big-M gating of constraints when a tour is not served.

All energy, time, and corridor-fee calculations use the length of the reconstructed VAM shortest paths from Chapter 3, ensuring a common metric between operations and infrastructure decisions. The subsequent subsections present the sets/indices, parameters and decision variables, followed by the objective and the key constraints that guarantee energy feasibility and logical consistency.

It is important to note that the assignment of tours to trucks is not part of the optimisation problem itself. Instead, the mapping $\kappa(t)$ linking each tour t to a truck k is generated exogenously during the data preparation stage (Chapter 3). The optimisation framework therefore assumes this mapping as fixed input: trucks are pre-defined carriers of a set of tours, and the model only decides on battery investment and infrastructure deployment. This distinction ensures that battery capacities are consistently applied across all tours of the same truck, while avoiding any endogenous reallocation of tours between trucks.

4.3.1 Sets and Indices

Table 7 summarises the sets and indices used in the formulation. The corridor graph follows the Delaunay construction in Chapter 3: nodes are SCS candidates and *undirected* links are ERS candidates; link lengths are used throughout for energy and cost calculations.

Table 7: Sets and indices used in the model

Symbol	**Meaning**	**Index**
N	VAM nodes (SCS candidates), $\lvert N\rvert = 542$	$n \in N$
L	Undirected inter-zonal links (ERS candidates), $\lvert L\rvert = 1613$	$\ell \in L$
K	Set of trucks	$k \in K$
T	Set of tours	$t \in T$
B	Battery-capacity classes	$b \in B$
I_t	Ordered segments within tour t	$i \in I_t$
T_k	Tours assigned to truck k ($T_k = \{t \in T : \kappa(t) = k\}$)	$t \in T_k$

Auxiliary mappings and notations. For each tour $t \in T$ and segment $i \in I_t$ we define:

- $\kappa(t) \in K$: the truck assigned to tour t (from Chapter 3).
- $n(t,i) \in N$: the *start node* of segment i (the VAM centroid where the $(i-1)$-th trip of tour t ends). Static charging opportunities (SCS) are evaluated at $n(t,i)$ before traversing segment i.
- $\ell(t,i) \in L$: the (undirected) corridor link traversed in segment i. ERS availability is determined by whether this link is equipped.
- $d_{t,i} = d_{\ell(t,i)}$: length of segment i (km), inherited from the link table; used for energy and corridor-fee calculations. Optionally, $v_{t,i}$ denotes the reference speed for time conversion.

This representation keeps operational evaluation (SOC balance, SCS/ERS opportunities) consistent with infrastructure siting on the same corridor graph.

4.3.2 Parameters

Table 8 lists the key parameters. They align the economic and technical inputs with the graph-consistent evaluation used in this study.

Table 8: Parameters used in the bi-level optimisation model

Symbol	**Description**
c_n^{SCS}	Annualised CAPEX of building an SCS at node n (unit: €; upper level).
c_ℓ^{ERS}	Annualised CAPEX of electrifying link ℓ (per km; unit: €/km; upper level).
d_ℓ	Length of link ℓ on the VAM graph; basis for energy, time and corridor-fee calculations (unit: km).
c_b^{Bat}	Battery CAPEX for capacity class b (per truck, upper level). Alternatively computed as $c^{\mathrm{Bat/kWh}} \times Q_b$ (units: € or €/kWh).
Q_b	Nominal capacity of battery class b (unit: kWh).
α	Initial SOC fraction at tour start ($SOC_0 = \alpha Q$ for the assigned truck battery; unit: –).
P^{SCS}	SCS charging power (unit: kW).
P^{ERS}	ERS in–motion charging power (unit: kW).
η^{SCS}, η^{ERS}	Charging efficiencies for SCS/ERS (constants; unit: –).
v_{avg}	Average driving speed used to couple ERS power to time when segment speeds are unavailable (unit: km/h).
p^{stat}, p^{dyn}	Electricity tariffs for static/dynamic charging (unit: €/kWh).
τ^{SCS}	Station service fee per SCS charging session (optional; unit: €).
τ_ℓ	Corridor fee/toll applied on link ℓ (per km; unit: €/km).
c^{time}	Value of time (driver/vehicle/cargo; unit: €/h).
Ω	Feasibility weight multiplying unserved tours in the objective (unit: –).
π_t	Base penalty if tour t is not served by BEV (outsourcing, etc.; unit: €).
p_{diesel}	Diesel price (unit: €/L).
f_{diesel}	Diesel fuel intensity (unit: L/km).
$e_{\mathrm{diesel}}^{\mathrm{CO2}}$	CO_2 emission factor of diesel (unit: gCO_2/L).
λ_{CO2}	Carbon price (unit: €/gCO_2).
ϕ	Additional distance-based surcharge used in the penalty term (unit: €/km).
β	BEV energy consumption rate per km (constant; unit: kWh/km).
M	Big-M constant used to gate energy-balance constraints when a tour is unserved (unit: –).

Note

(i) ERS/SCS costs and operations are evaluated on the same VAM graph; hence d_ℓ is the sole distance metric for energy and corridor fees. (ii) Battery CAPEX is an *upper-level, per-truck* investment; tours never choose batteries. (iii) CO_2 emissions for penalties are derived as $f_{\mathrm{diesel}} \cdot d \times e_{\mathrm{diesel}}^{\mathrm{CO2}}$ rather than treated as a per-tour parameter. (iv) If detailed segment speeds $v_{t,i}$ are available, they supersede v_{avg} for ERS time coupling.

4.3.3 Decision Variables

Upper level: Strategic decisions include SCS siting on nodes, ERS deployment on links, and the truck–level battery assignment:

$$x_n \in \{0,1\}\ (n \in N), \qquad y_\ell \in \{0,1\}\ (\ell \in L), \qquad \delta_{k,b} \in \{0,1\}\ (k \in K,\ b \in B).$$

Lower level: Operational variables evaluate each tour on its reconstructed VAM path:

$$s_t, r_t \in \{0,1\}\ (t \in T), \qquad e_{t,i}^{\mathrm{SCS}} \geq 0,\ e_{t,i}^{\mathrm{ERS}} \geq 0,\ SOC_{t,i} \geq 0\ (i \in I_t).$$

Here $n(t,i)$ denotes the *start node* of segment i in tour t and $\ell(t,i)$ the (undirected) link traversed; $SOC_{t,0}$ is the initial SOC and $SOC_{t,i}$ is defined for $i = 0, \dots, |I_t|$.

Table 9: Decision variables used in the bi-level optimisation model

Symbol	**Description**
$\delta_{k,b} \in \{0,1\}$	Equals 1 if truck k is assigned battery class b; exactly one class is chosen per truck at the upper level.
$x_n \in \{0,1\}$	Equals 1 if a static charging station (SCS) is built at node n.
$y_\ell \in \{0,1\}$	Equals 1 if link ℓ is equipped with ERS (dynamic charging). Links are undirected and identified on the VAM corridor graph.
$r_t \in \{0,1\}$	Infeasibility / outsourcing flag: equals 1 if tour t is unserved by BEV (infeasible w.r.t. SOC) and therefore outsourced; 0 otherwise.
$s_t \in \{0,1\}$	Service indicator: equals 1 if tour t is executed by a BEV; 0 if outsourced. Enforced by $s_t = 1 - r_t$ at the model level.
$e^{\mathrm{SCS}}_{t,i} \geq 0$	Energy charged *statically* at the start node $n(t,i)$ before traversing segment i (kWh, measured on the battery side and limited by headroom).
$e^{\mathrm{ERS}}_{t,i} \geq 0$	Energy charged *dynamically* on link $\ell(t,i)$ during segment i (kWh, limited by ERS deployment, travel time and headroom).
$z^{\mathrm{SCS}}_{t,i} \in \{0,1\}$	Binary indicator: equals 1 if static charging is activated at node $n(t,i)$ before traversing segment i; 0 otherwise. Used to trigger SCS tolls in the OPEX formulation.
$z^{\mathrm{ERS}}_{t,i} \in \{0,1\}$	Binary indicator: equals 1 if dynamic charging is activated on link $\ell(t,i)$ during segment i; 0 otherwise. Used to trigger ERS tolls in the OPEX formulation.
$SOC_{t,i} \geq 0$	State of charge of tour t at segment index i (kWh). $SOC_{t,0}$ is the initial SOC; $SOC_{t,\lvert I_t \rvert}$ is the terminal SOC.

4.3.4 Objective Function

We minimise a single composite objective that mirrors the implementation:

$$\min\ Z \;=\; \underbrace{\Omega \sum_{t\in T} r_t}_{\text{service-first priority}} \;+\; \underbrace{\mathrm{CAPEX}_{\mathrm{fac}}(x,y)}_{\text{SCS/ERS}} \;+\; \underbrace{\mathrm{CAPEX}_{\mathrm{bat}}(\delta)}_{\text{truck-level batteries}} \;+\; \underbrace{\sum_{t\in T} s_t\, \mathrm{OPEX}_t}_{\text{served tours}} \;+\; \underbrace{\sum_{t\in T} r_t\, C^{\mathrm{pen}}_t}_{\text{unserved tours}}\,. \tag{1}$$

Upper Level (Strategic)

Facility CAPEX

Infrastructure investment follows the VAM corridor graph:

$$\mathrm{CAPEX}_{\mathrm{fac}}(x,y) = \sum_{n\in N} c^{\mathrm{SCS}}_n\, x_n + \sum_{\ell\in L} c^{\mathrm{ERS}}_\ell\, d_\ell\, y_\ell. \tag{2}$$

Battery CAPEX (truck level)

Each truck receives exactly one battery class at the upper level:

$$\mathrm{CAPEX}_{\mathrm{bat}}(\delta) = \sum_{k\in K} \sum_{b\in B} c^{\mathrm{Bat}}_b\, \delta_{k,b}. \tag{3}$$

(When costs are specified per kWh in the data interface, c^{Bat}_b is taken as $c^{\mathrm{Bat/kWh}} \times Q_b$ in implementation.)

Lower Level (Operational)

Operating cost

For each tour $t \in T$, the operating cost is decomposed into energy payments, charging dwell time, and facility tolls:

$$OPEX_t = \sum_{i \in I_t} \Big[p^{\text{stat}}\, e^{\text{SCS}}_{t,i} + p^{\text{dyn}}\, e^{\text{ERS}}_{t,i} + c^{\text{time}}\, \frac{e^{\text{SCS}}_{t,i}}{P^{\text{SCS}}} \\ + \tau^{\text{SCS}}\, d_{t,i}\, z^{\text{SCS}}_{t,i} + \tau^{\text{ERS}}_{\ell(t,i)}\, d_{t,i}\, z^{\text{ERS}}_{t,i} \Big], \tag{4}$$

where $z^{\text{SCS}}_{t,i}, z^{\text{ERS}}_{t,i} \in \{0,1\}$ are binary variables indicating whether SCS or ERS charging is activated on segment i of tour t.

Penalty for an unserved tour

If tour t is infeasible under the assigned battery and infrastructure (r_t=1), the penalty is applied per travelled distance $D_t = \sum_{i \in I_t} d_{t,i}$:

$$C^{\text{pen}}_t = D_t \Big(p_{\text{diesel}}\, f_{\text{diesel}} + \lambda_{\text{CO2}}\, e^{\text{CO2}}_{\text{diesel}} + \phi \Big). \tag{5}$$

The first two terms represent diesel fuel and carbon costs; ϕ is an additional distance-based outsourcing surcharge.

Notes on implementation

(i) The service indicators satisfy $s_t + r_t = 1$. (ii) ERS/SCS fees in (4) use indicator functions that are *derived* in the evaluator (fees are counted only if $e^{\text{ERS}}_{t,i}$ or $e^{\text{SCS}}_{t,i}$ is positive); no extra binaries are introduced for these indicators. (iii) All distances d_ℓ and $d_{t,i}$ are measured on the reconstructed VAM shortest paths, ensuring consistency between siting and operating-cost evaluation.

4.3.5 Key Constraints

Indexing

For each tour t, let $I_t = \{1, \ldots, m_t\}$ be the ordered set of segments along its reconstructed VAM path; $SOC_{t,i}$ is the state of charge *at the start of segment* i, and SOC_{t,m_t+1} is the terminal SOC.

(i) Battery selection and capacity (mandatory).

$$\sum_{b \in B} \delta_{k,b} = 1 \qquad \forall k \in K, \tag{B1}$$

$$\delta_{k,b} \in \{0,1\} \qquad \forall k \in K,\ \forall b \in B, \tag{B2}$$

$$Q_k = \sum_{b \in B} Q_b\, \delta_{k,b} \qquad \forall k \in K. \tag{B3}$$

Explanation: one truck can only assign to one type of battery

(ii) Service–infeasibility link

$$s_t + r_t = 1 \qquad \forall\, t \in T. \tag{6}$$

Explanation: each tour is either served by a BEV (s_t=1, r_t=0) or outsourced/unserved (s_t=0, r_t=1); voluntary outsourcing of feasible tours is excluded.

(iii) Initial SOC and bounds (start at $i = 1$)

$$SOC_{t,1} = \alpha\, Q_{\kappa(t)} \qquad \forall\, t \in T, \tag{7}$$
$$0 \;\le\; SOC_{t,i} \;\le\; Q_{\kappa(t)} + M^{SOC} r_t \qquad \forall\, t \in T,\; i = 1,\dots,m_t{+}1. \tag{8}$$

Explanation: each tour starts at a fixed fraction α of its truck's battery $Q_{\kappa(t)}$, and SOC cannot exceed capacity; when a tour is marked unserved (r_t=1) these bounds are relaxed by big-M.

(iv) Segment energy balance (gated by infeasibility)

$$-M^{SOC} r_t \;\le\; SOC_{t,i+1} - SOC_{t,i} + \beta\, d_{t,i} - \eta^{\mathrm{SCS}} e^{\mathrm{SCS}}_{t,i} - e^{\mathrm{ERS}}_{t,i} \;\le\; M^{SOC} r_t, \quad \forall\, t \in T,\; i = 1,\dots,m_t. \tag{9}$$

Explanation: SOC decreases by driving ($\beta d_{t,i}$) and increases by SCS/ERS charging; if r_t=1 the equality is deactivated via big-M so infeasible tours do not bind the physics.

(v) Static (SCS) charging at start nodes

$$0 \;\le\; e^{\mathrm{SCS}}_{t,i} \;\le\; x_{\,n(t,i)}\big(Q_{\kappa(t)} - SOC_{t,i}\big) + M^{E} r_t, \quad \forall\, t \in T,\; i = 1,\dots,m_t. \tag{10}$$

Explanation: static charging before segment i is only allowed if an SCS is installed at the start node and is capped by battery headroom; it is disabled for unserved tours.

(vi) Dynamic (ERS) charging on links

$$0 \;\le\; e^{\mathrm{ERS}}_{t,i} \;\le\; y_{\,\ell(t,i)}\left(\eta^{\mathrm{ERS}} \frac{P^{\mathrm{ERS}}}{v_{t,i}}\, d_{t,i}\right) + M^{E} r_t, \qquad \forall\, t \in T,\; i = 1,\dots,m_t, \tag{11}$$
$$e^{\mathrm{ERS}}_{t,i} \;\le\; Q_{\kappa(t)} - SOC_{t,i} + M^{E} r_t, \qquad \forall\, t \in T,\; i = 1,\dots,m_t. \tag{12}$$

Explanation: in-motion charging on segment i is limited by (i) whether the link is ERS-equipped and the power–time budget, and (ii) the battery headroom; it is disabled for unserved tours.

(vii) Terminal SOC (served tours)

$$SOC_{t,m_t+1} \;\ge\; 0 \;-\; M^{SOC} r_t \qquad \forall\, t \in T. \tag{13}$$

Explanation: a served tour must end with non-negative SOC; if unserved, this requirement is relaxed by big-M.

(viii) Linking and feasibility constraints

$$e^{\mathrm{SCS}}_{t,i} \;\le\; M^{\mathrm{SCS}}\, z^{\mathrm{SCS}}_{t,i}, \qquad \forall t \in T, \forall i \in I_t, \tag{14}$$
$$e^{\mathrm{ERS}}_{t,i} \;\le\; M^{\mathrm{ERS}}\, z^{\mathrm{ERS}}_{t,i}, \qquad \forall t \in T, \forall i \in I_t, \tag{15}$$
$$z^{\mathrm{SCS}}_{t,i} \;\le\; x_{n(t,i)}, \qquad z^{\mathrm{ERS}}_{t,i} \;\le\; y_{\ell(t,i)}, \qquad \forall t, i, \tag{16}$$
$$z^{\mathrm{SCS}}_{t,i} \;\le\; 1 - r_t, \qquad z^{\mathrm{ERS}}_{t,i} \;\le\; 1 - r_t, \qquad \forall t, i, \tag{17}$$
$$z^{\mathrm{SCS}}_{t,i}, z^{\mathrm{ERS}}_{t,i} \in \{0,1\}. \tag{18}$$

Here M^{SCS} and M^{ERS} are sufficiently large constants (e.g., taken from M_E) to enforce that $z = 1$ whenever the corresponding charging energy is positive.

(ix) Non-negativity

$$e^{\mathrm{SCS}}_{t,i} \ge 0, \qquad e^{\mathrm{ERS}}_{t,i} \ge 0 \qquad \forall\, t \in T,\; i = 1,\dots,m_t. \tag{19}$$

Explanation: charged energies are non-negative by definition.

(x) Big-M choices (tight)

$$M^{SOC} = \max_{k\in K} Q_k, \qquad M^{E} = \max\left\{\eta^{\mathrm{SCS}}Q^{\max},\ \eta^{\mathrm{ERS}}\frac{P^{\mathrm{ERS}}}{v_{\min}}d^{\max}\right\}, \tag{20}$$

Explanation: big-M constants are chosen as tight, data-driven upper bounds (max battery, max ERS energy per segment) to stabilise the relaxation without over-penalising numerics; in experiments $\eta^{\mathrm{SCS}} = \eta^{\mathrm{ERS}} = 1$.

4.4 Integrated Model Mechanism

The model operates by linking upper-level investment decisions with lower-level tour feasibility checks in a unified framework. At the core lies the assignment of batteries to trucks: each vehicle is equipped with exactly one battery capacity class, which determines its maximum usable energy throughout all tours. This constraint ensures consistency across a truck's operations and prevents unrealistic switching of battery sizes between trips.

Service feasibility is enforced through the penalty weight Ω, which magnifies the cost of infeasible tours. By doing so, the model prioritizes serving tours with electric vehicles whenever technically possible, and only resorts to outsourcing or diesel-based penalties if no feasible charging configuration exists. This mechanism strongly drives the solution toward high electrification coverage.

Battery selection is embedded in the objective function through $CAPEX^{bat}(\delta)$. The optimizer weighs the trade-off between investing in larger, more expensive batteries that can cover long tours without recharging, and choosing smaller, cheaper batteries that rely on strategically placed infrastructure. This co-optimization between vehicle-side investment and network-side infrastructure deployment enables cost-efficient planning at scale.

Finally, the charging strategy at the lower level follows an on-demand principle. Static charging at SCS facilities only occurs when the current state-of-charge is insufficient to cover the upcoming segment, and only the minimum required energy is drawn. Dynamic charging on ERS segments is similarly conditional: it is activated solely when the remaining energy would otherwise fall short of completing the tour. This prevents unnecessary charging and ensures that operational costs reflect rational economic behavior.

Through this combination of truck-level battery assignment, strong infeasibility penalties, balanced investment trade-offs, and cost-driven charging behavior, the model delivers a realistic and comprehensive framework for planning electric road freight systems.

5 Solution Approach

5.1 Genetic Algorithm for the Bi-level Model

The optimisation problem formulated in Chapter 4 is a large-scale, NP-hard combinatorial problem. It simultaneously decides on static charging station siting, dynamic ERS deployment, and battery capacity assignment for a nationwide truck fleet. The decision space is extremely large due to the binary nature of facility placement and the discrete multi-class structure of battery choices. Moreover, the feasibility of each tour depends on nonlinear SOC balance constraints under the smart charging strategy. These features make exact methods such as mixed-integer linear programming (MILP) computationally intractable at scale, motivating the use of metaheuristics.

The Genetic Algorithm (GA) is particularly well suited to this problem setting. GA evolves a population of solutions through selection, crossover, and mutation, enabling exploration of vast search spaces without relying on convexity or linearity. It is capable of handling mixed discrete decision variables, nonlinear feasibility checks, and multi-objective cost components. Furthermore, each individual's fitness evaluation is independent, making the approach amenable to parallelisation.

Empirical support for GA has been demonstrated in several charging-infrastructure studies. Vazifeh et al. (2019) showed that GA improved both user inconvenience and the number of charging stations compared to greedy approaches when optimising EV station siting in Boston [42]. Akbari et al. (2018) applied GA to Milan and found rapid convergence within a few hundred generations [1]. Cintrano et al. (2021) also confirmed superior performance of GA over Variable Neighbourhood Search (VNS) in a case study in Málaga, Spain [6]. More recently, Seilabi et al. (2025) verified the applicability of a GA-based bi-level optimisation framework in sustainable EV charging station planning, underscoring the method's relevance for large-scale real-world networks [10].

In the context of this thesis, GA is tailored to the bi-level model. Chromosomes are structured in three segments to jointly encode SCS siting, ERS deployment, and grouped truck battery capacities. The initialisation procedure is biased toward small batteries to reduce investment costs and improve facility utilisation, while mutation is designed to favour downward transitions in battery capacity. Diversity is preserved through Hamming distance checks, and a periodic restart strategy is adopted to avoid premature convergence. The fitness evaluation directly corresponds to the objective function defined in Chapter 4, including CAPEX, OPEX, infeasibility penalties, and the Ω term enforcing service priority. This tailored GA design ensures both scalability to national-level datasets and alignment with the model's decision structure.

5.2 GA Mechanisms and Workflow

The evolutionary process of the tailored GA follows the classical structure of genetic algorithms, adapted to the bi-level optimisation model presented in Chapter 4. After each chromosome is decoded into infrastructure and battery decisions, the following operators are applied in each generation:

- **Selection**: A tournament-based scheme is used to ensure that solutions with lower system cost (better fitness) have a higher probability of reproducing in the next generation.
- **Crossover**: Offspring are generated by exchanging subsequences of parent chromosomes, with crossover points aligned with the three main segments (SCS siting, ERS deployment, and battery grouping). This preserves structural integrity while allowing diverse recombinations.
- **Mutation**: Individual bits of the chromosome are flipped with low probability to maintain diversity. In the battery segment, mutation is biased toward smaller capacity classes to avoid over-investment and encourage facility reliance.
- **Elitism**: The best-performing individuals in each generation are directly copied into the next generation, ensuring that high-quality solutions are never lost.

The *fitness evaluation* of each individual corresponds exactly to the objective function defined in Chapter 4. For a candidate solution, the evaluator computes the total system cost

$$f \;=\; \Omega \sum_{t\in T} r_t \;+\; CAPEX(x,y,\delta) \;+\; OPEX(e^{\mathrm{SCS}}, e^{\mathrm{ERS}}) \;+\; \sum_{t\in T} r_t C_t^{pen}, \tag{21}$$

where the Ω term penalises infeasible tours, $CAPEX$ represents facility and battery investment, $OPEX$ covers operational energy, toll, and time costs, and C_t^{pen} denotes the outsourcing penalty. This alignment between the model objective and the GA fitness ensures that only cost-efficient and feasible infrastructure–battery configurations survive across generations.

Through repeated cycles of selection, crossover, mutation, and elitism, the GA gradually evolves the population toward high-quality solutions, while the additional mechanisms described in Section 5.2 guarantee scalability and robustness.

5.2.1 GA Solution Strategy for the Bi-level Model

While the mechanisms of the GA define how populations evolve, the essence of solving the bi-level model lies in how the algorithm directs the search toward cost-efficient and feasible configurations. The evaluation of each chromosome is guided by the hierarchical cost structure of the optimisation model, and the GA naturally balances these objectives through its fitness function.

Minimising infeasible tours: The highest priority of the GA is to reduce the number of infeasible tours. This is achieved through the Ω term in the fitness function, which imposes a very large penalty whenever a tour cannot be served by an electric truck under the given infrastructure and battery configuration. As a result, the evolutionary search first focuses on finding configurations that maximise feasibility, ensuring that a large share of tours can be operated electrically.

Minimising infrastructure investment: Once feasibility is improved, the GA gradually seeks to reduce system cost by minimising infrastructure CAPEX. Through crossover and mutation, solutions with fewer SCS sites and shorter ERS stretches are favoured, provided they still maintain feasibility. This ensures that the algorithm avoids over-deployment of facilities and converges to leaner network layouts.

Minimising battery investment: Battery costs are incorporated directly through the $\delta_{k,b}$ variables in the chromosome. The biased initialisation and mutation operators promote smaller battery capacities whenever feasible, thereby reducing vehicle-side CAPEX. This mechanism reflects the trade-off between investing in large batteries, which increase feasibility but are expensive, and relying on infrastructure to support smaller, cheaper batteries.

Minimising operational cost (OPEX); Among feasible solutions with reasonable investment levels, the GA further discriminates based on operating cost. The smart charging strategy embedded in the evaluator ensures that SCS and ERS are only used when strictly necessary, preventing unnecessary charging costs. Thus, OPEX is reduced by both efficient facility placement and rational vehicle-level charging behaviour.

Penalty as a last resort: Tours that cannot be served by any feasible combination incur penalty costs reflecting diesel usage, emissions, and outsourcing. Although these terms are always dominated by the Ω penalty, they still influence the search by differentiating between alternative infeasible solutions. This ensures that, if infeasibility cannot be entirely eliminated, the algorithm still favours configurations with lower environmental and outsourcing impact.

Through this layered fitness evaluation, the GA sequentially drives the population toward solutions that (i) maximise feasibility, (ii) minimise infrastructure cost, (iii) minimise battery cost, (iv) reduce operational expenses, and (v) only as a last resort accept penalties.

This ordering of priorities mirrors the objective structure of the model and ensures that the evolutionary process converges toward globally cost-efficient and practically implementable infrastructure–battery configurations.

Figure 11 provides a visual summary of the tailored GA workflow. The diagram illustrates how the algorithm begins with a biased initialisation of the population, decodes each chromosome into infrastructure and battery decisions, and then evaluates fitness according to the layered priority structure: first minimising infeasible tours through the Ω penalty, then reducing infrastructure and battery investment, and finally refining operational costs and penalties. Genetic operators (selection, crossover, mutation, and elitism) are applied to evolve the population, while diversity control and periodic restarts ensure robustness against premature convergence. The loop continues until the termination criterion is met, after which the best configuration of infrastructure and battery assignments, along with detailed performance statistics, is reported.

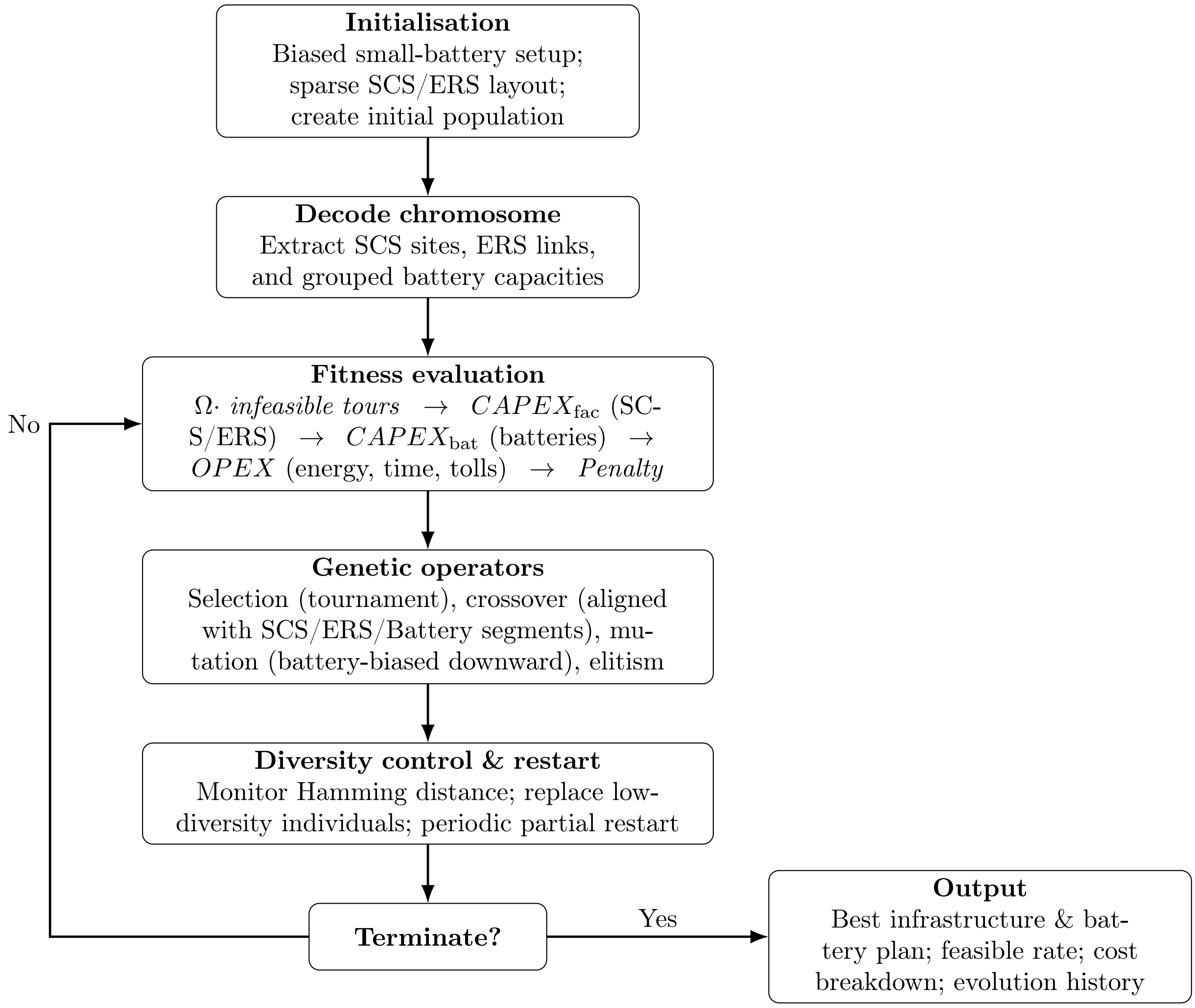


Figure 11: Workflow of the tailored GA for the bi-level optimisation model.

5.3 Parameter Configuration

5.3.1 GA Parameters

The GA parameters are set following both literature guidance and pilot experiments on the national dataset. Table 10 summarises the adopted configuration. These values represent a balance between solution quality and computational tractability: a moderate population size and number of generations to keep runtime within practical limits, a relatively high crossover rate to encourage recombination of infrastructure and battery decisions, and a slightly elevated mutation rate to sustain diversity in the enlarged search space. The restart interval and diversity threshold provide additional safeguards against premature convergence.

Table 10: Genetic Algorithm parameter configuration

Parameter	**Description**	**Value**
Population size	Number of individuals in each generation	30
Number of generations	Maximum number of evolutionary cycles	30
Crossover rate	Probability of exchanging subsequences between parent chromosomes	0.80

Parameter	Description	Value
Mutation rate	Probability of flipping chromosome bits; biased toward smaller battery capacities	0.15
Elitism size	Number of best-performing individuals copied to the next generation	5
Tournament size k	Number of individuals competing in each tournament selection	3
Restart interval	Generations after which part of the population is reinitialised	12
Diversity threshold	Minimum Hamming ratio enforced between individuals	0.12

5.3.2 Value of Model Parameters

Ω In this study, the feasibility weight Ω is set to a fixed value of 100,000. The role of Ω is not to represent a real economic cost, but to enforce a strong service-first preference in the optimisation. In practice, the outsourcing penalty for a tour is typically much lower than the capital cost of procuring additional batteries or installing new infrastructure. Without Ω, the optimiser would tend to declare many tours infeasible and outsource them, because this appears cheaper in monetary terms. By introducing Ω as a very large weight multiplied by the number of unserved tours, the model strongly discourages outsourcing and nudges the search toward investing in batteries and infrastructure. This greatly reduces the incidence of infeasible tours and ensures that the optimisation produces solutions with high electrification coverage. Importantly, Ω is excluded from the reported cost totals in the results (Chapter 6), as it functions purely as a modelling device to guide the search rather than a real-world cost component.

$C_{ij}^{\mathbf{ERS}}$ A study in Germany indicates that the infrastructure cost of suspended cable (catenary) ERS is approximately €1.7 - 3.1 million EUR per kilometer. For other types (electromagnetic or inductive), it is estimated to be €0.4 - 2.7 million per kilometer [36]. In this paper, 2 million EUR per kilometer is selected as the construction cost of ERS.

$C^{\mathbf{SCS}}$ According to the Rocky Mountain Institute, the hardware cost of a 150 kW DC fast charger is typically \$75,600–100,000, while a full site installation including make-ready and grid upgrades can be three to five times higher [27]. Based on these estimates, a representative value of €200,000 per logistics-scale SCS site is adopted in this study.

$c_{\mathbf{ba,B}}$ According to the BloombergNEF report, the price of lithium ion battery packs has fallen to \$115 per kWh in 2024 and has decreased by 20% compared to last year [35]. To account for the expected future costs or convert them into the large-scale logistics purchase price, and for the convenience of calculation, this article sets the battery price at €100 per kWh.

β The actual energy consumption of electric heavy trucks is approximately 1.08 - 1.3 kWh per kilometer [36]. Some studies have indicated that for urban areas and highways, it is within the range of 1.2 - 1.8 kWh per kilometer [37]. In this paper, 1.6 kWh per kilometer is selected.

$c^e_{\text{stat}}, c^e_{\text{dyn}}$ In the Netherlands, the national average for public charging (excluding fast chargers) is approximately €0.36/kWh [28]. For fast DC charging (over 50 kW), average prices range from €0.67 to €0.86/kWh [12], with an overall average of about €0.76/kWh. Accordingly, this study assumes a static charging price of €0.73/kWh—consistent with typical fast-charging rates—and a lower dynamic (ERS) charging price of €0.36/kWh, reflecting potential operational efficiencies or contractual arrangements in dynamic charging contexts.

c^{time} The value of freight vehicle driver time used in UK cost–benefit analysis (COBA framework) is approximately £18.95/h (2010 prices), equivalent to roughly €23/h today [26]. Given the additional operational and waiting costs typical in logistics operations, this study adopts a higher value of €38/h to reflect the true opportunity cost of driver time in freight transport.

$c_{\text{stat}}^{\text{toll}}, c_{\text{dyn}}^{\text{toll}}$ In Germany, heavy goods vehicles are charged an average road toll of approximately €0.15 per km on highways (LKW-Maut) [44]. Similarly, from 2026 the Netherlands will introduce a kilometre-based toll for trucks over 3.5 tonnes of around €0.15/km [31]. Therefore, a static charging toll of €0.15/km is adopted in this study. For dynamic ERS charging, a lower fee of €0.10/km is assumed, reflecting potential operational efficiencies or lower marginal infrastructure costs associated with dynamic charging.

P_S^{SCS}, P_E^{ERS} A typical ultra-fast DC charging station can provide around 150 kW continuous power, as demonstrated by commercially available systems (e.g., Heliox) [13]. For electric road systems capable of dynamic charging—such as Honda's conductive overhead implementation—power delivery can reach up to 450 kW [18]. In this study, a conservative yet realistic value of 150 kW is used for SCS, while 200 kW is assumed for ERS, reflecting moderate dynamic charging performance of current infrastructure prototypes.

v^{avg} In most European countries, heavy goods vehicles (HGVs) are legally limited to a maximum speed of approximately 80 km/h on motorways and expressways [14]. Therefore, an average truck speed of 80 km/h is adopted, representing a realistic operational average under combined motorway and interurban driving conditions.

C_t^{pen} The penalty cost per unserved tour is modelled using four components:

- Diesel fuel cost: €1.6/L, consistent with typical European retail prices (e.g., Belgium: €1.603/L; Denmark: €1.797/L) [5].
- Fuel consumption: 0.35 L/km, slightly above the ICCT-reported average of 0.326 L/km for tractor-trailers on long-haul routes [20].
- CO_2 emissions cost: €0.00008 per gram CO_2, aligned with current EU-ETS trading prices (€100/ton CO_2) [43].
- External surcharge: €0.10/km, included to reflect additional marginal costs (e.g., logistics delays, congestion, indirect environmental impacts), adopted as a conservative estimate.

5.4 GA Validation

To verify the accuracy of the GA, we formulated an exact Mixed-Integer Linear Programming (MILP) model identical to the problem as a "gold standard" for benchmarking under the same data and parameter settings. Specifically, we adopted unified cost and technical parameters (SCS = €200,000 per station, ERS = €500,000 per km, $\beta = 1.6$ kWh/km, $P_{SCS} = 150$ kW, $P_{ERS} = 200$ kW, battery capacity set = [100, 150, 200] kWh, ERS length options = [0, 25, 50] km, and excluding time costs) and solved on the same small-scale subsets (8, 15, and 25 tours) for comparison. The results show that the optimal total costs obtained by the GA match exactly with those of the MILP in all three cases: €68,000 for 8 tours, €124,000 for 15 tours, and €208,000 for 25 tours (difference = 0, error tolerance < €1,000). Moreover, the solution structures are consistent between the two: neither deploys SCS/ERS infrastructure (optimal being zero infrastructure), and battery selection is predominantly 100 kWh, with a few tours selecting 150 kWh to meet constraints. Furthermore, in the larger-scale case with 40 tours, the MILP could not run due to the size limitations of the free Gurobi license, whereas the GA was still able to stably produce feasible and cost-effective solutions, demonstrating its scalability. This comparison confirms that within the scale solvable by the MILP, the GA yields identical results, thereby proving its correctness and reliability in the given problem setting; and in larger-scale cases, the GA retains practical solution capability.

5.5 Summary

This chapter has presented the solution approach for the bi-level optimisation model. A tailored Genetic Algorithm was introduced as the heuristic method capable of handling the large-scale, combinatorial, and non-linear structure of the problem. The algorithm was specifically adapted to the joint optimisation of charging infrastructure siting and truck battery capacities through a three-segment chromosome design, truck grouping strategy, biased initialisation and mutation, and layered fitness evaluation prioritising feasibility, investment, and operating costs. Parameter settings were discussed to ensure both convergence quality and computational tractability.

Together, these methodological innovations establish a robust and scalable optimisation framework. The GA not only guarantees alignment with the model objectives but also provides practical decision support for national-scale electrification of heavy-duty freight. The next chapter will present the experimental results obtained with this framework and analyse the implications for infrastructure planning.

6 Result

6.1 Overall Performance

Building upon the modelling framework and the heuristic solution method introduced in Chapter 4 and Chapter 5, this section evaluates the nationwide optimisation outcomes obtained with the Genetic Algorithm (GA). The GA is employed to address the computational intractability of the bi-level MILP at full scale, while retaining the ability to explore investment trade-offs between static charging stations (SCS), electrified road segments (ERS), and truck battery capacities. Convergence plots are presented to illustrate how key performance metrics evolve over generations, providing insights into the algorithm's efficiency and the quality of the final solutions.

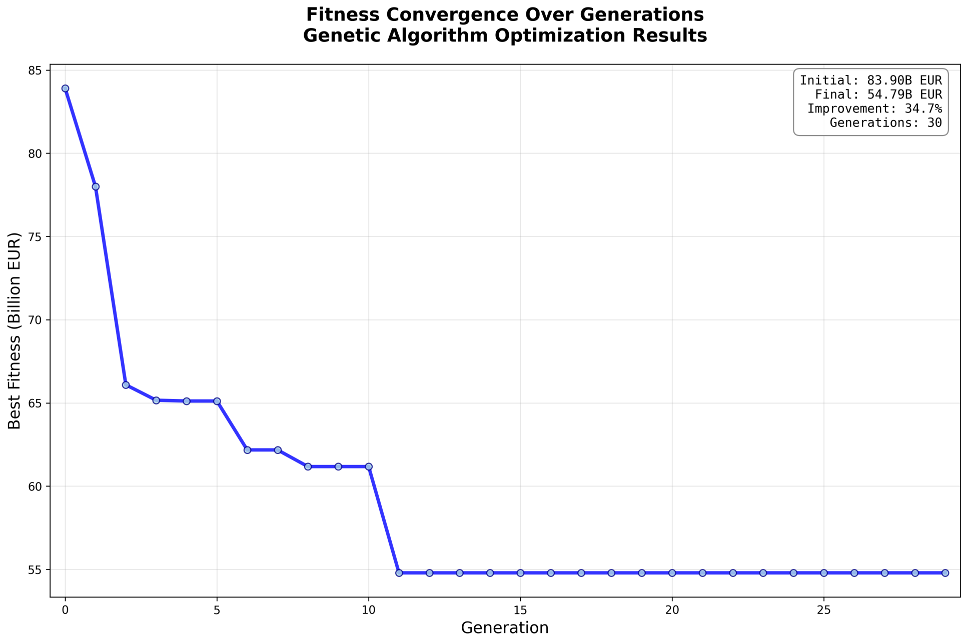


Figure 12: Convergence of best fitness over generations.

Figure 12 shows that the best fitness value decreases sharply within the first few generations, stabilising after approximately 12 generations. The overall fitness improves from €83.9 billion to €54.8 billion, corresponding to a 34.7% reduction. It should be noted that these values include the artificial Ω-penalty for infeasible tours, which enforces service-first behaviour during optimisation. In subsequent cost reporting, the Ω component is excluded to reflect true economic expenditures. The rapid convergence within fewer than 15 generations illustrates the computational efficiency of the GA on a nationwide-scale problem.

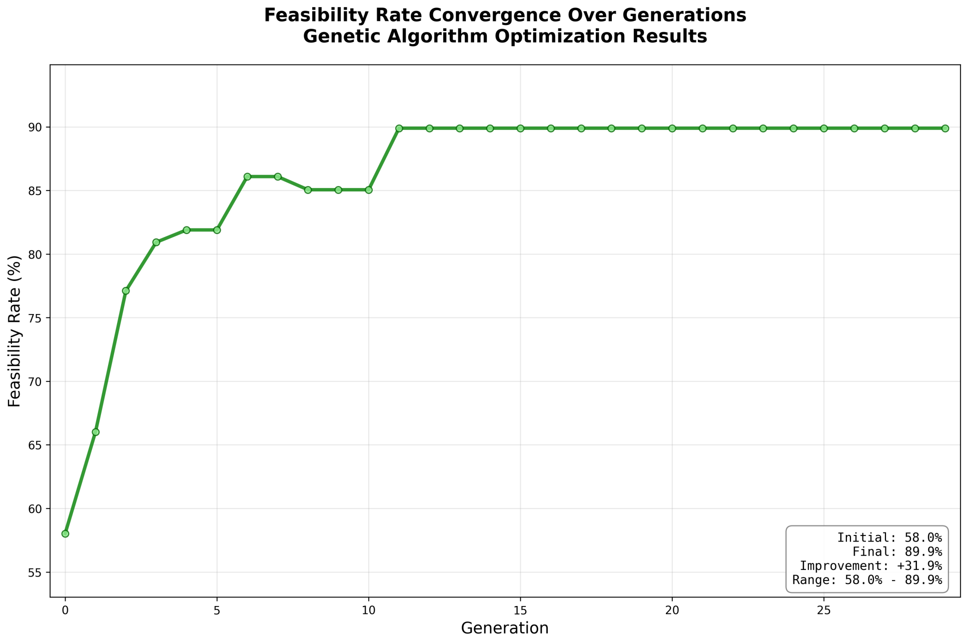


Figure 13: Feasibility rate convergence across generations.

A similar convergence pattern is observed in the feasibility rate (Figure 13). The proportion of tours that can be served by battery-electric trucks rises from 58.0% initially to 89.9% after optimisation, reflecting a 31.9 percentage-point improvement. The feasibility stabilises after about ten generations, highlighting that the GA prioritises reducing infeasible tours in the early search phase. Nevertheless, about 10.1% of tours remain infeasible, which motivates the residual infeasibility analysis presented in Section 6.3.

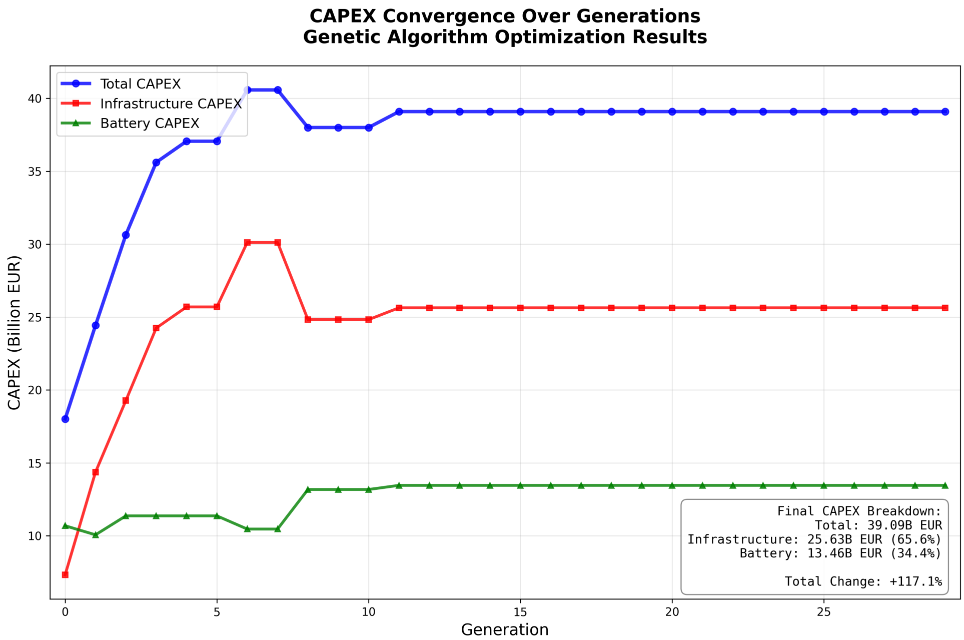


Figure 14: CAPEX convergence across generations, split by infrastructure and battery.

In terms of capital expenditure (Figure 14), the GA raises total CAPEX from €18 billion in the initial generation to €39.1 billion at convergence, split into €25.6 billion (65.6%) for infrastructure and €13.5 billion (34.4%) for batteries. This increase reflects a deliberate investment strategy: higher upfront costs are accepted to achieve large reductions in operating costs and infeasible tours.

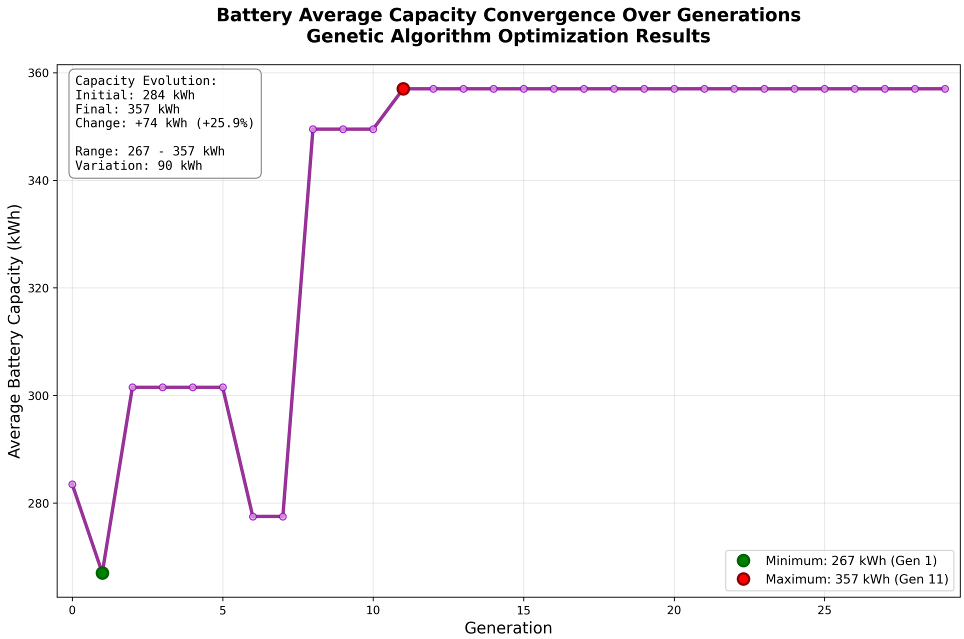


Figure 15: Convergence of average truck battery capacity.

Battery capacity assignment also converges to a stable configuration (Figure 15). The average assigned capacity per truck increases from 284 kWh to 357 kWh (+25.9%), with most of the adjustment occurring within the first 11 generations. This outcome illustrates that the GA does not simply assign maximum-capacity batteries; instead, it balances marginal battery costs against coverage benefits, leading to a heterogeneous but stable distribution of battery classes.

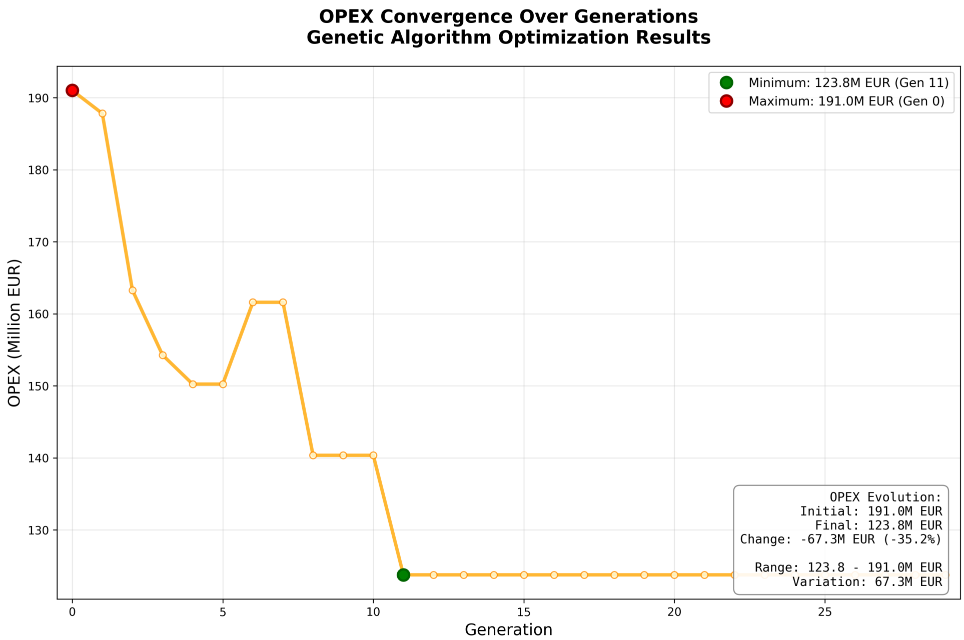


Figure 16: OPEX convergence across generations.

Indeed, the operating expenditure (OPEX) falls significantly over the search process (Figure 16), from €191 million to €124 million (-35.2%). The sharp decline in OPEX coincides with the period when both feasibility and battery capacity stabilise, confirming that improved service coverage directly translates into lower operational burden.

Overall, these results confirm the GA's ability to efficiently explore the large-scale search space and converge toward balanced solutions. The optimisation process yields a substantial reduction in infeasible tours, a clear trade-off between CAPEX and OPEX, and a stable battery configuration, thereby demonstrating both the algorithm's robustness and the practical feasibility of the proposed modelling framework. The following section disaggregates the resulting infrastructure layout, highlighting how ERS and SCS investments jointly contribute to the achieved feasibility gains.

6.2 Infrastructure Deployment

Figure 17 and Figure 18 illustrate the optimised nationwide charging infrastructure layout, combining dynamic ERS links and static SCS nodes. The deployment pattern reflects the trade-off between corridor-based electrification for long-haul flows and nodal charging opportunities for regional tours. Investment costs are based on unit assumptions of €2.0 million per kilometre of ERS (200 kW rated power, average segment length 20.1 km) and €0.20 million per SCS facility (150 kW rated power).

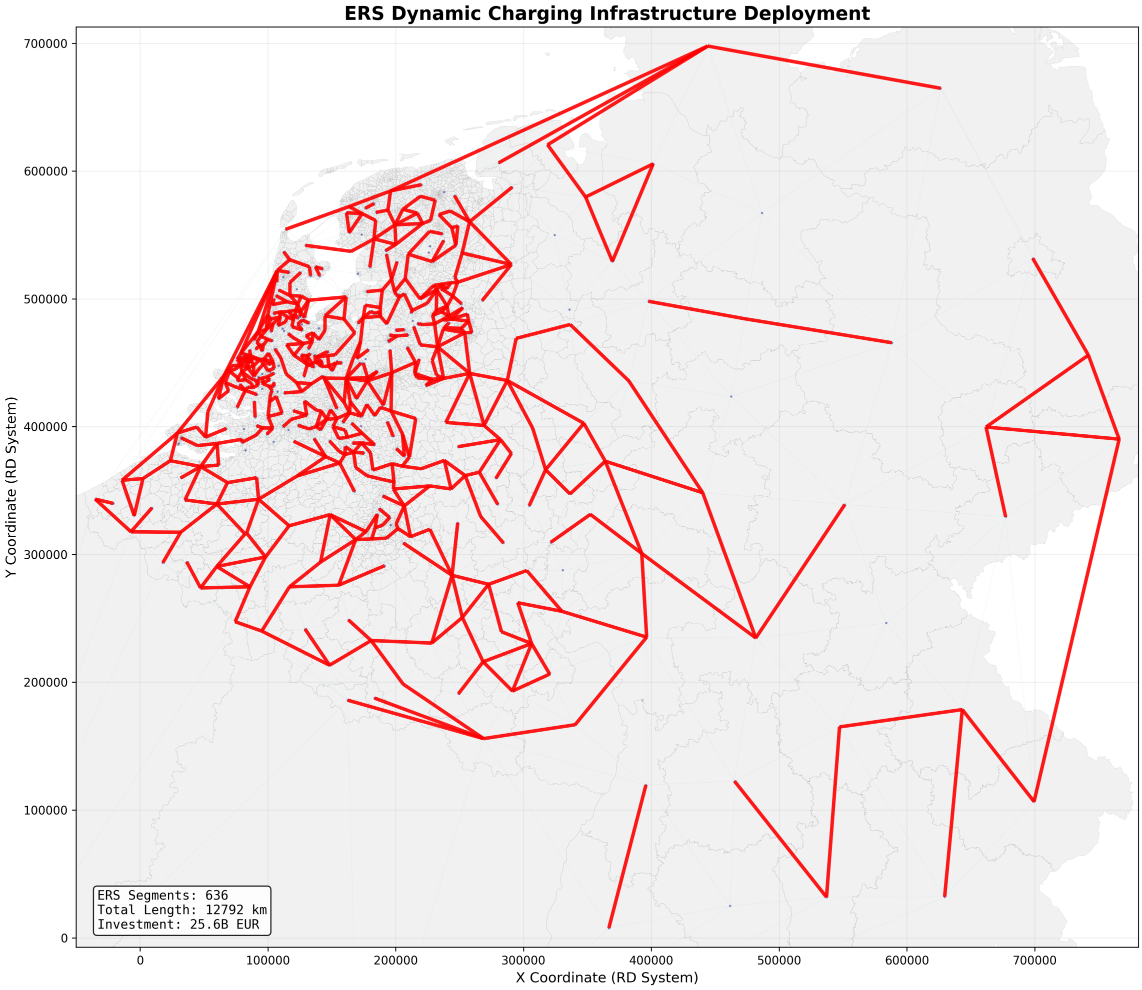


Figure 17: Optimised ERS deployment across the national freight network.

As shown in Figure 17, the GA selects **636 ERS segments** with a total electrified length of **12,792 km**, corresponding to an investment of **€25.7 billion**. The deployment is concentrated along major inter-urban and international freight corridors, forming a backbone that supports high-volume, long-distance tours. Notably, the densest electrification appears in the Randstad region and major east–west axes, consistent with observed freight demand patterns. Overall, ERS accounts for **65.6% of total CAPEX**, demonstrating that corridor electrification is the dominant investment component in the optimised strategy.

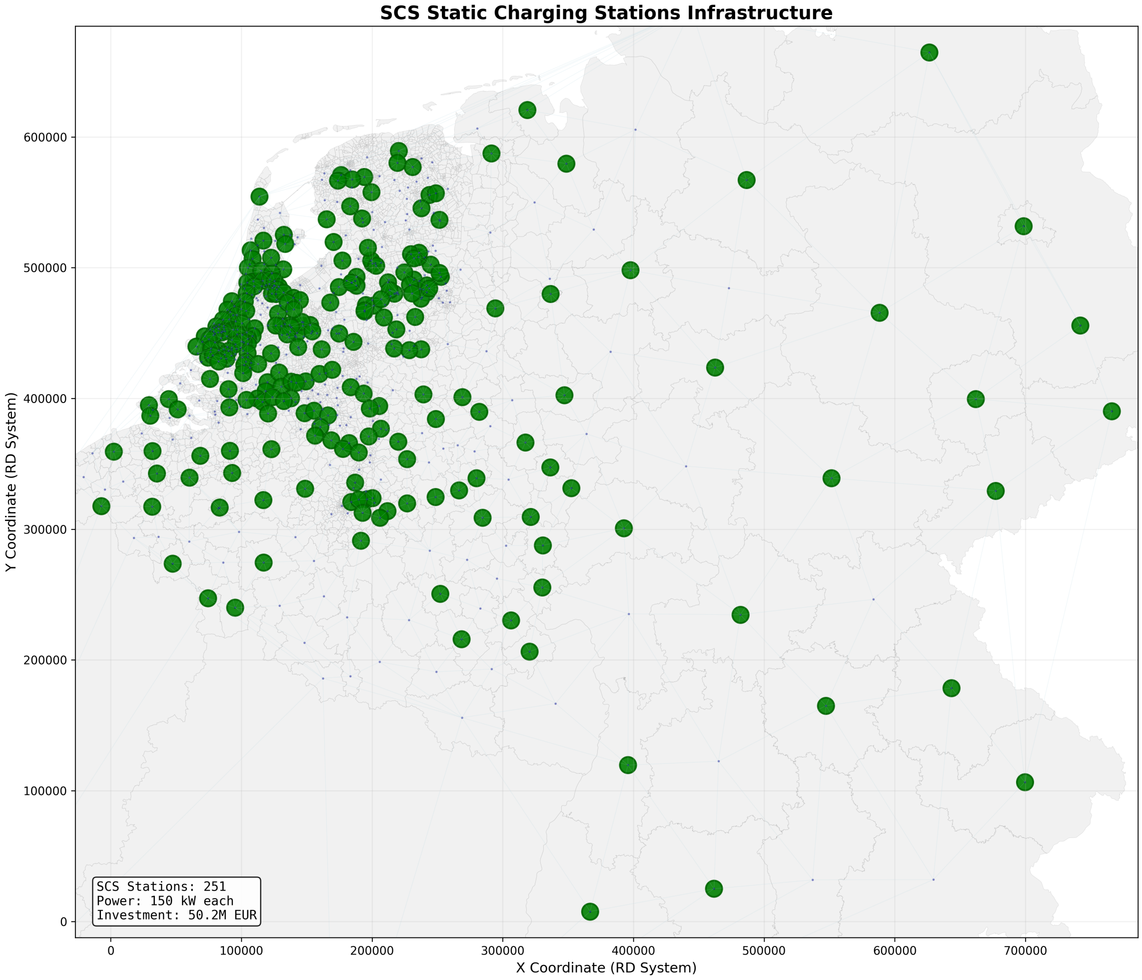


Figure 18: Optimised SCS deployment across the national freight network.

Figure 18 shows the complementary deployment of **251 SCS facilities**, each rated at 150 kW, requiring a total investment of only **€50.2 million**—less than 1% of total CAPEX. The spatial distribution of SCS nodes is notably clustered around urban and regional freight hubs, particularly in the western Netherlands, while additional coverage is provided across secondary corridors. Although financially marginal, the SCS network plays an important operational role: it provides critical redundancy and allows medium-capacity trucks to complete tours that would otherwise require outsourcing. This nodal charging network ensures that tours not fully covered by ERS can still recharge at intermediate points, thereby supporting medium-distance and regional freight operations.

Together, the ERS and SCS deployments highlight a layered investment strategy: ERS forms the long-distance electrification backbone, while SCS nodes provide redundancy and flexibility at key hubs. This integrated pattern enables high feasibility, with nearly 90% of tours electrified, while balancing capital cost with operational reliability. Importantly, the availability of ERS reduces reliance on ultra-large batteries, while SCS stations further mitigate infeasibility for tours operating at the system periphery. Nevertheless, **10.1% of tours remain infeasible**, indicating that residual service gaps persist. These will be examined in detail in Section 6.3.

6.3 Residual Infeasibility Analysis

Although the optimized deployment of ERS and SCS substantially improves tour feasibility, a non-negligible fraction of tours remain infeasible. In total, 155,749 tours (10.1% of all tours) could not be executed by BE-HDTs under the optimized infrastructure layout. This highlights that even large-scale investments cannot fully eliminate residual infeasibility.

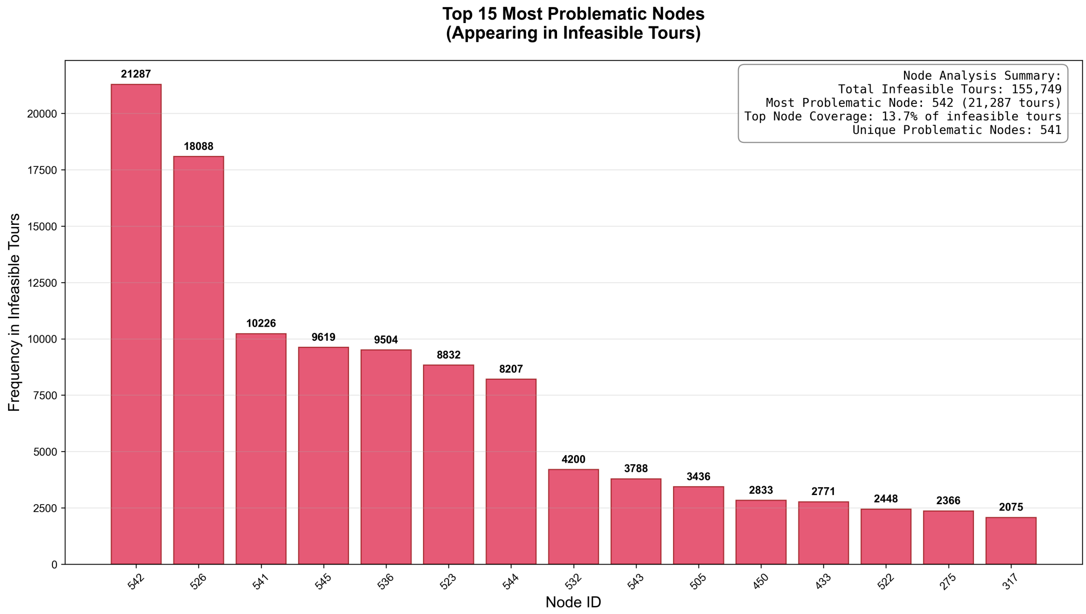


Figure 19: Top 15 most problematic nodes, ranked by frequency of appearance in infeasible tours.

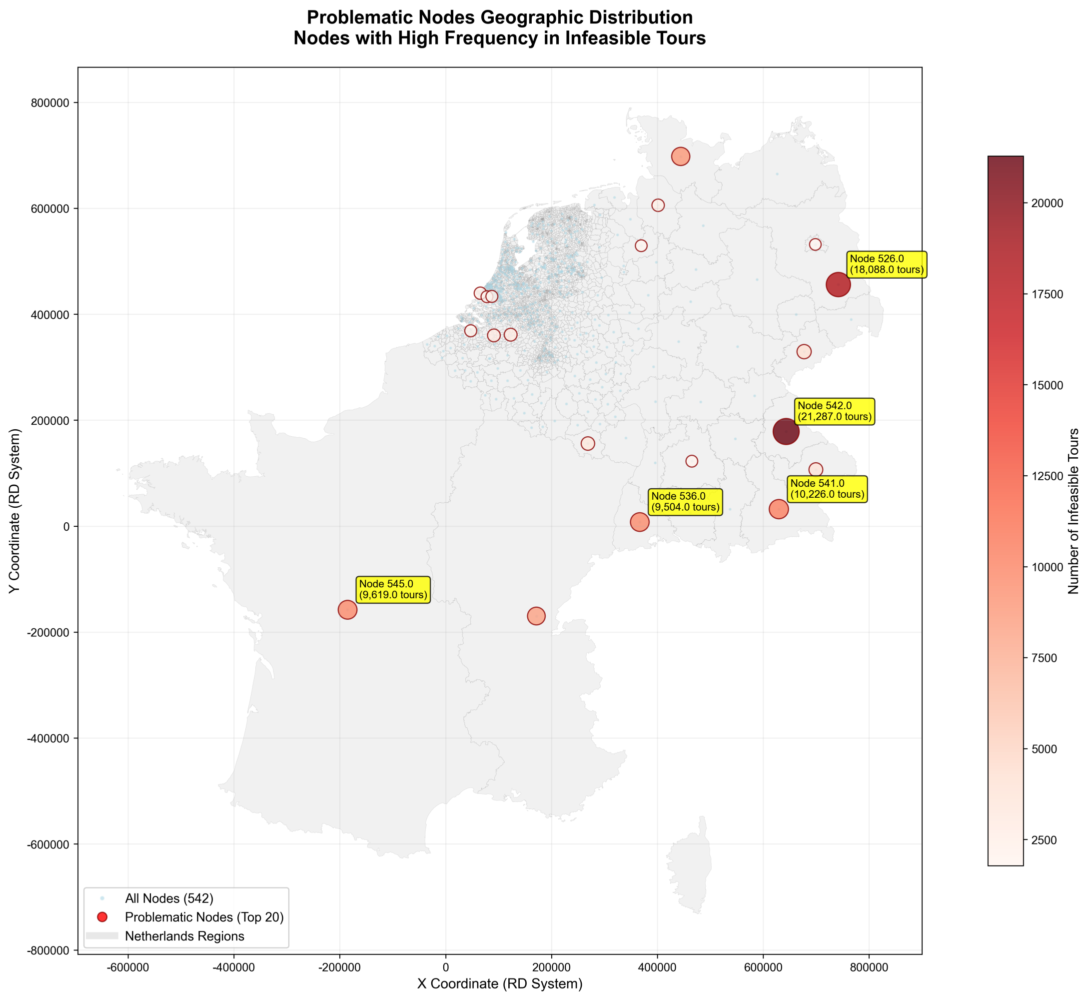


Figure 20: Geographic distribution of problematic nodes with high frequency in infeasible tours.

Figures 19 and 20 provide further insight into the distribution of these infeasible tours. A small set of nodes is responsible for a disproportionately high share of infeasibility. For instance, Node 542 alone appears in 21,287 infeasible tours, while Node 526 contributes to 18,088 cases. The top 15 nodes together account for more than 40% of all infeasible tours, illustrating the presence of structural bottlenecks in the network.

Geographically, these problematic nodes (Figure 20) are concentrated in cross-border regions and peripheral areas with sparse charging coverage. This pattern suggests that infeasibility is not uniformly distributed across the system but rather clustered in specific high-risk zones. Addressing these bottlenecks through targeted additional investments in critical nodes or corridor segments could drastically reduce the residual infeasibility rate.

At the same time, the persistence of infeasible tours despite extensive deployment underscores the necessity of fallback strategies, such as hybrid trucks or limited diesel operations, particularly in areas where electrification is prohibitively costly. This residual infeasibility analysis provides a roadmap for future incremental improvements and highlights where investments would have the greatest marginal impact.

It is important to note that part of the residual infeasibility can be attributed to the limitations of the aggregate VAM network representation. Outside the Netherlands, zone partitioning is considerably coarser, particularly in France and other neighbouring regions. For cross-border tours entering such areas, the inter-zonal distances are substantially larger than within the Dutch core network. Since the model restricts SCS deployment to VAM centroids, charging opportunities are often absent along these long-distance corridors, forcing trucks to rely solely on ERS. However, when corridors span very long distances, the exclusive reliance on ERS becomes prohibitively costly due to its high investment requirements. As a result, even with the introduction of a large infeasibility penalty parameter Ω to discourage outsourcing, certain tours remain infeasible under the current network representation. This limitation highlights that residual infeasibility is not only a function of infrastructure siting, but also of the spatial resolution of the underlying network model.

6.4 Battery Distribution and Charging Behavior

In addition to infrastructure deployment, the optimisation jointly determines the battery capacity assignment for each truck and the charging strategies adopted during operation. Figure 21 and Figure 22 summarise the resulting distributions.

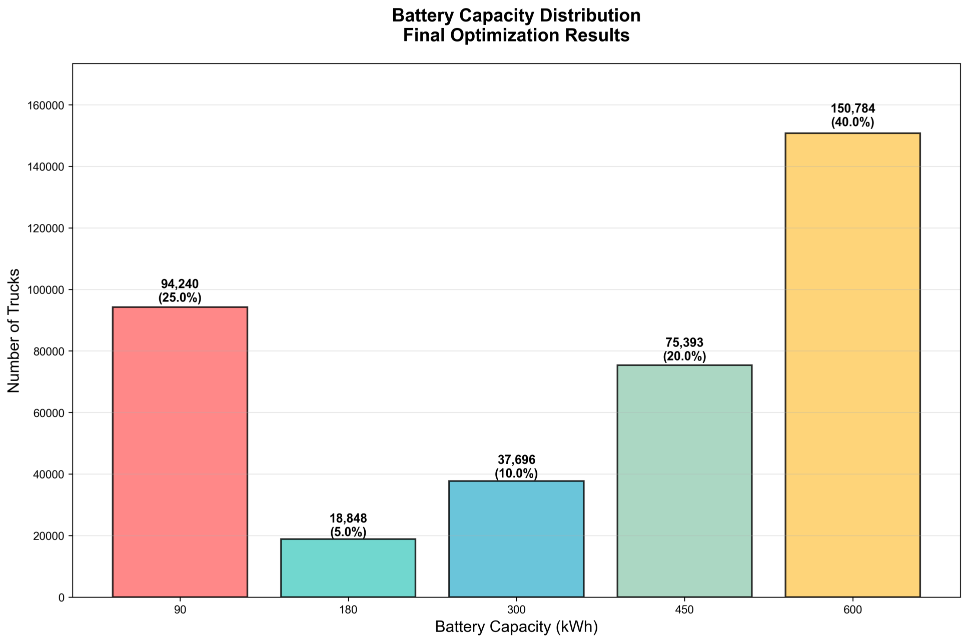


Figure 21: Distribution of battery capacities across the truck fleet.

As shown in Figure 21, the fleet is allocated across five discrete battery classes. The distribution is skewed toward the extremes: 25% of trucks are assigned the smallest capacity (90 kWh), while 40% adopt the largest class (600 kWh). Medium classes (180–450 kWh) jointly cover 35% of the fleet, with the 450 kWh option representing the largest share among them (20%). This pattern indicates that the optimisation exploits heterogeneity in tour lengths: trucks operating predominantly short tours can minimise investment with small batteries, while those serving long-distance or ERS-sparse corridors require large capacities to maintain feasibility. The prevalence of 600 kWh batteries suggests that even with extensive ERS deployment, significant on-board energy storage remains necessary for coverage. Compared to a uniform 300 kWh assignment across all trucks, this heterogeneous distribution reduces total battery CAPEX by approximately 19%, confirming the value of optimised fleet-level differentiation.

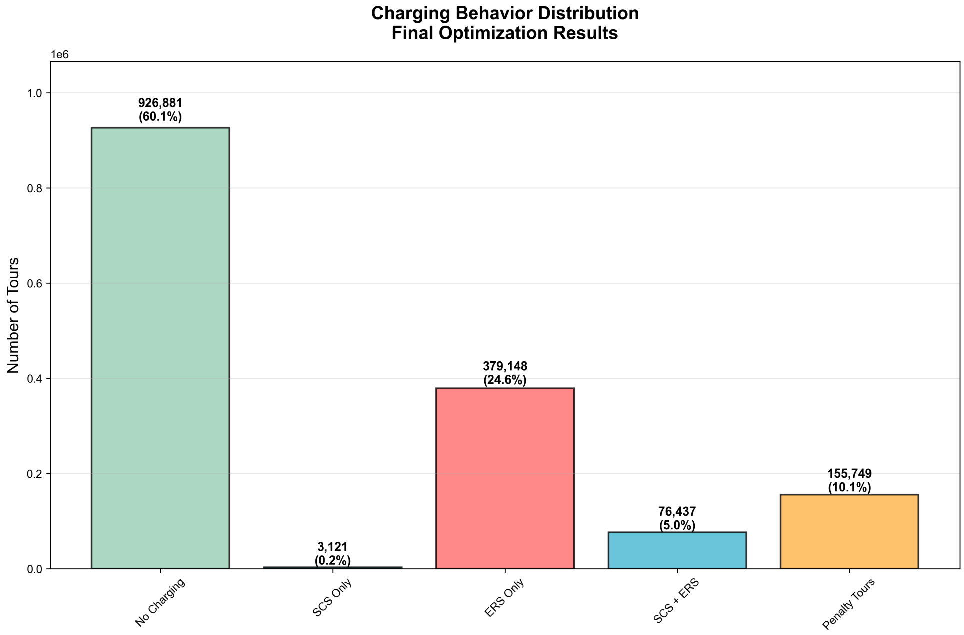


Figure 22: Distribution of charging behaviors across all tours.

Figure 22 illustrates the charging behavior at the tour level. A majority of tours (60.1%) do not require any en-route charging, reflecting both the suitability of battery sizing and the presence of ERS on critical links. Another 24.6% of tours rely exclusively on ERS, while 5.0% combine both SCS and ERS, meaning that nearly 30% of all tours directly depend on ERS for completion. Only 0.2% of tours use SCS exclusively, consistent with the model logic that static charging is triggered only when strictly necessary, since unnecessary SCS stops would add OPEX without improving feasibility. Finally, 10.1% of tours remain infeasible and are outsourced, highlighting the residual service gap even under optimised deployment.

Together, these results demonstrate the interplay between battery investment and infrastructure availability. The heterogeneous distribution of battery sizes enables cost-efficient fleet electrification, while the charging behavior outcomes underscore that ERS dominates as the backbone of long-distance electrification, with SCS serving a niche but critical backup role. The persistence of a small penalty share suggests that complete electrification would require either further infrastructure expansion or even larger on-board batteries, both of which carry additional costs. These patterns also resonate with the residual infeasibility analysis in Section 6.3: large batteries and ERS corridors together cover the majority of demand, but structural bottlenecks at system peripheries prevent full electrification.

6.5 Operating Cost Structure

A detailed breakdown of operating expenditures (OPEX) under the optimized infrastructure layout is shown in Figure 23. The total OPEX amounts to €123.8 million, which combines both feasible-tour operating costs and penalties for infeasible tours. Of this total, 54.4% (€67.3M) arises from feasible tours, while 45.6% (€56.4M) corresponds to penalties for infeasible tours.

Among the feasible tours, dynamic ERS charging dominates the cost structure, accounting for €51.2M (41.4% of total OPEX) and representing over three-quarters of feasible-tour operating costs. This confirms the central role of ERS in enabling long-haul electrification, but also highlights its associated financial burden. By contrast, static SCS charging contributes only €6.1M (4.9%), reflecting its role as a supplementary rather than primary charging option. ERS tolls (€6.6M, 5.3%) and time costs (€2.1M, 1.7%) remain relatively minor components, while SCS service fees (€1.4M, 1.1%) are negligible.

Penalty costs are split between diesel fuel and emissions surcharges. The largest single penalty component is the diesel penalty (€47.9M, 38.7%), reflecting the significant economic and environmental burden of outsourced tours. A smaller portion arises from the outsourcing penalty surcharge (€8.6M, 6.9%), designed to internalize the systemic impact of infeasibility. Together, these penalties underscore the importance of further infrastructure expansion or fallback hybridization strategies to reduce reliance on non-electrified operations.

Overall, the OPEX results highlight a dual structure: while ERS provides the backbone of cost-effective long-haul electrification, the residual penalties from infeasible tours remain substantial. This reinforces the conclusions from Section 6.2.1, suggesting that targeted investments in critical nodes or corridors could reduce penalty costs substantially and shift more tours into the ERS-based charging regime.

In addition to the aggregate distribution, the results can also be interpreted on a per-tour basis. The average cost across all tours is €80.3 per tour, with feasible tours requiring only €48.6 per tour, while infeasible tours incur an average penalty of €362.4 per tour. This stark contrast demonstrates how residual infeasibility disproportionately inflates the overall system cost, further motivating infrastructure reinforcement and policy measures to minimize the occurrence of infeasible tours.

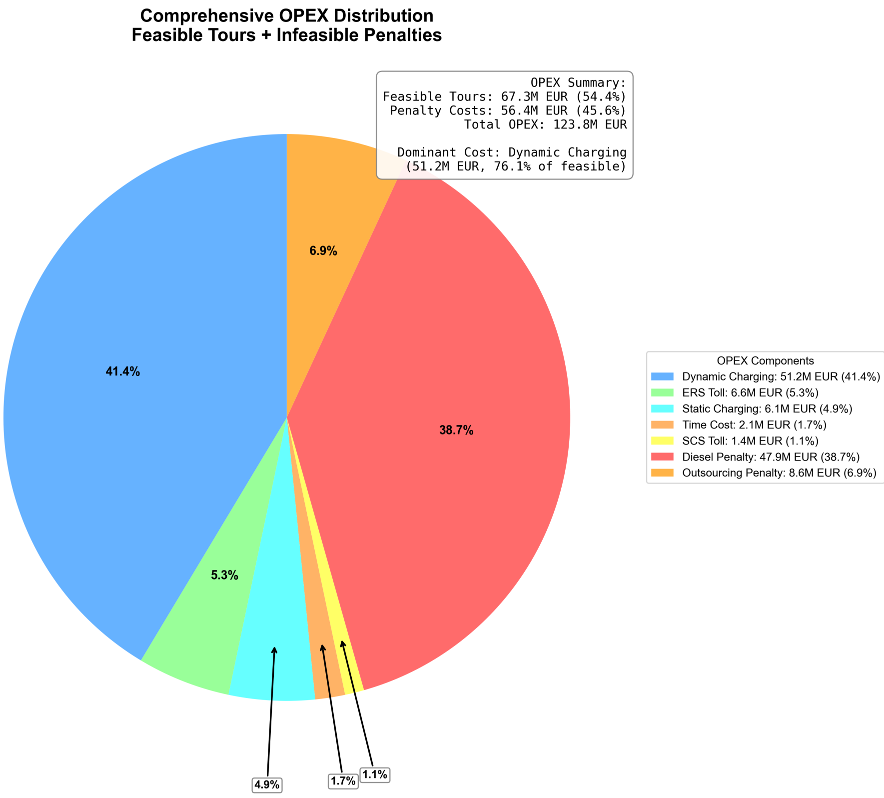


Figure 23: Comprehensive OPEX distribution for feasible tours and infeasible penalties under the optimized layout.

6.6 Summary

This chapter has presented the optimisation results of the nationwide bi-level model, highlighting how the GA effectively balances infrastructure investment, battery assignment, and operational feasibility. The overall performance analysis (Section 6.1) showed rapid convergence within fewer than 15 generations, reducing infeasible tours from 42% to 10.1% and lowering the fitness value by 34.7%. The infrastructure layout (Section 6.2) confirmed that ERS is the dominant investment component, with 12,792 km deployed at a cost of €25.7 billion (65.6% of CAPEX), complemented by 251 SCS facilities costing only €50.2 million. Despite this large-scale deployment, residual infeasibility remains concentrated in a small set of cross-border and peripheral nodes (Section 6.3), suggesting that targeted investments could yield disproportionate improvements.

Battery allocation and charging behaviour (Section 6.4) demonstrated the value of fleet heterogeneity: 25% of trucks were equipped with only 90 kWh, while 40% required 600 kWh, with medium classes covering the remainder. Compared to a uniform 300 kWh baseline, this heterogeneous assignment reduced battery CAPEX by approximately 19%. Charging patterns revealed ERS as the primary charging solution (nearly 30% of tours), while SCS served only a small backup role. The operating cost structure (Section 6.5) further emphasised this reliance: dynamic ERS charging accounted for over 40% of OPEX, while SCS contributed less than 5%. Meanwhile, penalties for infeasible tours—dominated by diesel fuel and associated CO_2 emissions—made up 46% of OPEX, underscoring both the economic and environmental cost of residual infeasibility.

Taken together, the results confirm three main insights. First, large-scale ERS deployment provides the backbone of cost-effective long-haul electrification, while SCS adds marginal but critical redundancy. Second, optimised fleet heterogeneity significantly reduces battery investment needs, highlighting the importance of integrated infrastructure–battery planning. Third, scale effects are evident: spreading infrastructure across nearly 1.5 million tours dilutes capital costs, keeping OPEX below 0.5% of total annualised system costs. Nevertheless, the persistence of infeasible tours indicates that full electrification will require either additional targeted infrastructure or fallback hybrid strategies. These findings set the stage for Chapter 7, where the results are placed in a broader policy and literature context.

7 Discussion

7.1 Interpretation of Key Findings

The nationwide optimisation results reveal several interrelated patterns that illustrate how investment, operational feasibility, and cost components interact. A central dynamic is the trade-off between capital expenditure (CAPEX) and system-wide feasibility. As shown in Chapter 6, the Genetic Algorithm (GA) deliberately increases CAPEX from €18 billion to €39.1 billion, primarily through large-scale ERS deployment and heterogeneous battery investments. While this represents more than a doubling of upfront costs, the effect is a sharp reduction in infeasible tours (from 42% to 10.1%) and a corresponding improvement in system feasibility by 31.9 percentage points. This pattern underscores that higher investment is justified by the long-term savings it unlocks in operating expenditure (OPEX) and penalty costs, highlighting the substitution of recurring costs with upfront infrastructure.

The battery assignment outcomes further reinforce this logic. The average installed capacity converges to 357 kWh per truck, closely aligned with ElaadNL's reported daily energy requirement of 289.5 kWh per heavy-duty vehicle. This alignment suggests that the optimisation framework captures realistic energy demand profiles. At the same time, the heterogeneous allocation across five classes demonstrates that electrification need not rely exclusively on maximum-capacity batteries: while 40% of the fleet requires 600 kWh packs to cover long or ERS-sparse tours, a substantial share operates with much smaller batteries, thereby reducing over-investment and lowering total battery CAPEX by roughly 19% compared to a uniform-capacity baseline.

Infrastructure deployment patterns further illustrate the differentiated roles of SCS and ERS. ERS forms the long-haul electrification backbone, accounting for 65.6% of total CAPEX and directly enabling nearly 30% of tours to rely on in-motion charging. SCS, by contrast, accounts for less than 1% of CAPEX, but plays a critical role in reducing infeasibility at regional hubs and peripheral corridors. This layered deployment ensures that the system combines efficiency on high-volume corridors with redundancy in secondary locations. Nevertheless, infeasible tours remain concentrated around cross-border and low-density nodes, highlighting the persistent challenge of spatial bottlenecks.

The cost breakdown also provides broader insights into the system's economic and environmental dimensions. OPEX represents less than 0.5% of total annualised cost, confirming that infrastructure investment dominates the economics of electrification. Within OPEX, dynamic ERS charging constitutes the majority of feasible-tour costs, while penalty expenditures for infeasible tours—especially diesel fuel and associated CO_2 emissions—remain substantial. This confirms that residual infeasibility not only represents an economic inefficiency but also an environmental liability. At the same time, scale effects are evident: with 1.53 million tours electrified, each installed facility serves over 5,000 tours on average, allowing CAPEX to be diluted across a wide demand base and reinforcing the long-term feasibility of large-scale electrification.

Overall, the key findings illustrate a consistent mechanism: higher upfront CAPEX in ERS and batteries reduces infeasibility and operating costs, producing a lower total system cost. The results align closely with empirical energy demand benchmarks, highlight the importance of heterogeneous battery strategies, and demonstrate that scale effects further strengthen economic viability. These insights provide the foundation for the subsequent comparison with literature and the policy implications discussed in Sections 7.2 and 7.3.

7.2 Trade-offs Between Cost, Feasibility, and Investment

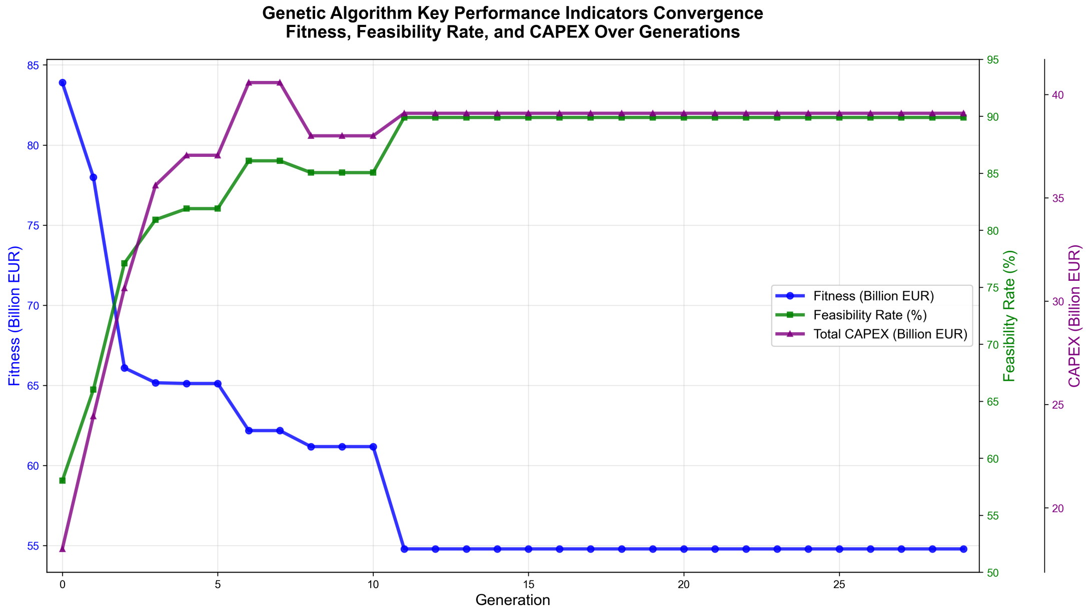


Figure 24: Joint convergence of fitness, feasibility rate, and CAPEX across generations.

Figure 24 integrates the convergence of three key performance indicators—fitness, feasibility rate, and CAPEX—to illustrate the internal trade-offs of the optimisation process. The trajectories reveal a consistent pattern of interaction. The best fitness value (blue) decreases sharply from €83.9 billion to €54.8 billion within the first 12 generations, corresponding to a 34.7% improvement. This decline coincides with a steep rise in feasibility (green), which increases from 58% to nearly 90%. In contrast, total CAPEX (purple) more than doubles, from €18 billion to €39 billion, reflecting the investment required to achieve such gains in feasibility.

This trade-off highlights the internal mechanism of the optimisation: infeasible tours can only be reduced substantially by deploying large-scale ERS corridors, complemented with strategically placed SCS, and equipping part of the fleet with larger batteries. While these measures sharply increase capital expenditures, they simultaneously reduce penalty costs and operating expenses, producing a lower overall system cost. In other words, the GA converges towards solutions where higher upfront investments enable more tours to be served electrically, thereby lowering recurring operational burdens.

Overall, the combined convergence illustrates a key insight: *low-capex solutions are not cost-optimal once feasibility is accounted for.* Instead, the optimum arises from accepting higher investment costs in exchange for systemic reductions in OPEX and infeasibility penalties, resulting in a substantially lower long-term fitness value. This demonstrates the central role of investment–feasibility interactions in shaping the cost-effectiveness of nationwide freight electrification.

7.3 Comparison with Literature

The findings of this study can be positioned within the growing body of research on heavy-duty truck electrification. Industry-oriented studies, such as the ElaadNL Outlook Logistiek (2025), primarily focus on aggregate projections of fleet penetration, annual mileage, and electricity demand. ElaadNL reports an average annual mileage of 68,698 km and an energy intensity of 1.1 kWh/km, corresponding to a daily energy requirement of 289.5 kWh per truck. The present model yields an average assigned battery capacity of 357 kWh, closely aligned with this benchmark, indicating that the optimisation results are consistent with realistic Dutch freight operations. At the same time, the *tour-level* results presented here provide a finer granularity than ElaadNL, identifying a heterogeneous battery allocation across five classes and quantifying infeasible tours (10.1%), which are not captured in purely aggregate outlooks [11].

In the academic domain, Inez (2024) proposed a bi-level framework that couples infrastructure siting with operational feasibility. However, that work remained at the *trip level*, without explicitly modelling battery capacity assignment across complete tours. The present study extends this formulation by adopting a *tour-level perspective*, ensuring that battery allocation is consistent across all trips belonging to the same truck. This methodological step is crucial, since feasibility depends not only on isolated segments but on the cumulative state-of-charge across multiple trips within a tour. Moreover, the inclusion of infeasibility penalties enables the model to highlight structural bottlenecks that would remain invisible in trip-based formulations, while the use of a nationwide dataset with over 1.5 million tours demonstrates tractability at scale [19].

Similarly, Liao et al. (2024) examined the deployment of ERS across a trans-European network and demonstrated the potential to reduce average battery sizes from 370 kWh to as little as 90 kWh under full electrification. While the findings of this thesis are consistent with Liao's conclusion that ERS serves as the backbone for long-distance electrification, this study adds value by explicitly modelling tours rather than trips. This allows the identification of trucks that still require very large batteries (600 kWh) to maintain feasibility in ERS-sparse regions, while also decomposing costs into CAPEX, OPEX, and infeasibility penalties, thereby offering a more integrated view of the system trade-offs [24]. Furthermore, the explicit incorporation of SCS demonstrates their supplementary role in reducing infeasibility at regional hubs, despite their minimal CAPEX share.

Taken together, this thesis complements both industry and academic literature. Relative to ElaadNL, it provides micro-level feasibility insights that inform targeted interventions at bottleneck nodes and corridors. Relative to Inez (2024) and Liao et al. (2024), its main contribution lies in advancing from trip-level to *tour-level* modelling, thereby ensuring realistic consistency in battery allocation and uncovering systemic infeasibility patterns at scale. These contributions strengthen the evidence base for policy decisions on cost-effective freight electrification pathways in the Netherlands and beyond.

7.4 Limitations

While the results provide valuable insights into nationwide electrification pathways, several limitations of the modelling framework should be acknowledged. First, the study relies on an aggregate network representation based on VAM zones, and all truck trips are mapped onto fixed shortest-path routes. This assumption neglects the heterogeneity of real-world routing, where carriers may deviate from shortest paths due to congestion, tolls, delivery constraints, or charging opportunities. As a result, actual operational flexibility is not fully captured.

Second, infeasibility is handled through the introduction of a large penalty parameter Ω, which discourages outsourcing by strongly penalising infeasible tours. Although this mechanism effectively reduces infeasibility in the optimisation process, it represents a modelling device rather than a direct economic cost, and thus may bias the solution towards more electrified outcomes than would arise under true market conditions.

Third, the model necessarily incorporates simplifying assumptions to maintain tractability. Charging station capacity is treated as unlimited, with no consideration of queuing or power grid constraints; all tours are assumed to start with fully charged batteries; electricity and toll prices are assumed time-invariant; and battery degradation and replacement are excluded. These assumptions omit important temporal and operational dynamics that would influence the real-world feasibility and cost of freight electrification.

Taken together, these limitations suggest that while the optimisation framework provides useful strategic insights, the quantitative results should be interpreted with caution. Future work could relax these assumptions by integrating stochastic demand, dynamic routing, grid-capacity constraints, and long-term battery degradation into the modelling framework.

7.5 Future Work

Building on the limitations identified above, several promising directions for future research can be outlined. First, the reliance on an aggregate VAM network with fixed shortest-path routing should be relaxed. Future studies could employ higher-resolution zoning, particularly in peripheral and cross-border regions such as France, where coarse partitions exaggerate inter-zonal distances and constrain the flexibility of charging deployment. Incorporating more realistic network representations, including secondary corridors and multiple feasible routes, would allow charging opportunities to be modelled in line with actual freight operations.

Second, the current modelling framework uses a large infeasibility weight parameter (Ω) to discourage outsourcing. While effective in guiding the optimisation, this remains a heuristic device rather than a reflection of true market costs. Future work could refine this mechanism by calibrating penalty functions against empirical outsourcing prices, CO_2 taxation schemes, or hybrid fleet operations, thereby improving the behavioural realism of infeasibility treatment.

Third, several simplifying assumptions should be relaxed. In particular, the assumption of unlimited charging capacity at SCS and ERS facilities overlooks potential queuing and grid constraints. Extending the model to capture congestion effects, time-of-use electricity pricing, and stochastic demand would better reflect operational realities. Similarly, battery ageing and replacement dynamics could be incorporated to assess long-term investment implications.

Finally, advancing the solution approach itself offers an avenue for improvement. While the GA has demonstrated strong performance at national scale, hybrid methods that combine metaheuristics with exact solvers or decomposition strategies could further improve scalability and solution quality. Embedding uncertainty analysis and sensitivity experiments would also strengthen the robustness of the policy insights derived from the optimisation.

Taken together, these future research directions would enable more realistic, granular, and policy-relevant assessments of freight electrification strategies, bridging the gap between theoretical optimisation and practical deployment.

8 Conclusion

8.1 Summary of the Research

This thesis has developed and applied a nationwide optimisation framework for planning the electrification of heavy-duty road freight transport in the Netherlands. The framework is formulated as a bi-level model that integrates three interdependent decision layers: the siting of static charging stations (SCS), the deployment of electrified road segments (ERS), and the assignment of battery capacities to individual trucks. To address the computational intractability of solving this model exactly at national scale, a Genetic Algorithm (GA) was designed, enabling the optimisation of 1.53 million tours while retaining operational feasibility constraints.

The model was operationalised using large-scale truck trajectory data, aggregated to the Dutch VAM network. Each tour was evaluated against energy balance, charging opportunities, and battery constraints, with infeasibility penalised through a large penalty factor to reflect the economic and environmental cost of outsourcing. The results demonstrate clear improvements in nationwide electrification feasibility: the share of electrifiable tours increased from 58% in the initial generation to 89.9% after optimisation, reducing the infeasible fraction to 10.1%. At the same time, the overall system fitness improved by 34.7%, from €83.9 billion to €54.8 billion.

The optimisation further revealed distinct investment and operational patterns. Total CAPEX rose from €18 billion to €39.1 billion, with €25.7 billion invested in ERS (65.6% of CAPEX, 12,792 km deployed) and €50.2 million in SCS (251 stations, less than 1% of CAPEX). Battery investments accounted for €13.5 billion, with capacity assignments distributed heterogeneously: 25% of trucks received 90 kWh packs, 40% were assigned 600 kWh packs, and the remainder occupied intermediate classes. The average capacity converged to 357 kWh, closely aligned with empirical benchmarks of daily energy demand (289.5 kWh per truck). On the operational side, OPEX remained relatively small at €124 million (less than 0.5% of total annualised cost), dominated by ERS charging (41.4%) and diesel penalties for infeasible tours (€47.9 million).

In sum, the study demonstrates that nationwide electrification is feasible under a layered strategy where ERS provides the long-haul backbone, SCS delivers regional redundancy, and heterogeneous batteries reduce fleet-level investment costs. While infeasibility persists in peripheral and cross-border regions, the results highlight both the potential and the structural challenges of achieving cost-effective freight decarbonisation at scale.

8.2 Main Contributions

This thesis makes several contributions to the methodological literature, empirical evidence, and academic debate on the electrification of heavy-duty road freight. From a methodological perspective, the study advances existing bi-level optimisation frameworks by explicitly integrating truck-level battery assignment into the upper-level decision space. This ensures that each truck is consistently allocated to a single capacity class, avoiding the unrealistic assumption in earlier models that batteries could be reselected on a per-trip basis. A distinctive feature of this work is the adoption of a *tour-level* formulation, as opposed to the trip-level models prevalent in the literature. By operating at the tour level, the framework more realistically captures operational feasibility, since feasibility depends on the energy trajectory across multiple linked trips rather than isolated segments. This is a major departure from Inez (2024) and Liao et al. (2024), both of which optimise at the trip level without ensuring consistency across tours.

A further methodological innovation lies in the explicit modelling of infeasibility. By introducing a large penalty parameter (Ω), the framework reduces outsourcing and internalises the systemic costs of unserved tours, while retaining tractability at scale. The optimisation is solved nationwide using a Genetic Algorithm, capable of handling over 1.5 million tours. This scale demonstrates that heuristic methods can overcome the computational intractability of exact MILP approaches, while retaining the ability to explore infrastructure–battery trade-offs in a realistic national setting. The model also decomposes costs into CAPEX (ERS, SCS, battery), OPEX (electricity, tolls, time), and penalty terms (diesel, CO_2, outsourcing surcharge), enabling a richer interpretation of trade-offs.

Empirically, the results provide novel insights into nationwide electrification. The model raises tour feasibility from 58% to 89.9% and reduces system fitness by 34.7%, highlighting the efficiency gains of integrated planning. ERS deployment dominates as the long-haul backbone (12,792 km, €25.7 billion), complemented by SCS as a low-cost redundancy (€50.2 million, 251 sites). Battery assignment is heterogeneous, with 25% of trucks using 90 kWh and 40% requiring 600 kWh, producing an average of 357 kWh that aligns with ElaadNL's benchmark of 289.5 kWh/day. This heterogeneity reduces battery CAPEX by roughly 19% relative to a uniform-capacity baseline. Nevertheless, 10.1% of tours remain infeasible, concentrated in cross-border and peripheral zones, where sparse charging coverage and aggregate network limitations constrain feasibility. Importantly, the scale effects are pronounced: with 1.53 million tours electrified, each installed facility serves more than 5,000 tours, spreading CAPEX over a wide demand base.

Relative to the literature, this thesis complements and extends prior studies. Compared to Inez (2024), it advances the bi-level framework by integrating heterogeneous battery assignment and explicit infeasibility penalties. Compared to Liao et al. (2024), it confirms ERS as the backbone of electrification but demonstrates the supplementary role of SCS and quantifies systemic infeasibility. Compared to ElaadNL (2025), it moves beyond aggregate projections by offering tour-level insights into feasibility, battery allocation, and problematic nodes. Taken together, these contributions establish a comprehensive, large-scale, and operationally realistic framework that provides both methodological innovation and policy-relevant evidence for cost-effective freight electrification.

8.3 Future Research Directions

Building upon the limitations identified in this study, several promising directions for future research can be outlined. First, the reliance on an aggregate VAM-based network constitutes a key simplification: while it enables nationwide tractability, it inevitably introduces distortions in corridor representation, particularly for cross-border routes such as long-haul flows into France. Future work should therefore employ more detailed road networks and node representations, which would reduce infeasibility artefacts arising from overly coarse aggregation.

Second, the assumption of fixed shortest-path routing neglects alternative route choice and stochasticity in freight movements. Incorporating flexible routing, demand variability, and dynamic re-routing into the optimisation framework would provide a more realistic assessment of tour feasibility under real-world operating conditions.

Third, infeasibility was handled here through a large penalty weight Ω, which effectively prioritises feasible electrification. While this mechanism succeeds in reducing infeasible tours, future studies could investigate multi-objective optimisation approaches or explicitly model hybrid/diesel fallback vehicles as part of the fleet, yielding more nuanced trade-offs between electrification, outsourcing, and system cost.

Fourth, operational realism in charging infrastructure remains an open challenge. This study assumed unlimited charging capacity at both SCS and ERS facilities, omitting queuing dynamics, charging time restrictions, and local grid constraints. Extending the model to capture such operational bottlenecks would substantially strengthen its applicability for infrastructure planning.

Finally, the framework could be enriched by explicitly linking infrastructure deployment to policy and market instruments. Incorporating carbon pricing, subsidy schemes, or differentiated tolls would enable scenario analysis of how policy incentives shift the balance between ERS, SCS, and battery investment. Such extensions would provide valuable insights for policymakers and industry stakeholders seeking cost-effective and resilient freight electrification strategies.

References

[1] Milad Akbari, Morris Brenna, and Michela Longo. Optimal locating of electric vehicle charging stations by application of genetic algorithm. *Sustainability*, 10(4):1076, 2018.

[2] Md Rakibul Alam and Zhaomiao Guo. A review on the charging station planning and fleet operation for electric freight vehicles. *arXiv preprint arXiv:2311.16419*, 2023.

[3] Jeroen Boerkamps and Arjan Van Binsbergen. Goodtrip–a new approach for modelling and evaluation of urban goods distribution. *City logistics*, 1, 1999.

[4] Lesley Brown. Electric road systems, towards decarbonising road transport. https://futuramobility.org/en/%F0%9F%93%8C-electric-road-systems-towards-decarbonising-road-transport/, August 2024. Accessed: 2025-09-09.

[5] Cargopedia. European fuel prices. https://www.cargopedia.net/europe-fuel-prices, 2025. Accessed: 2025-08-24.

[6] Christian Cintrano, Jamal Toutouh, and Enrique Alba. Citizen centric optimal electric vehicle charging stations locations in a full city: case of malaga. In *Conference of the Spanish Association for Artificial Intelligence*, pages 247–257. Springer, 2021.

[7] Michiel de Bok, Bart Wesseling, Adrián Estrada, Dick Bakker, and Peter Mijjer. De bestelauto in modellen voor verkeer- en vervoer: tussen wal en schip? https://significance.nl/wp-content/uploads/2019/03/2017-MDB-De-bestelauto-in-modellen-voor-verkeer-en-vervoer-tussen-wal-en-schip.pdf, 2017. Accessed: 2025-08-24.

[8] Christopher de Saxe, Daniel Ainalis, John Miles, Phil Greening, Adam Gripton, Christopher Thorne, and David Cebon. Vehicle and charging requirements for an electrified road freight system demonstrator in the uk. *Transportation Research Procedia*, 72:4033–4040, 2023.

[9] Decisio and EVConsult and Sweco. Cost-effectiveness analysis electric road systems (ers) for the netherlands. Technical report, Decisio, EVConsult, Sweco, March 2022. Accessed: 2025-09-09.

[10] Sania E Seilabi, Mohammadhosein Pourgholamali, Mohammad Miralinaghi, Gonçalo Homem de Almeida Correia, Zongzhi Li, and Samuel Labi. Sustainable planning of electric vehicle charging stations: A bi-level optimization framework for reducing vehicular emissions in urban road networks. *Sustainability*, 17(1):1, 2025.

[11] ElaadNL. Outlook logistiek: Elektrisch geladen – update 2025. Technical report, ElaadNL, January 2025. Accessed: 2025-09-20.

[12] Tap Electric. How much does it cost to charge an electric car in the netherlands in 2025? https://www.tapelectric.app/blog/cost-to-charge-an-electric-car-netherlands/, 2023. Accessed: 2025-08-24.

[13] Heliox Energy. Rapid 150 kw ev charging station. https://www.heliox-energy.com/products/rapid-150kw-modular, 2025. Accessed: 2025-08-24.

[14] ETSC. Europe-wide truck speed limits around 80 km/h. https://etsc.eu/speed-data/, 2024. Accessed: 2025-08-24.

[15] European Commission. Heavy-duty vehicles. https://climate.ec.europa.eu/eu-action/transport-decarbonisation/road-transport/heavy-duty-vehicles_en. Accessed: 2025-08-23.

[16] Carlos A Gonzalez-Calderon and José Holguín-Veras. Entropy-based freight tour synthesis and the role of traffic count sampling. *Transportation Research Part E: Logistics and Transportation Review*, 121:63–83, 2019.

[17] Mikael Hellgren and Nicholas Honeth. Efficiency of an ac conductive in-road charging system for electric vehicles-analysis of pilot project data. *SAE International Journal of Electrified Vehicles*, 9(1):27–40, 2020.

[18] Honda R&D Co., Ltd. Honda electric road system – charging power capabilities. https://global.honda/en/safety/ITS/2021/pdf/07_Honda_Electric_Road_System.pdf, 2021. ERS dynamic charging power up to 450 kW; Accessed: 2025-08-24.

[19] Inez. Combination of static and dynamic charging facilities for road freight electrification, 2024. Working paper.

[20] International Council on Clean Transportation (ICCT). Comparison of fuel consumption and emissions for representative hdvs in europe. https://theicct.org/wp-content/uploads/2021/06/HDV-EU-fuel-consumption_ICCT-Fact-Sheet_08042018_vF.pdf, 2018. Typical tractor-trailer fuel consumption: 32.6 L/100 km; Accessed: 2025-08-24.

[21] Dusica Joksimovic and Rik van Grol. New generation of dutch national and regional models—an overview of theory and practice. In *European Transport Conference*, 2012.

[22] Patcharida Kunawong, Warisa Nakkiew, Parida Jewpanya, and Wasawat Nakkiew. Optimizing electric bus charging infrastructure: A bi-level mathematical model for strategic station location and off-board charger allocation in transportation network. *Mathematics (2227-7390)*, 13(5), 2025.

[23] Danny Lewis. Recharge as you drive? the future of evs could be wireless. https://www.wsj.com/podcasts/wsj-the-future-of-everything/recharge-as-you-drive-the-future-of-evs-could-be-wireless/d06c0f9e-9d7a-40af-803e-8bcb32a924aa, March 2024. Wall Street Journal Podcast, Bold Names. Accessed: 2025-09-09.

[24] Ximeng Liao, Mahnam Saeednia, Maria Nogal, and Lori Tavasszy. Scaling up dynamic charging infrastructure: Significant battery cost savings. *Transportation Research Part D: Transport and Environment*, 129:104128, 2024.

[25] Amy Moore. Compendium of tour-based freight modeling literature. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2017.

[26] DG MOVE. Update of the handbook on external costs of transport. *Report for the*, 2014.

[27] Chris Nelder and Emily Rogers. Reducing ev charging infrastructure costs. *Rocky Mountain Institute*, 10, 2019.

[28] NL Times. Electric car charging costs vary drastically across dutch cities. https://nltimes.nl/2024/03/11/electric-car-charging-costs-vary-drastically-across-dutch-cities, 2024. Accessed: 2025-08-24.

[29] Johanna Olovsson, Maria Taljegard, Michael Von Bonin, Norman Gerhardt, and Filip Johnsson. Impacts of electric road systems on the german and swedish electricity systems—an energy system model comparison. *Frontiers in Energy Research*, 9:631200, 2021.

[30] Emma Piedel, Enrico Lauth, Alexander Grahle, and Dietmar Göhlich. Review and evaluation of automated charging technologies for heavy-duty vehicles. *World Electric Vehicle Journal*, 15(6), 2024.

[31] PTV Logistics. Introduction to truck toll costs in logistics. https://www.ptvlogistics.com/en/solutions/truck-toll-cost, 2025. Accessed: 2025-08-24.

[32] Jakob Rogstadius, Hampus Alfredsson, Henrik Sällberg, and Karl-Filip Faxén. Electric road systems: A no-regret investment with policy support, 2023. Unpublished manuscript.

[33] Jakob Rogstadius, Hampus Alfredsson, Henrik Sällberg, and Karl-Filip Faxén. Correcting market failure for no-regret electric road investments under uncertainty. *Nature Communications*, 16(1):1–19, 2025.

[34] Iván Sánchez-Díaz, José Holguín-Veras, and Xuegang Jeff Ban. A time-dependent freight tour synthesis model. *Transportation Research Part B: Methodological*, 78:144–168, 2015.

[35] Lithium-Ion Battery Pack Prices See. Largest drop since 2017, falling to $115 per kilowatt-hour: Bloombergnef| bloombergnef, 2025.

[36] Wasim Shoman, Sten Karlsson, and Sonia Yeh. Benefits of an electric road system for battery electric vehicles. *World Electric Vehicle Journal*, 13(11):197, 2022.

[37] Wasim Shoman, Sonia Yeh, Frances Sprei, Patrick Plötz, and Daniel Speth. Battery electric long-haul trucks in europe: Public charging, energy, and power requirements. *Transportation Research Part D: Transport and Environment*, 121:103825, 2023.

[38] AG Siemens. Siemens mobility gmbh (2017). ehighway solutions for electrified road freight transport, 2020.

[39] Siemens AG. ehighway: Electrified heavy duty road transport, 2018.

[40] Xiaotong Sun, Zhibin Chen, and Yafeng Yin. Integrated planning of static and dynamic charging infrastructure for electric vehicles. *Transportation Research Part D: Transport and Environment*, 83:102331, 2020.

[41] Sebastiaan Thoen, Lóránt Tavasszy, Michiel de Bok, Goncalo Correia, and Ron van Duin. Descriptive modeling of freight tour formation: A shipment-based approach. *Transportation Research Part E: Logistics and Transportation Review*, 140:101989, 2020.

[42] Mohammad M Vazifeh, Hongmou Zhang, Paolo Santi, and Carlo Ratti. Optimizing the deployment of electric vehicle charging stations using pervasive mobility data. *Transportation Research Part A: Policy and Practice*, 121:75–91, 2019.

[43] Wikipedia. Carbon price. `https://en.wikipedia.org/wiki/Carbon_price`, 2025. Accessed: 2025-08-24.

[44] Wikipedia. Lkw-maut (heavy goods vehicle toll in germany). `https://en.wikipedia.org/wiki/LKW-Maut`, 2025. Accessed: 2025-08-24.

[45] Wikipedia contributors. Truck. `https://en.wikipedia.org/wiki/Truck`. Accessed: 2025-08-23.

[46] Wikipedia contributors. Electric road. `https://en.wikipedia.org/wiki/Electric_road`, 2025. Accessed: 2025-09-09.

[47] Wikipedia contributors. Ground-level power supply. `https://en.wikipedia.org/wiki/Ground-level_power_supply`, 2025. Accessed: 2025-09-09.

[48] Wikipedia contributors. Swedish transport administration electric road program. `https://en.wikipedia.org/wiki/Swedish_Transport_Administration_electric_road_program`, 2025. Accessed: 2025-09-09.

[49] Bo Zeng, Houqi Dong, Ramteen Sioshansi, Fuqiang Xu, and Ming Zeng. Bilevel robust optimization of electric vehicle charging stations with distributed energy resources. *IEEE Transactions on Industry Applications*, 56(5):5836–5847, 2020.

A Appendix

A.1 Main File

```
# -*- coding: utf-8 -*-
import os, time, json
import pandas as pd
from datetime import datetime
from typing import Dict, List, Any, Set, Tuple

from bilevel_milp_model import EnhancedBilevelEvaluator
from data_loader import load_all_from_csv
from genetic_algorithm import GroupedGeneticAlgorithm


CONFIG: Dict = {
    "alpha_init_soc": 1.0,
    "beta_kwh_per_km": 1.6,
    "v_avg_kmh": 60.0,
    "P_SCS_kw": 150.0,
    "P_ERS_kw": 200.0,

    "price_stat_eur_per_kwh": 0.73,
    "price_dyn_eur_per_kwh": 0.36,
    "toll_scs_eur_per_km": 0.15,
    "toll_ers_eur_per_km": 0.10,
    "c_time_eur_per_h": 38.0,

    "capex_scs_eur": 200000.0,
    "capex_ers_eur_per_km": 2000000.0,
    "battery_cost_eur_per_kwh": 100.0,

    "battery_classes_kwh": [90, 180, 300, 450, 600],
    "penalty": {
        "diesel_eur_per_l": 1.6,
        "diesel_l_per_km": 0.35,
        "co2_eur_per_g": 0.00008,
        "outsourcing_eur_per_km": 0.10,
    },
    "omega": 1e5,
    "min_feasible_rate": 0.70,
    "big_penalty_for_rate": 1e9,

    "ga_pop_size": 30,
    "ga_generations": 30,
    "ga_elite_size": 5,
    "ga_mutation_rate": 0.15,
    "ga_crossover_rate": 0.80,
    "ga_tournament_k": 3,
    "init_prob_scs": 0.4,
    "init_prob_ers": 0.1,
    "min_hamming_ratio": 0.12,
    "ga_restart_every": 12,
    "truck_battery_policy": "genetic_optimized",
    "battery_margin": 0.15,
}


class EnhancedLogger:
    def __init__(self, out_dir="results_enhanced"):
        self.dir = out_dir
        os.makedirs(out_dir, exist_ok=True)
        self.st = datetime.now()
```

```python
        self.tag = self.st.strftime("%Y%m%d_%H%M%S")
        self.path = os.path.join(out_dir, f"log_enhanced_{self.tag}.txt")
        self._w("=" * 70)
        self._w("Enhanced Charging Infrastructure Optimization")
        self._w("=" * 70)

    def _w(self, s: str):
        print(s)
        with open(self.path, "a", encoding="utf-8") as f:
            f.write(s + "\n")

    def save_truck_battery_analysis(self, best_individual: Any, trucks: List[str], truck_groups: Dict[str, int], evaluator: Any):
        try:
            from genetic_algorithm import GroupedGeneticAlgorithm
            temp_ga = GroupedGeneticAlgorithm(evaluator, [], [], trucks, truck_groups, CONFIG)
            _, _, truck_batteries = temp_ga._decode(best_individual)
            battery_stats = {}
            total_battery_cost = 0

            for truck_id, battery_kwh in truck_batteries.items():
                if battery_kwh not in battery_stats:
                    battery_stats[battery_kwh] = 0
                battery_stats[battery_kwh] += 1
                total_battery_cost += battery_kwh * CONFIG["battery_cost_eur_per_kwh"]
            df = pd.DataFrame([
                {"truck_id": k, "battery_kwh": v, "battery_cost_eur": v * CONFIG["battery_cost_eur_per_kwh"]}
                for k, v in sorted(truck_batteries.items())
            ])

            battery_file = os.path.join(self.dir, f"battery_optimization_{self.tag}.csv")
            df.to_csv(battery_file, index=False)
            battery_summary = {
                "total_trucks": len(trucks),
                "total_battery_cost_eur": total_battery_cost,
                "average_battery_kwh": sum(truck_batteries.values()) / len(truck_batteries),
                "battery_distribution": {str(k): v for k, v in battery_stats.items()},
                "cost_savings_vs_uniform_300kwh": len(trucks) * 300 * CONFIG["battery_cost_eur_per_kwh"] - total_battery_cost
            }

            battery_stats_file = os.path.join(self.dir, f"battery_statistics_{self.tag}.json")
            with open(battery_stats_file, 'w', encoding='utf-8') as f:
                json.dump(battery_summary, f, indent=2, ensure_ascii=False)

            self._w(f"Battery analysis saved: {battery_file}")
            avg_battery = sum(truck_batteries.values()) / len(truck_batteries)
            savings = len(trucks) * 300 * CONFIG["battery_cost_eur_per_kwh"] - total_battery_cost
            self._w(f"Average battery: {avg_battery:.1f}kWh, Savings: {savings:,.0f}€")

        except Exception as e:
            self._w(f"Error saving battery analysis: {e}")

    def save_summary(self, info: Dict[str, Any]):
        p = os.path.join(self.dir, f"summary_enhanced_{self.tag}.json")
        with open(p, "w", encoding="utf-8") as f:
            json.dump(info, f, indent=2, ensure_ascii=False)
        self._w(f"Enhanced summary saved: {p}")

    def save_generation_history(self, info: Dict[str, Any]):
        try:
```

```python
            if 'generation_history' in info and info['generation_history']:
                generation_history = info['generation_history']
                history_df = pd.DataFrame(generation_history)
                history_csv = os.path.join(self.dir, f"generation_history_enhanced_{self.tag}.csv")
                history_df.to_csv(history_csv, index=False)

                history_json = os.path.join(self.dir, f"generation_history_enhanced_{self.tag}.json")
                with open(history_json, 'w', encoding='utf-8') as f:
                    json.dump(generation_history, f, indent=2, ensure_ascii=False)

                self._w(f"Evolution history saved: {history_csv}")

        except Exception as e:
            self._w(f"Error saving evolution history: {e}")

    def save_infrastructure_details(self, scs_optimal: set, ers_optimal: set, nodes: list, edges: list,
                                    scs_capex: float = 0, ers_capex: float = 0, ers_total_length_km: float = 0, evaluator=None):
        try:
            scs_details = []
            for i, node_id in enumerate(sorted(scs_optimal)):
                scs_details.append({
                    "scs_id": i + 1,
                    "node_id": node_id,
                    "power_kw": CONFIG["P_SCS_kw"],
                    "capex_eur": CONFIG["capex_scs_eur"],
                    "price_eur_per_kwh": CONFIG["price_stat_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_scs_eur_per_km"]
                })

            scs_df = pd.DataFrame(scs_details)
            scs_file = os.path.join(self.dir, f"scs_stations_{self.tag}.csv")
            scs_df.to_csv(scs_file, index=False)
            ers_details = []
            for i, edge_id in enumerate(sorted(ers_optimal)):
                edge_length = 1.0
                if evaluator and hasattr(evaluator, 'edge_length_km') and evaluator.edge_length_km:
                    edge_length = evaluator.edge_length_km.get(edge_id, 1.0)

                ers_details.append({
                    "ers_id": i + 1,
                    "edge_id": edge_id,
                    "length_km": edge_length,
                    "power_kw": CONFIG["P_ERS_kw"],
                    "capex_eur_per_km": CONFIG["capex_ers_eur_per_km"],
                    "total_capex_eur": edge_length * CONFIG["capex_ers_eur_per_km"],
                    "price_eur_per_kwh": CONFIG["price_dyn_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_ers_eur_per_km"]
                })

            ers_df = pd.DataFrame(ers_details)
            ers_file = os.path.join(self.dir, f"ers_segments_{self.tag}.csv")
            ers_df.to_csv(ers_file, index=False)
            infrastructure_summary = {
                "scs_stations": {
                    "count": len(scs_optimal),
                    "total_capex_eur": scs_capex,
                    "capex_per_station_eur": CONFIG["capex_scs_eur"],
                    "node_ids": sorted(list(scs_optimal)),
```

```
                    "power_kw_per_station": CONFIG["P_SCS_kw"],
                    "price_eur_per_kwh": CONFIG["price_stat_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_scs_eur_per_km"],
                    "coverage_ratio": f"{len(scs_optimal)}/{len(nodes)} ({len(scs_optimal)/len(nodes)*100:.1f}%)"
                },
                "ers_segments": {
                    "count": len(ers_optimal),
                    "total_length_km": ers_total_length_km,
                    "average_length_km": ers_total_length_km / len(ers_optimal) if len(ers_optimal) > 0 else 0,
                    "total_capex_eur": ers_capex,
                    "capex_eur_per_km": CONFIG["capex_ers_eur_per_km"],
                    "edge_ids": sorted(list(ers_optimal)),
                    "power_kw_per_segment": CONFIG["P_ERS_kw"],
                    "price_eur_per_kwh": CONFIG["price_dyn_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_ers_eur_per_km"],
                    "coverage_ratio": f"{len(ers_optimal)}/{len(edges)} ({len(ers_optimal)/len(edges)*100:.1f}%)"
                },
                "total_infrastructure": {
                    "scs_count": len(scs_optimal),
                    "ers_count": len(ers_optimal),
                    "total_capex_eur": scs_capex + ers_capex,
                    "scs_capex_eur": scs_capex,
                    "ers_capex_eur": ers_capex,
                    "infrastructure_capex_ratio": {
                        "scs_percentage": scs_capex / (scs_capex + ers_capex) * 100 if (scs_capex + ers_capex) > 0 else 0,
                        "ers_percentage": ers_capex / (scs_capex + ers_capex) * 100 if (scs_capex + ers_capex) > 0 else 0
                    },
                    "coverage_summary": {
                        "scs_nodes": f"{len(scs_optimal)}/{len(nodes)} ({len(scs_optimal)/len(nodes)*100:.1f}%)",
                        "ers_edges": f"{len(ers_optimal)}/{len(edges)} ({len(ers_optimal)/len(edges)*100:.1f}%)"
                    }
                }
            }

            infrastructure_file = os.path.join(self.dir, f"infrastructure_summary_{self.tag}.json")
            with open(infrastructure_file, 'w', encoding='utf-8') as f:
                json.dump(infrastructure_summary, f, indent=2, ensure_ascii=False)

            self._w(f"Infrastructure details saved: {scs_file}, {ers_file}")

        except Exception as e:
            self._w(f"Error saving infrastructure details: {e}")

    def save_infrastructure_distribution_json(self, scs_optimal: set, ers_optimal: set,
                                              nodes: list, edges: list, evaluator):
        try:
            def get_edge_length(edge_id):
                if hasattr(evaluator, 'edge_length_km') and evaluator.edge_length_km:
                    return evaluator.edge_length_km.get(edge_id, 1.0)
                return 1.0
            scs_distribution = {
                "summary": {
                    "total_stations": len(scs_optimal),
                    "total_nodes": len(nodes),
                    "coverage_percentage": len(scs_optimal) / len(nodes) * 100,
```

```
                    "total_capex_eur": len(scs_optimal) * CONFIG["capex_scs_eur"],
                    "power_per_station_kw": CONFIG["P_SCS_kw"],
                    "capex_per_station_eur": CONFIG["capex_scs_eur"],
                    "charging_price_eur_per_kwh": CONFIG["price_stat_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_scs_eur_per_km"]
                },
                "stations": []
            }
            for i, node_id in enumerate(sorted(scs_optimal)):
                scs_distribution["stations"].append({
                    "station_id": i + 1,
                    "node_id": node_id,
                    "power_kw": CONFIG["P_SCS_kw"],
                    "capex_eur": CONFIG["capex_scs_eur"],
                    "charging_price_eur_per_kwh": CONFIG["price_stat_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_scs_eur_per_km"]
                })
            ers_total_length = 0
            ers_total_capex = 0
            ers_segments_detail = []

            for i, edge_id in enumerate(sorted(ers_optimal)):
                edge_length = get_edge_length(edge_id)
                segment_capex = edge_length * CONFIG["capex_ers_eur_per_km"]
                ers_total_length += edge_length
                ers_total_capex += segment_capex

                ers_segments_detail.append({
                    "segment_id": i + 1,
                    "edge_id": edge_id,
                    "length_km": edge_length,
                    "power_kw": CONFIG["P_ERS_kw"],
                    "capex_eur_per_km": CONFIG["capex_ers_eur_per_km"],
                    "total_capex_eur": segment_capex,
                    "charging_price_eur_per_kwh": CONFIG["price_dyn_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_ers_eur_per_km"]
                })

            ers_distribution = {
                "summary": {
                    "total_segments": len(ers_optimal),
                    "total_edges": len(edges),
                    "coverage_percentage": len(ers_optimal) / len(edges) * 100,
                    "total_length_km": ers_total_length,
                    "average_segment_length_km": ers_total_length / len(ers_optimal) if len(ers_optimal) > 0 else 0,
                    "total_capex_eur": ers_total_capex,
                    "power_per_segment_kw": CONFIG["P_ERS_kw"],
                    "capex_per_km_eur": CONFIG["capex_ers_eur_per_km"],
                    "charging_price_eur_per_kwh": CONFIG["price_dyn_eur_per_kwh"],
                    "toll_eur_per_km": CONFIG["toll_ers_eur_per_km"]
                },
                "segments": ers_segments_detail
            }
            infrastructure_distribution = {
                "metadata": {
                    "generation_time": datetime.now().isoformat(),
                    "total_nodes": len(nodes),
                    "total_edges": len(edges),
                    "optimization_tag": self.tag
                },
                "scs_distribution": scs_distribution,
                "ers_distribution": ers_distribution,
```

```
                "cost_comparison": {
                    "scs_total_capex_eur": len(scs_optimal) * CONFIG["capex_scs_eur"],
                    "ers_total_capex_eur": ers_total_capex,
                    "infrastructure_total_capex_eur": len(scs_optimal) * CONFIG["capex_scs_eur"] + ers_total_capex,
                    "scs_percentage": (len(scs_optimal) * CONFIG["capex_scs_eur"]) / (len(scs_optimal) * CONFIG["capex_scs_eur"] + ers_total_capex) * 100 if (len(scs_optimal) * CONFIG["capex_scs_eur"] + ers_total_capex) > 0 else 0,
                    "ers_percentage": ers_total_capex / (len(scs_optimal) * CONFIG["capex_scs_eur"] + ers_total_capex) * 100 if (len(scs_optimal) * CONFIG["capex_scs_eur"] + ers_total_capex) > 0 else 0
                },
                "coverage_analysis": {
                    "scs_coverage": {
                        "selected_nodes": len(scs_optimal),
                        "total_nodes": len(nodes),
                        "coverage_ratio": len(scs_optimal) / len(nodes),
                        "coverage_percentage": len(scs_optimal) / len(nodes) * 100
                    },
                    "ers_coverage": {
                        "selected_edges": len(ers_optimal),
                        "total_edges": len(edges),
                        "coverage_ratio": len(ers_optimal) / len(edges),
                        "coverage_percentage": len(ers_optimal) / len(edges) * 100,
                        "total_length_km": ers_total_length
                    }
                }
            }
            distribution_file = os.path.join(self.dir, f"infrastructure_distribution_{self.tag}.json")
            with open(distribution_file, 'w', encoding='utf-8') as f:
                json.dump(infrastructure_distribution, f, indent=2, ensure_ascii=False)

            self._w(f"Infrastructure distribution saved: {distribution_file}")

        except Exception as e:
            self._w(f"Error saving infrastructure distribution JSON: {e}")


def extract_truck_ids_from_tours(tours: List) -> Tuple[List[str], Dict[str, int]]:
    from truck_grouping_strategy import group_trucks_by_distance_profile

    truck_ids = set()
    for tour in tours:
        truck_id = getattr(tour, "truck_id", "")
        if truck_id:
            truck_ids.add(truck_id)

    if not truck_ids:
        estimated_trucks = max(1, len(tours) // 10)
        truck_ids = {f"virtual_truck_{i:04d}" for i in range(estimated_trucks)}
        print(f"No truck_id detected, creating {len(truck_ids)} virtual trucks")

    truck_list = sorted(list(truck_ids))

    if len(truck_list) > 1000:
        print(f"Detected {len(truck_list)} trucks, enabling grouping strategy...")
        n_groups = min(20, max(5, len(truck_list) // 10000))
        truck_groups = group_trucks_by_distance_profile(tours, n_groups)
        print(f"Grouped {len(truck_list)} trucks into {n_groups} groups")
        return truck_list, truck_groups
    else:
```

```python
        truck_groups = {truck_id: i for i, truck_id in enumerate(truck_list)}
        return truck_list, truck_groups


def main():
    lg = EnhancedLogger()
    try:
        base = os.path.dirname(os.path.abspath(__file__))
        node_csv = os.path.join(base, "vam_node.csv")
        link_csv = os.path.join(base, "vam_link.csv")
        traj_csv = os.path.join(base, "tour_trajectory.csv")

        for f in [node_csv, link_csv, traj_csv]:
            if not os.path.exists(f):
                raise FileNotFoundError(f"Missing data file: {f}")

        lg._w("Data files validated")

        lg._w("\nLoading data...")
        t0 = time.time()
        tours, nodes, edges, edge_len = load_all_from_csv(
            node_csv, link_csv, traj_csv,
            default_speed_kmh=CONFIG["v_avg_kmh"]
        )
        lg._w(f"Data loaded: {len(tours)} tours, {len(nodes)} nodes, {len(edges)} edges")

        trucks, truck_groups = extract_truck_ids_from_tours(tours)
        n_groups = len(set(truck_groups.values()))
        lg._w(f"Trucks: {len(trucks)}, Groups: {n_groups}")

        evaluator = EnhancedBilevelEvaluator(CONFIG, edge_length_km=edge_len, truck_batt_map={})
        lg._w("Starting genetic algorithm optimization...")

        ga = GroupedGeneticAlgorithm(evaluator, nodes, edges, trucks, truck_groups, CONFIG)
        best_indiv, info = ga.evolve(
            tours,
            pop_size=CONFIG["ga_pop_size"],
            generations=CONFIG["ga_generations"],
            elite_size=CONFIG["ga_elite_size"],
            restart_every=CONFIG.get("ga_restart_every", 12),
        )

        scs_optimal, ers_optimal, truck_batteries_optimal = ga._decode(best_indiv)
        infeasible_count = info.get('infeasible_count', 0)
        omega_penalty = infeasible_count * CONFIG.get('omega', 1e6)
        real_cost = info['fitness'] - omega_penalty
        total_battery_cost = sum(battery_kwh * CONFIG["battery_cost_eur_per_kwh"]
                                for battery_kwh in truck_batteries_optimal.values())
        avg_battery_kwh = sum(truck_batteries_optimal.values()) / len(truck_batteries_optimal)
        uniform_300_cost = len(trucks) * 300 * CONFIG["battery_cost_eur_per_kwh"]
        battery_savings = uniform_300_cost - total_battery_cost
        scs_capex = len(scs_optimal) * CONFIG["capex_scs_eur"]
        ers_capex = 0
        ers_total_length_km = 0
        if hasattr(evaluator, 'edge_length_km') and evaluator.edge_length_km:
            for edge_id in ers_optimal:
                edge_length = evaluator.edge_length_km.get(edge_id, 1.0)
                ers_capex += edge_length * CONFIG["capex_ers_eur_per_km"]
                ers_total_length_km += edge_length
        else:
            ers_capex = len(ers_optimal) * CONFIG["capex_ers_eur_per_km"]
            ers_total_length_km = len(ers_optimal) * 1.0
```

```python
        infrastructure_capex = scs_capex + ers_capex

        lg._w("=" * 70)
        lg._w("Enhanced Optimization Results")
        lg._w("=" * 70)
        lg._w(f"Fitness: {info['fitness']:.0f}, Feasible Rate: {info['feasible_rate']:.1%}")
        lg._w(f"CAPEX: {info['capex']:,.0f}€, OPEX: {info['opex']:,.0f}€")
        lg._w(f"SCS: {len(scs_optimal)} stations, ERS: {len(ers_optimal)} segments")
        lg._w(f"Avg battery: {avg_battery_kwh:.1f}kWh, Savings: {battery_savings:,.0f}€")
        lg._w("=" * 70)

        lg._w("Saving results...")
        lg.save_summary(info)
        lg.save_truck_battery_analysis(best_indiv, trucks, truck_groups, evaluator)
        lg.save_generation_history(info)
        lg.save_infrastructure_details(scs_optimal, ers_optimal, nodes, edges,
                                       scs_capex, ers_capex, ers_total_length_km, evaluator)
        lg.save_infrastructure_distribution_json(scs_optimal, ers_optimal, nodes, edges,
        ↪   evaluator)
        lg._w("All result files saved")

    except Exception as e:
        lg._w(f"Runtime error: {e}")
        import traceback
        lg._w(f"Detailed error information: {traceback.format_exc()}")
        raise


if __name__ == "__main__":
    main()


```

A.2 Data Loader

```python
# -*- coding: utf-8 -*-
from typing import List, Tuple, Dict
import pandas as pd
from bilevel_milp_model import Tour, Segment


def _read_csv(path: str) -> pd.DataFrame:
    for enc in ("utf-8", "utf-8-sig", "latin-1"):
        try:
            return pd.read_csv(path, encoding=enc)
        except Exception:
            continue
    return pd.read_csv(path)

def _ensure(df: pd.DataFrame, cols: set, name: str):
    miss = cols - set(df.columns)
    if miss:
        raise ValueError(f"{name} missing required columns: {miss}")

def _canon_edge_id(u: str, v: str) -> str:
    a, b = str(u), str(v)
    return f"{a}|{b}" if a <= b else f"{b}|{a}"


def load_from_new_csvs(node_csv: str, link_csv: str, trajectory_csv: str,
                       default_speed_kmh: float = 60.0) -> Tuple[List[Tour], List[str],
                       ↪   List[str], Dict[str, float]]:
```

```python
    df_nodes = _read_csv(node_csv);  _ensure(df_nodes, {"node"}, node_csv)
    nodes = [str(x) for x in df_nodes["node"].tolist()]

    df_links = _read_csv(link_csv);  _ensure(df_links, {"from_node", "to_node", "distance_km"}, link_csv)
    tmp = (df_links.assign(u=lambda d: d["from_node"].astype(str),
                           v=lambda d: d["to_node"].astype(str),
                           length_km=lambda d: d["distance_km"].astype(float))
           [["u", "v", "length_km"]])
    tmp["edge_id"] = tmp.apply(lambda r: _canon_edge_id(r["u"], r["v"]), axis=1)
    edge_len = tmp.groupby("edge_id")["length_km"].mean().to_dict()
    edges = list(edge_len.keys())

    df_traj = _read_csv(trajectory_csv); _ensure(df_traj, {"carrier_id", "tour_id", "trip_id", "vam_path"}, trajectory_csv)
    truck_col = "truck_id" if "truck_id" in df_traj.columns else None

    tours: List[Tour] = []
    for (carrier, tour), g in df_traj.groupby(["carrier_id", "tour_id"], sort=False):
        g = g.sort_values("trip_id")
        truck = str(g.iloc[0][truck_col]) if truck_col else ""

        node_chain: List[str] = []
        for _, r in g.iterrows():
            raw = str(r["vam_path"])
            if not raw or raw.strip() == "":
                continue
            parts = [s.strip() for s in (raw.split("|") if "|" in raw else raw.split("->")) if s.strip()]
            if not parts:
                continue
            if not node_chain:
                node_chain.extend(parts)
            else:
                node_chain.extend(parts[1:] if node_chain[-1] == parts[0] else parts)

        segs: List[Segment] = []
        total = 0.0
        for i in range(len(node_chain) - 1):
            u, v = str(node_chain[i]), str(node_chain[i + 1])
            eid = _canon_edge_id(u, v)
            if eid not in edge_len:
                raise KeyError(f"[Path missing edge] {u}->{v} (canonicalized {eid}) not found in {link_csv}")
            dist = float(edge_len[eid]); total += dist
            segs.append(Segment(edge_id=eid, u=u, v=v, dist_km=dist, speed_kmh=float(default_speed_kmh)))

        if segs:
            tour_uid = f"{carrier}_{tour}"
            tours.append(Tour(tour_id=tour_uid, segments=segs,
                              origin_node=segs[0].u, dest_node=segs[-1].v,
                              total_dist_km=total, truck_id=truck))
    return tours, nodes, edges, edge_len


def load_from_old_csvs(centroids_csv: str, links_csv: str, paths_csv: str,
                       default_speed_kmh: float = 60.0) -> Tuple[List[Tour], List[str], List[str], Dict[str, float]]:
    df_nodes = _read_csv(centroids_csv); _ensure(df_nodes, {"LMSVAM"}, centroids_csv)
    nodes = [str(x) for x in df_nodes["LMSVAM"].tolist()]
    df_links = _read_csv(links_csv); _ensure(df_links, {"from_vam", "to_vam", "distance"}, links_csv)
```

```python
    tmp = (df_links.assign(u=lambda d: d["from_vam"].astype(str),
                           v=lambda d: d["to_vam"].astype(str),
                           length_km=lambda d: d["distance"].astype(float) / 1000.0)
           [["u", "v", "length_km"]])
    tmp["edge_id"] = tmp.apply(lambda r: _canon_edge_id(r["u"], r["v"]), axis=1)
    edge_len = tmp.groupby("edge_id")["length_km"].mean().to_dict()
    edges = list(edge_len.keys())

    dfp = _read_csv(paths_csv); _ensure(dfp, {"carrier_id", "tour_id", "trip_id", "path"}, paths_csv)
    tours: List[Tour] = []
    for (carrier, tour), g in dfp.groupby(["carrier_id", "tour_id"], sort=False):
        g = g.sort_values("trip_id")
        node_chain: List[str] = []
        for _, r in g.iterrows():
            raw = str(r["path"])
            parts = [s.strip() for s in raw.split("->") if s.strip()]
            if not parts:
                continue
            node_chain.extend(parts[1:] if node_chain and node_chain[-1] == parts[0] else parts)
        segs: List[Segment] = []
        total = 0.0
        for i in range(len(node_chain) - 1):
            u, v = str(node_chain[i]), str(node_chain[i + 1])
            eid = _canon_edge_id(u, v)
            if eid not in edge_len:
                raise KeyError(f"[Path missing edge] {u}->{v} (canonicalized {eid}) not found in {links_csv}")
            dist = float(edge_len[eid]); total += dist
            segs.append(Segment(eid, u, v, dist, float(default_speed_kmh)))
        if segs:
            tour_uid = f"{carrier}_{tour}"
            tours.append(Tour(tour_uid, segs, segs[0].u, segs[-1].v, total, ""))

    return tours, nodes, edges, edge_len


def load_all_from_csv(csv_a: str, csv_b: str, csv_c: str,
                      default_speed_kmh: float = 60.0) -> Tuple[List[Tour], List[str], List[str], Dict[str, float]]:
    dfs = {csv_a: _read_csv(csv_a), csv_b: _read_csv(csv_b), csv_c: _read_csv(csv_c)}
    node_like = next((p for p, d in dfs.items() if "node" in d.columns), None)
    link_like = next((p for p, d in dfs.items() if {"from_node", "to_node", "distance_km"} <= set(d.columns)), None)
    traj_like = next((p for p, d in dfs.items() if {"carrier_id", "tour_id", "trip_id", "vam_path"} <= set(d.columns)), None)
    if node_like and link_like and traj_like:
        return load_from_new_csvs(node_like, link_like, traj_like, default_speed_kmh)

    cent_like = next((p for p, d in dfs.items() if "LMSVAM" in d.columns), None)
    links_like = next((p for p, d in dfs.items() if {"from_vam", "to_vam", "distance"} <= set(d.columns)), None)
    paths_like = next((p for p, d in dfs.items() if {"carrier_id", "tour_id", "trip_id", "path"} <= set(d.columns)), None)
    if cent_like and links_like and paths_like:
        return load_from_old_csvs(cent_like, links_like, paths_like, default_speed_kmh)

    raise RuntimeError("Unable to auto-detect structure of the three CSV files, please check column names.")
```

A.3 Model

```python
# -*- coding: utf-8 -*-
from dataclasses import dataclass
from typing import Dict, List, Tuple, Set, Optional

@dataclass
class Segment:
    edge_id: str
    u: str
    v: str
    dist_km: float
    speed_kmh: Optional[float] = None


@dataclass
class Tour:
    tour_id: str
    segments: List[Segment]
    origin_node: str
    dest_node: str
    total_dist_km: float
    truck_id: str = ""


class EnhancedBilevelEvaluator:
    def __init__(self, config: Dict,
                 edge_length_km: Optional[Dict[str, float]] = None,
                 truck_batt_map: Optional[Dict[str, float]] = None):
        self.cfg = config
        self.edge_length_km = edge_length_km or {}
        self.truck_batt_map = truck_batt_map or {}

    def _diesel_penalty_cost(self, dist_km: float) -> float:
        p = self.cfg["penalty"]
        fuel_cost = p["diesel_l_per_km"] * dist_km * p["diesel_eur_per_l"]
        co2_cost = p.get("co2_g_per_km", 0.0) * dist_km * p["co2_eur_per_g"] if "co2_g_per_km" in p else 0.0
        outsource = p["outsourcing_eur_per_km"] * dist_km
        return fuel_cost + co2_cost + outsource

    def _travel_one_segment_smart(self, soc: float, Q: float, seg: Segment,
                                  ers_edges: Set[str], scs_nodes: Set[str],
                                  remaining_energy_after_this_seg: float = 0.0) -> Tuple[bool, float, float]:
        """Smart charging: only charge when battery is insufficient"""
        cfg = self.cfg
        kwh_per_km = cfg["beta_kwh_per_km"]
        v = seg.speed_kmh if (seg.speed_kmh and seg.speed_kmh > 0) else cfg["v_avg_kmh"]
        t_h = seg.dist_km / max(v, 1e-6)
        current_need = kwh_per_km * seg.dist_km

        cost = 0.0
        total_remaining_need = current_need + remaining_energy_after_this_seg
        if seg.edge_id in ers_edges and soc < total_remaining_need:
            energy_deficit = total_remaining_need - soc
            max_can_charge = cfg["P_ERS_kw"] * t_h
            available_capacity = Q - soc
            ers_energy = min(energy_deficit, max_can_charge, available_capacity)

            if ers_energy > 1e-9:
                soc += ers_energy
                cost += ers_energy * cfg["price_dyn_eur_per_kwh"]
```

```python
                cost += cfg["toll_ers_eur_per_km"] * seg.dist_km

        if soc < current_need:
            if seg.u in scs_nodes:
                add = min(Q - soc, current_need - soc)
                if add > 1e-9:
                    tchg = add / cfg["P_SCS_kw"]
                    soc += add
                    cost += add * cfg["price_stat_eur_per_kwh"]
                    cost += tchg * cfg["c_time_eur_per_h"]
                    cost += cfg["toll_scs_eur_per_km"] * seg.dist_km
            if soc < current_need - 1e-9:
                return False, soc, cost
        soc -= current_need
        if soc < 0:
            soc = 0.0

        return True, soc, cost

    def _travel_one_segment(self, soc: float, Q: float, seg: Segment,
                            ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, float, float]:
        """Compatible original interface, using smart charging strategy"""
        return self._travel_one_segment_smart(soc, Q, seg, ers_edges, scs_nodes, 0.0)

    def _evaluate_tour_with_batt_smart(self, tour: Tour, Q: float,
                                       ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, float]:
        """Smart tour evaluation with global energy demand consideration"""
        soc = min(Q, self.cfg["alpha_init_soc"] * Q)
        cost = 0.0
        beta = self.cfg["beta_kwh_per_km"]
        remaining_energy_need = sum(beta * seg.dist_km for seg in tour.segments)

        for seg in tour.segments:
            current_need = beta * seg.dist_km
            remaining_energy_need -= current_need

            ok, soc, inc = self._travel_one_segment_smart(
                soc, Q, seg, ers_edges, scs_nodes, remaining_energy_need
            )
            cost += inc
            if not ok:
                return False, 0.0

        return True, cost

    def _evaluate_tour_with_batt(self, tour: Tour, Q: float,
                                 ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, float]:
        """Compatible original interface, using smart evaluation"""
        return self._evaluate_tour_with_batt_smart(tour, Q, ers_edges, scs_nodes)

    def evaluate_tour(self, tour: Tour, ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, float]:
        if tour.truck_id and tour.truck_id in self.truck_batt_map:
            Q = float(self.truck_batt_map[tour.truck_id])
            ok, opex = self._evaluate_tour_with_batt_smart(tour, Q, ers_edges, scs_nodes)
            if ok:
                return True, opex
            else:
                return False, self._diesel_penalty_cost(tour.total_dist_km)

```

```
        for Q in self.cfg["battery_classes_kwh"]:
            ok, opex = self._evaluate_tour_with_batt_smart(tour, Q, ers_edges, scs_nodes)
            if ok:
                return True, opex
        return False, self._diesel_penalty_cost(tour.total_dist_km)

    def evaluate_tour_detailed(self, tour: Tour, ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, Dict[str, float]]:
        """Detailed evaluation of single tour, returns cost breakdown"""

        # Get battery capacity
        tid = getattr(tour, "truck_id", "")
        if tid and tid in self.truck_batt_map:
            Q = float(self.truck_batt_map[tid])
        else:
            # Fallback: enumerate battery classes to find minimum feasible
            for Q in sorted(self.cfg["battery_classes_kwh"]):
                ok, cost_detail = self._evaluate_tour_with_batt_detailed_smart(tour, Q, ers_edges, scs_nodes)
                if ok:
                    return ok, cost_detail
            # If all infeasible, return penalty breakdown
            return False, self._get_penalty_breakdown(tour)

        return self._evaluate_tour_with_batt_detailed_smart(tour, Q, ers_edges, scs_nodes)

    def _evaluate_tour_with_batt_detailed_smart(self, tour: Tour, Q: float, ers_edges: Set[str], scs_nodes: Set[str]) -> Tuple[bool, Dict[str, float]]:
        """Detailed evaluation of tour with specified battery capacity, using smart charging strategy"""

        # Initial SOC
        alpha = float(self.cfg["alpha_init_soc"])
        soc = alpha * Q

        # Detailed cost breakdown
        cost_detail = {
            "time_cost": 0.0,
            "static_charging_cost": 0.0,
            "dynamic_charging_cost": 0.0,
            "scs_toll_cost": 0.0,
            "ers_toll_cost": 0.0,
            "total_feasible_opex": 0.0
        }

        # Pre-calculate total energy demand for remaining tour
        beta = self.cfg["beta_kwh_per_km"]
        remaining_energy_need = sum(beta * seg.dist_km for seg in tour.segments)

        # Iterate through each segment
        for seg in tour.segments:
            current_need = beta * seg.dist_km
            remaining_energy_need -= current_need

            ok, soc, seg_cost_detail = self._travel_one_segment_detailed_smart(
                soc, Q, seg, ers_edges, scs_nodes, remaining_energy_need
            )
            if not ok:
                return False, self._get_penalty_breakdown(tour)

            # Accumulate various costs
            for key in cost_detail:
                if key in seg_cost_detail:
```

```
                cost_detail[key] += seg_cost_detail[key]

    # Calculate total feasible OPEX
    cost_detail["total_feasible_opex"] = sum(cost_detail[k] for k in cost_detail if k != "total_feasible_opex")

    return True, cost_detail

def _travel_one_segment_detailed_smart(self, soc: float, Q: float, seg: Segment,
                                       ers_edges: Set[str], scs_nodes: Set[str],
                                       remaining_energy_after_this_seg: float) -> Tuple[bool, float, Dict[str, float]]:
    """Detailed smart single segment travel, returns cost breakdown"""

    cfg = self.cfg
    beta = float(cfg["beta_kwh_per_km"])
    current_need = beta * seg.dist_km
    total_remaining_need = current_need + remaining_energy_after_this_seg

    v = seg.speed_kmh if (seg.speed_kmh and seg.speed_kmh > 0) else cfg["v_avg_kmh"]
    travel_time = seg.dist_km / max(v, 1e-6)

    cost_detail = {
        "time_cost": 0.0,
        "static_charging_cost": 0.0,
        "dynamic_charging_cost": 0.0,
        "scs_toll_cost": 0.0,
        "ers_toll_cost": 0.0
    }

    # 1) ERS smart charging: only use when insufficient battery
    if seg.edge_id in ers_edges and soc < total_remaining_need:
        energy_deficit = total_remaining_need - soc
        max_can_charge = cfg["P_ERS_kw"] * travel_time
        available_capacity = Q - soc

        ers_energy = min(energy_deficit, max_can_charge, available_capacity)

        if ers_energy > 1e-9:
            soc += ers_energy
            cost_detail["dynamic_charging_cost"] = ers_energy * cfg["price_dyn_eur_per_kwh"]
            cost_detail["ers_toll_cost"] = seg.dist_km * cfg["toll_ers_eur_per_km"]

    # 2) SCS charging: if still insufficient for current segment
    if soc < current_need:
        if seg.u in scs_nodes:
            energy_needed = current_need - soc
            available_capacity = Q - soc
            scs_energy = min(energy_needed, available_capacity)

            if scs_energy > 1e-9:
                charging_time = scs_energy / cfg["P_SCS_kw"]
                soc += scs_energy

                cost_detail["static_charging_cost"] = scs_energy * cfg["price_stat_eur_per_kwh"]
                cost_detail["scs_toll_cost"] = seg.dist_km * cfg["toll_scs_eur_per_km"]
                cost_detail["time_cost"] = charging_time * cfg["c_time_eur_per_h"]

        # Check if still insufficient
        if soc < current_need - 1e-9:
            return False, soc, cost_detail
```

```python

        # 3) Travel energy consumption
        soc -= current_need
        if soc < 0:
            soc = 0.0

        return True, soc, cost_detail

    def _get_penalty_breakdown(self, tour: Tour) -> Dict[str, float]:
        """Get penalty breakdown for infeasible tour"""
        distance = tour.total_dist_km

        penalty_detail = {
            "diesel_cost": distance * self.cfg["penalty"]["diesel_l_per_km"] * self.cfg["penalty"]["diesel_eur_per_l"],
            "co2_cost": 0.0,
            "outsourcing_cost": distance * self.cfg["penalty"]["outsourcing_eur_per_km"],
            "total_infeasible_penalty": 0.0
        }

        # CO2 cost
        if hasattr(tour, 'co2_gram'):
            penalty_detail["co2_cost"] = tour.co2_gram * self.cfg["penalty"]["co2_eur_per_g"]
        elif "co2_g_per_km" in self.cfg["penalty"]:
            co2_emissions = distance * self.cfg["penalty"]["co2_g_per_km"]
            penalty_detail["co2_cost"] = co2_emissions * self.cfg["penalty"]["co2_eur_per_g"]

        # Calculate total penalty
        penalty_detail["total_infeasible_penalty"] = sum(penalty_detail[k] for k in penalty_detail if k != "total_infeasible_penalty")

        return penalty_detail

    # --------------------- Solution level: Omega + CAPEX + OPEX(feasible) + C_pen(infeasible) ---------------------
    def evaluate_solution(self, tours: List[Tour], scs_nodes: Set[str], ers_edges: Set[str]) -> Dict:
        # Detailed cost breakdown
        opex_components = {
            "time_cost": 0.0,
            "static_charging_cost": 0.0,
            "dynamic_charging_cost": 0.0,
            "scs_toll_cost": 0.0,
            "ers_toll_cost": 0.0,
            "total_feasible_opex": 0.0
        }

        penalty_components = {
            "diesel_cost": 0.0,
            "co2_cost": 0.0,
            "outsourcing_cost": 0.0,
            "total_infeasible_penalty": 0.0
        }

        # Collect infeasible tours detailed information
        infeasible_tours_info = []
        feasible_cnt = 0

        # New: charging behavior classification statistics
        charging_behavior_stats = {
            "no_charging": 0,           # No charging
            "scs_only": 0,              # SCS only
            "ers_only": 0,              # ERS only
```

```
            "scs_and_ers": 0,           # Both SCS and ERS
            "penalty_tours": 0          # Infeasible tours
        }

        for t in tours:
            ok, cost_detail = self.evaluate_tour_detailed(t, ers_edges, scs_nodes)
            if ok:
                feasible_cnt += 1
                # Accumulate feasible costs
                for key in opex_components:
                    if key in cost_detail:
                        opex_components[key] += cost_detail[key]

                # Analyze charging behavior
                has_scs = cost_detail.get("static_charging_cost", 0) > 0
                has_ers = cost_detail.get("dynamic_charging_cost", 0) > 0

                if has_scs and has_ers:
                    charging_behavior_stats["scs_and_ers"] += 1
                elif has_scs:
                    charging_behavior_stats["scs_only"] += 1
                elif has_ers:
                    charging_behavior_stats["ers_only"] += 1
                else:
                    charging_behavior_stats["no_charging"] += 1
            else:
                # Count penalty tours
                charging_behavior_stats["penalty_tours"] += 1

                # Record infeasible tour information
                truck_id = getattr(t, "truck_id", "")
                battery_capacity = self.truck_batt_map.get(truck_id, 300.0) if truck_id in self.truck_batt_map else 300.0
                energy_need = self.cfg["beta_kwh_per_km"] * t.total_dist_km

                infeasible_info = {
                    "tour_id": t.tour_id,
                    "truck_id": truck_id,
                    "origin_node": t.origin_node,
                    "destination_node": t.dest_node,
                    "total_distance_km": t.total_dist_km,
                    "segments_count": len(getattr(t, 'segments', [])),
                    "battery_capacity_kwh": battery_capacity,
                    "energy_need_kwh": energy_need,
                    "energy_deficit_kwh": energy_need - battery_capacity,
                    "has_scs_access": t.origin_node in scs_nodes or t.dest_node in scs_nodes,
                    "has_ers_access": any(seg.edge_id in ers_edges for seg in getattr(t, 'segments', [])),
                    "diesel_penalty": cost_detail.get('diesel_cost', 0),
                    "co2_penalty": cost_detail.get('co2_cost', 0),
                    "outsourcing_penalty": cost_detail.get('outsourcing_cost', 0),
                    "total_penalty": cost_detail.get('total_infeasible_penalty', 0)
                }
                infeasible_tours_info.append(infeasible_info)

                # Accumulate penalties
                for key in penalty_components:
                    if key in cost_detail:
                        penalty_components[key] += cost_detail[key]

        infeasible_cnt = len(tours) - feasible_cnt
        feasible_rate = feasible_cnt / max(1, len(tours))
```

```
        # Calculate total OPEX and penalties
        opex_feasible = opex_components["total_feasible_opex"]
        pen_infeasible = penalty_components["total_infeasible_penalty"]

        # CAPEX: infrastructure + battery (per vehicle)
        capex_fac = len(scs_nodes) * self.cfg["capex_scs_eur"] +
        ↪  self._ers_capex_from_edges(ers_edges)
        capex_batt = self._battery_capex_from_trucks(tours)
        capex = capex_fac + capex_batt

        # Omega * (#unserved tours)
        omega = float(self.cfg.get("omega", 1e6))
        omega_term = omega * infeasible_cnt

        # Total objective
        fitness = omega_term + capex + opex_feasible + pen_infeasible

        # If still want to keep "feasibility rate threshold big penalty", optional:
        if self.cfg.get("min_feasible_rate", None) is not None and feasible_rate <
        ↪  self.cfg["min_feasible_rate"]:
            fitness += self.cfg.get("big_penalty_for_rate", 0.0)

        return {
            "fitness": fitness,
            "capex": capex,
            "capex_fac": capex_fac,
            "capex_batt": capex_batt,
            "opex": opex_feasible + pen_infeasible,  # Report total operation + penalty
            "opex_feasible": opex_feasible,
            "pen_infeasible": pen_infeasible,
            "feasible_rate": feasible_rate,
            "infeasible_count": infeasible_cnt,
            "omega_term": omega_term,

            # New: detailed cost breakdown
            "opex_breakdown": opex_components,
            "penalty_breakdown": penalty_components,
            "total_tours": len(tours),

            # New: charging behavior classification statistics
            "charging_behavior_stats": charging_behavior_stats,

            # New: infeasible tours detailed information
            "infeasible_tours_info": infeasible_tours_info,
        }

    def _ers_capex_from_edges(self, ers_edges: Set[str]) -> float:
        per_km = self.cfg["capex_ers_eur_per_km"]
        length_km = sum(float(self.edge_length_km.get(e, 1.0)) for e in ers_edges)
        return per_km * length_km

    def _battery_capex_from_trucks(self, tours: List[Tour]) -> float:
        # Only count battery CAPEX once for vehicles participating in tours
        if not self.truck_batt_map:
            return 0.0
        used_trucks = {t.truck_id for t in tours if t.truck_id and t.truck_id in
        ↪  self.truck_batt_map}
        per_kwh = self.cfg.get("battery_cost_eur_per_kwh", None)
        class_prices = self.cfg.get("batt_class_capex_eur", None)
        capex = 0.0
        for tid in used_trucks:
            Q = float(self.truck_batt_map[tid])
            if per_kwh is not None:
```

```python
                capex += Q * float(per_kwh)
            elif class_prices:
                if Q in class_prices:
                    capex += float(class_prices[Q])
                else:
                    bigger = sorted([q for q in class_prices if q >= Q])
                    capex += float(class_prices[bigger[0]]) if bigger else float(class_prices[max(class_prices)])
        return capex
# Compatibility alias, maintain backward compatibility
BilevelEvaluator = EnhancedBilevelEvaluator

```

A.4 Genetic Algorithm

```python
# -*- coding: utf-8 -*-
from __future__ import annotations
import random
from typing import List, Tuple, Dict, Any, Set, Optional


class GroupedGeneticAlgorithm:
    def __init__(self, evaluator, nodes: List[str], edges: List[str],
                 trucks: List[str], truck_groups: Dict[str, int], cfg: Dict[str, Any], seed: Optional[int] = None):
        self.evaluator = evaluator
        self.nodes, self.edges, self.trucks = nodes, edges, trucks
        self.truck_groups = truck_groups
        self.cfg = cfg
        self.n_nodes, self.n_edges = len(nodes), len(edges)

        self.n_groups = len(set(truck_groups.values()))
        self.group_to_trucks = {}
        for truck_id, group_id in truck_groups.items():
            if group_id not in self.group_to_trucks:
                self.group_to_trucks[group_id] = []
            self.group_to_trucks[group_id].append(truck_id)

        self.forbidden_nodes = {'544', '545'}
        self.forbidden_edges = self._identify_forbidden_edges()
        self.battery_classes = cfg["battery_classes_kwh"]
        self.n_battery_classes = len(self.battery_classes)
        self.battery_bits_per_group = 3
        self.chrom_len = self.n_nodes + self.n_edges + self.n_groups * self.battery_bits_per_group

        if seed is not None:
            random.seed(seed)


    def _identify_forbidden_edges(self) -> Set[str]:
        try:
            forbidden_edges = set()
            for edge_id in self.edges:
                if '|' in edge_id:
                    from_node, to_node = edge_id.split('|')
                    if from_node in self.forbidden_nodes or to_node in self.forbidden_nodes:
                        forbidden_edges.add(edge_id)

            return forbidden_edges

```

```
        except Exception:
            return set()

    def _decode(self, indiv: List[int]) -> Tuple[Set[str], Set[str], Dict[str, float]]:
        """Decode individual to SCS set, ERS set, truck battery mapping"""
        scs = {self.nodes[i] for i, b in enumerate(indiv[:self.n_nodes])
               if b == 1 and self.nodes[i] not in self.forbidden_nodes}
        ers_start = self.n_nodes
        ers_end = self.n_nodes + self.n_edges
        ers = {self.edges[i] for i, b in enumerate(indiv[ers_start:ers_end])
               if b == 1 and self.edges[i] not in self.forbidden_edges}

        truck_batteries = {}
        battery_start = self.n_nodes + self.n_edges

        for group_id in range(self.n_groups):
            start_bit = battery_start + group_id * self.battery_bits_per_group
            end_bit = start_bit + self.battery_bits_per_group
            battery_bits = indiv[start_bit:end_bit]

            battery_idx = 0
            for j, bit in enumerate(battery_bits):
                battery_idx += bit * (2 ** (self.battery_bits_per_group - 1 - j))

            battery_idx = min(battery_idx, self.n_battery_classes - 1)
            group_battery_kwh = float(self.battery_classes[battery_idx])

            if group_id in self.group_to_trucks:
                for truck_id in self.group_to_trucks[group_id]:
                    truck_batteries[truck_id] = group_battery_kwh

        return scs, ers, truck_batteries

    def _random_individual(self) -> List[int]:
        """Generate random individual with smart battery initialization"""
        p_scs = self.cfg.get("init_prob_scs", 0.4)
        p_ers = self.cfg.get("init_prob_ers", 0.1)
        scs_bits = []
        for i in range(self.n_nodes):
            if self.nodes[i] in self.forbidden_nodes:
                scs_bits.append(0)
            else:
                scs_bits.append(1 if random.random() < p_scs else 0)
        ers_bits = []
        for i in range(self.n_edges):
            if self.edges[i] in self.forbidden_edges:
                ers_bits.append(0)
            else:
                ers_bits.append(1 if random.random() < p_ers else 0)
        battery_bits = []
        for group_id in range(self.n_groups):
            if random.random() < 0.7:
                battery_idx = random.randint(0, 2)
            else:
                battery_idx = random.randint(0, self.n_battery_classes - 1)
            bits = []
            temp_idx = battery_idx
            for _ in range(self.battery_bits_per_group):
                bits.append(temp_idx % 2)
                temp_idx //= 2
            bits.reverse()
            battery_bits.extend(bits)

```

```python
        return scs_bits + ers_bits + battery_bits

    @staticmethod
    def _hamming(a: List[int], b: List[int]) -> int:
        return sum(x != y for x, y in zip(a, b))

    def ensure_diversity(self, population: List[List[int]]) -> List[List[int]]:
        min_ratio = self.cfg.get("min_hamming_ratio", 0.15)
        L = max(1, self.chrom_len)
        need = set()
        for i in range(len(population)):
            for j in range(i):
                if self._hamming(population[i], population[j]) / L < min_ratio:
                    need.add(i)
                    break
        for i in need:
            population[i] = self._random_individual()
        return population

    def fitness(self, indiv: List[int], tours) -> Tuple[float, Dict[str, Any]]:
        scs, ers, truck_batteries = self._decode(indiv)
        self.evaluator.truck_batt_map = truck_batteries

        res = self.evaluator.evaluate_solution(tours, scs, ers)
        return res["fitness"], res

    def select(self, population: List[List[int]], fits: List[float]) -> List[int]:
        k = self.cfg.get("ga_tournament_k", 3)
        idxs = random.sample(range(len(population)), min(k, len(population)))
        idxs.sort(key=lambda i: fits[i])
        return population[idxs[0]]

    def crossover(self, a: List[int], b: List[int]) -> Tuple[List[int], List[int]]:
        if self.chrom_len < 2 or random.random() > self.cfg.get("ga_crossover_rate", 0.8):
            return a[:], b[:]
        cut_points = [self.n_nodes, self.n_nodes + self.n_edges]
        cut = random.choice(cut_points + [random.randint(1, self.chrom_len - 1)])

        return a[:cut] + b[cut:], b[:cut] + a[cut:]

    def mutate(self, indiv: List[int]) -> List[int]:
        rate = self.cfg.get("ga_mutation_rate", 0.1)
        out = indiv[:]

        for i in range(self.chrom_len):
            if random.random() < rate:
                if self.n_nodes <= i < self.n_nodes + self.n_edges:
                    ers_idx = i - self.n_nodes
                    edge_id = self.edges[ers_idx]
                    if edge_id in self.forbidden_edges:
                        out[i] = 0
                    else:
                        out[i] = 1 - out[i]
                elif i < self.n_nodes:
                    node_id = self.nodes[i]
                    if node_id in self.forbidden_nodes:
                        out[i] = 0
                    else:
                        out[i] = 1 - out[i]
                elif i >= self.n_nodes + self.n_edges:
                    group_idx = (i - self.n_nodes - self.n_edges) //
                    ↪   self.battery_bits_per_group
                    bit_pos = (i - self.n_nodes - self.n_edges) % self.battery_bits_per_group
```

```
                    if random.random() < 0.7:
                        start_bit = self.n_nodes + self.n_edges + group_idx *
                        ↪  self.battery_bits_per_group
                        current_bits = out[start_bit:start_bit + self.battery_bits_per_group]
                        current_idx = 0
                        for j, bit in enumerate(current_bits):
                            current_idx += bit * (2 ** (self.battery_bits_per_group - 1 - j))
                        current_idx = min(current_idx, self.n_battery_classes - 1)

                        if current_idx > 0:
                            new_idx = random.randint(0, current_idx)
                        else:
                            new_idx = random.randint(0, min(2, self.n_battery_classes - 1))
                        new_bits = []
                        temp_idx = new_idx
                        for _ in range(self.battery_bits_per_group):
                            new_bits.append(temp_idx % 2)
                            temp_idx //= 2
                        new_bits.reverse()

                        for j, bit in enumerate(new_bits):
                            out[start_bit + j] = bit
                    else:
                        out[i] = 1 - out[i]
                else:
                    out[i] = 1 - out[i]

        return out

    def evolve(self, tours,
               pop_size: int = None, generations: int = None,
               elite_size: int = None, restart_every: int = 10,
               verbose: bool = True) -> Tuple[List[int], Dict[str, Any]]:

        pop_size = pop_size or self.cfg.get("ga_pop_size", 30)
        generations = generations or self.cfg.get("ga_generations", 25)
        elite_size = elite_size or self.cfg.get("ga_elite_size", 5)

        population = [self._random_individual() for _ in range(pop_size)]
        population = self.ensure_diversity(population)

        best = None
        best_info = None
        generation_history = []

        for g in range(generations):
            fits, infos = zip(*[self.fitness(ind, tours) for ind in population])
            order = sorted(range(pop_size), key=lambda i: fits[i])
            population = [population[i] for i in order]
            fits = [fits[i] for i in order]
            infos = [infos[i] for i in order]

            if best is None or fits[0] < best:
                best, best_info = fits[0], infos[0]

            scs, ers, truck_batteries = self._decode(population[0])
            group_battery_dist = {}
            for group_id in range(self.n_groups):
                if group_id in self.group_to_trucks and self.group_to_trucks[group_id]:
                    sample_truck = self.group_to_trucks[group_id][0]
                    battery_kwh = truck_batteries.get(sample_truck, 300.0)
                    group_battery_dist[f"Group{group_id}"] = battery_kwh
```

```
            generation_history.append({
                "generation": g,
                "best_fitness": fits[0],
                "feasible_rate": infos[0]['feasible_rate'],
                "capex": infos[0]['capex'],
                "opex": infos[0]['opex'],
                "capex_fac": infos[0].get('capex_fac', 0),
                "capex_batt": infos[0].get('capex_batt', 0),
                "infeasible_count": infos[0].get('infeasible_count', 0),
                "omega_term": infos[0].get('omega_term', 0),
                "scs_count": len(scs),
                "ers_count": len(ers),
                "group_battery_distribution": group_battery_dist
            })

            if verbose:
                avg_battery = sum(truck_batteries.values()) / len(truck_batteries) if truck_batteries else 300
                print(f"[Gen {g:02d}] fitness={fits[0]:.0f} FR={infos[0]['feasible_rate']:.1%} "
                      f"CAPEX={infos[0]['capex']:,.0f} Battery_avg={avg_battery:.0f}kWh "
                      f"SCS={len(scs)} ERS={len(ers)}")
            new_pop: List[List[int]] = population[:elite_size]

            while len(new_pop) < pop_size:
                p1 = self.select(population, fits)
                p2 = self.select(population, fits)
                c1, c2 = self.crossover(p1, p2)
                new_pop.append(self.mutate(c1))
                if len(new_pop) < pop_size:
                    new_pop.append(self.mutate(c2))

            population = self.ensure_diversity(new_pop[:pop_size])
            if restart_every and (g + 1) % restart_every == 0:
                keep = max(2, int(0.2 * pop_size))
                survivors = population[:keep]
                newcomers = [self._random_individual() for _ in range(pop_size - keep)]
                population = self.ensure_diversity(survivors + newcomers)

        fits, infos = zip(*[self.fitness(ind, tours) for ind in population])
        i = min(range(len(population)), key=lambda k: fits[k])

        final_info = dict(infos[i])
        final_info['generation_history'] = generation_history

        return population[i], final_info
```

A.5 truck grouping strategy

```
# -*- coding: utf-8 -*-
from typing import List, Dict
from collections import defaultdict
import numpy as np


def group_trucks_by_distance_profile(tours: List, n_groups: int) -> Dict[str, int]:
    """Group trucks based on distance characteristics"""
    truck_profiles = defaultdict(list)

    for tour in tours:
```

```python
        truck_id = getattr(tour, "truck_id", "")
        if truck_id:
            total_distance = 0
            if hasattr(tour, 'segments') and tour.segments:
                total_distance = sum(seg.dist_km for seg in tour.segments if hasattr(seg, 'dist_km'))
            truck_profiles[truck_id].append(total_distance)
    if not truck_profiles:
        estimated_trucks = max(1, len(tours) // 10)
        truck_groups = {}
        for i in range(estimated_trucks):
            truck_id = f"virtual_truck_{i:04d}"
            truck_groups[truck_id] = i % n_groups
        return truck_groups
    truck_features = {}
    truck_ids = list(truck_profiles.keys())

    for truck_id, distances in truck_profiles.items():
        if distances:
            avg_distance = np.mean(distances)
            total_distance = sum(distances)
            tour_count = len(distances)
            max_distance = max(distances)
            min_distance = min(distances)
        else:
            avg_distance = total_distance = tour_count = max_distance = min_distance = 0

        truck_features[truck_id] = [avg_distance, total_distance, tour_count, max_distance, min_distance]

    if len(truck_features) <= n_groups:
        return {truck_id: i for i, truck_id in enumerate(truck_ids)}
    truck_groups = {}

    sorted_trucks = sorted(truck_features.items(), key=lambda x: x[1][0])
    trucks_per_group = len(sorted_trucks) // n_groups
    remainder = len(sorted_trucks) % n_groups

    current_idx = 0
    for group_id in range(n_groups):
        group_size = trucks_per_group + (1 if group_id < remainder else 0)

        for i in range(group_size):
            if current_idx < len(sorted_trucks):
                truck_id = sorted_trucks[current_idx][0]
                truck_groups[truck_id] = group_id
                current_idx += 1

    return truck_groups


def group_trucks_simple(truck_ids: List[str], n_groups: int) -> Dict[str, int]:
    """Simple truck grouping: Round-robin assignment"""
    truck_groups = {}
    for i, truck_id in enumerate(truck_ids):
        truck_groups[truck_id] = i % n_groups

    return truck_groups
```

B Scientific Paper

Integrated Optimization of Charging Infrastructure for Battery-Electric Trucks Using Tour-Based Freight Data

Mingyan Jin[1] , **Lóri Tavasszy**[2] , **Alessandro Bombelli**[3]

Abstract

Heavy-duty road freight faces the dual challenge of limited driving range and insufficient charging infrastructure for battery-electric trucks (BETs). This study develops a nationwide tour-based optimization framework to determine cost-effective deployment strategies for stationary charging stations (SCS), electric road systems (ERS), and heterogeneous truck battery capacities.

A bi-level mixed-integer model is combined with a Genetic Algorithm (GA) to ensure scalability, evaluating more than 1.5 million truck tours in the Netherlands. The framework captures cumulative energy requirements across linked trips and is validated against exact MILP solutions for smaller instances.

The optimization increases feasibility from 58% to 89.9%, reducing infeasible tours to 10.1%. ERS emerges as the system backbone, with 12,792 km deployed (€25.7 billion, 65.6% of CAPEX), complemented by 251 low-cost SCS facilities (€50.2 million, <1% of CAPEX). Battery adoption is heterogeneous (90–600 kWh), yielding an average capacity of 357 kWh and reducing battery CAPEX by 19% compared to a uniform baseline. Operating expenditures are dominated by ERS charging, while unserved tours incur average penalties of €362.

Nationwide electrification of freight is thus feasible under a layered strategy: ERS as the long-haul backbone, SCS for redundancy at regional hubs, and heterogeneous batteries to optimize costs. The framework advances the literature by moving from trip-based to tour-based modelling at national scale, explicitly quantifying infeasibility and decomposing system costs. These findings provide actionable insights for policymakers and industry stakeholders on cost-effective and operationally viable pathways to freight decarbonization.

Keywords: freight electrification; battery-electric trucks; bi-level optimization; genetic algorithm; electric road systems (ERS); stationary charging stations (SCS); tour-based modelling; infrastructure planning

1 Introduction

The decarbonization of freight transport has become a pressing global priority, given that heavy-duty trucks are responsible for a substantial share of greenhouse gas emissions. Battery-electric trucks (BETs) represent one of the most promising solutions to this challenge, offering the potential to significantly reduce emissions from road freight. However, large-scale electrification of the trucking sector is hindered by two major obstacles. First, the high cost, weight, and energy density limitations of current batteries make them impractical for long-haul operations when used in large sizes. Second, the sparse and uneven deployment of charging infrastructure constrains the operational feasibility of BETs and increases the risk of unmet freight demand. Addressing these

1. MSc candidate, Faculty of Civil Engineering and Geosciences, Delft University of Technology, The Netherlands. Corresponding author: <m.jin-2@student.tudelft.nl>.

2. Professor, Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands. <L.A.Tavasszy@tudelft.nl>.

3. Dr., Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. <A.Bombelli@tudelft.nl>.

two barriers requires an integrated infrastructure planning approach that not only balances cost and performance but also ensures service continuity for logistics operations.

Existing studies have primarily focused on either static charging stations (SCS), which provide fixed-location charging opportunities, or dynamic charging systems such as Electric Road Systems (ERS), which enable energy supply during driving. While each option has merits, planning them in isolation neglects the potential synergies between the two. Moreover, most studies have assumed uniform battery capacities across the fleet, an unrealistic simplification that ignores the heterogeneity of operational needs and leads to either excessive investment or limited feasibility. Another limitation is that infrastructure planning models have largely relied on trip-based demand representations, where each trip is considered independently. This approach overlooks the fact that freight carriers operate in tours, i.e., sequences of trips that must be feasible as a whole in terms of energy requirements. Ignoring tour continuity can therefore lead to misrepresentation of charging needs and an underestimation of feasibility issues.

To fill these gaps, this study proposes a nationwide integrated optimization framework that jointly determines the deployment of SCS and ERS, along with heterogeneous battery allocations across the truck fleet. The model is built on tour-based freight data from the Netherlands, enabling a realistic representation of logistics operations. A bi-level optimization framework is designed: the upper level selects infrastructure deployment and battery sizes, while the lower level evaluates the energy feasibility of complete tours, accounting for charging opportunities, state-of-charge dynamics, and penalties for unserved tours. Given the large scale of the problem—encompassing over one million tours and millions of trips—a meta-heuristic genetic algorithm is developed to efficiently solve the nationwide instance, with smaller test cases validated using a mixed-integer linear programming (MILP) formulation.

The results demonstrate that the integrated approach substantially improves the feasibility of BET operations, raising the share of feasible tours from 58% to nearly 90%. ERS emerges as the dominant infrastructure in terms of cost-effectiveness, accounting for the majority of capital expenditures, while SCS provides redundancy at a relatively small share of the budget. At the fleet level, heterogeneous battery sizing proves crucial, with a wide distribution of battery capacities across trucks resulting in 19% capital cost savings compared to uniform allocation. These findings highlight the importance of considering infrastructure synergies, realistic freight operations, and heterogeneous vehicle strategies when designing electrification pathways. Beyond the methodological contributions, the study offers practical insights for policymakers and industry stakeholders by quantifying trade-offs between capital investments, operational costs, and serviceability. The framework thus provides a basis for informed decision-making on the future of freight electrification at the national scale.

2 Literature Review

Electrifying heavy-duty road freight has advanced from a technological possibility to a system-level planning challenge in which battery technology, charging infrastructure, and freight operations must be co-designed. The literature converges on two complementary infrastructure paradigms. Static charging stations (SCS) provide high-power refuelling at nodes and are comparatively easy to deploy within existing grids, but they induce dwell time and typically presume large on-board batteries to bridge between stations. In contrast, electric road systems (ERS) deliver energy while driving and therefore relax battery sizing and time-loss constraints, albeit at the cost of substantial corridor investments and the need for targeted coverage. Technology assessments consistently classify ERS into overhead catenary, ground-level conductive rails, and wireless inductive transfer, documenting heterogeneous maturity, efficiencies, and implementation footprints (Piedel et al. 2024; Shoman, Karlsson, and Yeh 2022; Honda 2021). Comparative and policy-oriented analyses find that, on high-volume corridors, ERS can cut required battery capacity and total operating costs, but sustained utilisation hinges on corridor selection, traffic density, and policy instruments; mis-targeted rollout risks stranded assets (Rogstadius et al. 2023, 2025; Decisio 2022).

These insights motivate hybrid strategies that pair ERS on interurban spines with SCS at hubs and logistics clusters to provide redundancy, last-mile coverage, and recovery options—an idea increasingly explored in integrated models.

Methodologically, integrated planning has moved from single-technology facility location problems to frameworks that link investment to operating feasibility and costs. Bi-level formulations are a natural scaffold: the upper level chooses siting and capacity (e.g., ERS segments and SCS locations), while the lower level evaluates vehicle operations, charging decisions, state-of-charge (SOC) dynamics, and generalised costs. This structure has been adapted to combine static and dynamic charging and to capture behavioural or operational responses; scalability to national instances often relies on metaheuristics such as genetic algorithms (GA) (Sun, Chen, and Yin 2020; Zeng and Zhang 2020; Akbari, Brenna, and Longo 2018; Cintrano, Toutouh, and Alba 2021; Vazifeh et al. 2019; Seilabi and coauthors 2025). Evidence from corridor and national case studies shows that integrated designs outperform siloed rollouts because the model can trade corridor energy against node coverage and explicitly price dwell time, tolls, and electricity tariffs in the operating layer (Sun, Chen, and Yin 2020; Decisio 2022). Yet, many existing studies retain strong simplifications in fleet specification and demand representation, limiting realism when scaled up.

A second cross-cutting theme is battery sizing. For tractability, early infrastructure studies commonly imposed a uniform battery capacity. That assumption obscures the diversity of duty cycles, topography, and temporal charging opportunities that shape energy feasibility in practice. More recent contributions permit heterogeneous battery classes and co-optimise them with infrastructure, showing that modest batteries can serve a large share of tours when supported by well-placed ERS or frequent SCS, while a tail of long-haul missions justifies larger packs; the aggregate effect is lower capital expenditure without sacrificing feasibility (Liao et al. 2024; Kunawong and coauthors 2025; Saxe et al. 2023; Inez 2024). This reframes infrastructure versus battery investment as partial substitutes (on corridors with ERS) and complements (around hubs facing peak demand), underscoring the value of joint design.

Equally pivotal is how freight demand is represented. Most siting models are trip-based: each movement is treated independently, which eases computation but severs the energy continuity that operators confront during daily operations. Research in freight tour synthesis and urban goods modelling shows that tour-based demand—sequences of chained trips executed by the same vehicle, including empty returns and inter-stop dwell—captures SOC carry-over, headroom constraints, and realistic charging windows. Ignoring tour continuity systematically underestimates charging needs, overstates feasibility, and can misallocate infrastructure, especially where energy margins are tight (Boerkamps and Binsbergen 1999; Sánchez-Díaz, Holguín-Veras, and Ban 2015; Thoen et al. 2020). Incorporating tour-level physics into national planning remains rare but is essential for credible evaluation of infeasibility and penalty mechanisms (e.g., outsourcing or diesel fallback), and for diagnosing where corridor energy should be complemented by nodal redundancy.

Taken together, the literature points to three gaps that motivate the present study: (i) nationwide hybrid SCS–ERS planning rather than single-technology optimisation, so that corridor continuity and nodal accessibility are jointly targeted; (ii) explicit battery heterogeneity within infrastructure co-optimisation to capture investment–feasibility trade-offs across diverse duties; and (iii) scalable, tour-aware feasibility evaluation that preserves SOC continuity and charging opportunities at operational resolution. Addressing these gaps requires models that integrate strategic siting with lower-level energy balance, price-consistent operating costs, and penalty structures for unmet missions, solved with algorithms capable of handling national network sizes—an approach increasingly advocated across recent technical and policy-facing contributions (Sun, Chen, and Yin 2020; Zeng and Zhang 2020; Decisio 2022; Rogstadius et al. 2025).

3 Methodology

This paper develops a nationwide, tour-based planning framework that jointly decides (i) which road links to electrify with in-motion charging (*electric road systems*, ERS), (ii) which nodes to equip with static charging stations (SCS), and (iii) heterogeneous battery capacities at the vehicle level. The model links long-run investment to day-to-day operating feasibility at the level of complete *tours*—ordered chains of trips executed by the same truck—so that state-of-charge (SOC) carry-over, empty returns, and realistic charging windows are preserved. The tour representation is motivated by freight-tour synthesis and operations research showing that trip-wise abstractions systematically understate energy needs and overstate feasibility (Boerkamps and Binsbergen 1999; Sánchez-Díaz, Holguín-Veras, and Ban 2015; Thoen et al. 2020). Strategic–operational coupling is handled through a bi-level structure: an upper level that chooses infrastructure and battery assignments and a lower level that verifies tour feasibility and returns operating costs (Sun, Chen, and Yin 2020; Zeng and Zhang 2020). To meet page limits, we present modeling principles and a few illustrative equations in the main text; the complete mathematical program (sets, variables, and constraints) is documented in the Appendix.

3.1 Network, demand, and decisions

Let $G = (V, E)$ denote the corridor graph used for planning. Each observed tour $t \in \mathcal{T}$ is an ordered sequence of legs on E obtained by assigning component trips to shortest paths; this yields consistent mileage for both operations and infrastructure accounting. Strategic decisions comprise binary ERS activation on links, $x_e \in \{0,1\}$, binary SCS siting at nodes, $y_i \in \{0,1\}$, and one-of-$|\mathcal{B}|$ battery-class assignment for each vehicle k, encoded by $\delta_{k,b} \in \{0,1\}$. Battery classes $b \in \mathcal{B}$ have capacities $\bar{Q}_b$ and specific costs c_b^{bat}. Electricity consumed on ERS and SCS is priced at p^{ers} and p^{scs}. Tours that cannot be feasibly electrified may be outsourced (or executed with diesel) at cost c_t^{out}. A large scalar Ω is used as an *internal* penalty in the solver to discourage excessive outsourcing; Ω is not included in the reported economic totals.

3.2 Upper-level principle (investment and system cost)

The upper level minimizes long-run system cost—CAPEX for ERS/SCS/batteries plus OPEX for electricity and any outsourcing:

$$\min \underbrace{\sum_{e\in E} c_e^{\text{ers}}\,\ell_e\,x_e + \sum_{i\in V} c_i^{\text{scs}}\,y_i + \sum_k \sum_b c_b^{\text{bat}}\,\delta_{k,b}}_{\text{CAPEX}} + \underbrace{\sum_{t\in\mathcal{T}} \big(C_t^{\text{op}}(x,y,\delta) + C_t^{\text{out}}\big)}_{\text{OPEX + outsourcing}}, \tag{1}$$

subject to the one-battery-per-vehicle rule

$$\sum_{b\in\mathcal{B}} \delta_{k,b} = 1 \quad \forall k, \qquad \delta_{k,b} \in \{0,1\}, \tag{2}$$

and optional budget/policy constraints (e.g., caps on ERS length or SCS count). The lower-level terms C_t^{op} and C_t^{out} are defined by tour-feasibility evaluation described next.

3.3 Lower-level principle (tour feasibility and operating cost)

Given (x, y, δ), each tour is simulated leg by leg. Let $k = \kappa(t)$ denote the assigned vehicle and b the chosen battery class with capacity $\bar{Q}_b$. SOC evolves as

$$\text{SOC}_{t,0} = \alpha\,\bar{Q}_b, \qquad \text{SOC}_{t,\ell} = \text{SOC}_{t,\ell-1} - \underbrace{\eta_{e(\ell)}\,d_{e(\ell)}}_{\text{traction use}} + \underbrace{x_{e(\ell)}\,\gamma_{e(\ell)}\,d_{e(\ell)}}_{\text{ERS in-motion charge}}, \quad 0 \le \text{SOC}_{t,\ell} \le \bar{Q}_b, \tag{3}$$

where $e(\ell)$ is the link used by leg ℓ, d_e its length, η_e the energy intensity (kWh/km), and γ_e the effective ERS charge per km (power-over-speed). At intermediate stops that coincide with SCS nodes ($y_i = 1$), a *conditional top-up* is permitted to secure forthcoming legs without excessive dwell; timing and power constraints are given in the Appendix. A binary variable $s_t \in \{0,1\}$ indicates whether tour t is served electrically ($s_t = 1$) or outsourced ($s_t = 0$):

$$s_t = 1 \;\Rightarrow\; \text{SOC bounds respected under the SCS policy;} \qquad C_t^{\text{out}} = (1 - s_t)\, c_t^{\text{out}}. \tag{4}$$

Operating cost aggregates priced energy purchased on ERS and SCS along feasible tours,

$$C_t^{\text{op}} \;=\; s_t\Big(\sum_{\ell \in t} p^{\text{ers}}\, x_{e(\ell)}\, \gamma_{e(\ell)}\, d_{e(\ell)} \;+\; \sum_{\text{SCS events}} p^{\text{scs}}\, u_{t,\ell}\Big), \tag{5}$$

where $u_{t,\ell}$ is the SCS energy top-up taken at eligible stops. For search guidance, the solver's fitness augments (1) with an infeasibility penalty $\Omega \sum_t (1 - s_t)$.

3.4 Why tours and heterogeneous batteries

Preserving tour continuity allows the model to evaluate feasibility where energy margins are tight and charging windows are sparse, addressing well-known shortcomings of trip-based siting models (Boerkamps and Binsbergen 1999; Sánchez-Díaz, Holguín-Veras, and Ban 2015; Thoen et al. 2020). Assigning exactly one battery class per vehicle (Eq. (2)) reflects fleet practice and captures the substitution–complementarity ERS and battery investment: modest packs become viable on well-electrified corridors while a minority of long-haul duties justifies larger capacities (Liao et al. 2024; Saxe et al. 2023; Inez 2024).

3.5 Solution approach (bi-level with metaheuristics)

The decision vector concatenates three blocks—ERS links $\{x_e\}$, SCS nodes $\{y_i\}$, and battery assignments $\{\delta_{k,b}\}$. We use a genetic algorithm (GA) for the upper level: chromosomes are block-encoded; selection is elitist; two-point crossover is applied within blocks to preserve structure; mutation uses sparsity-aware flips for $\{x_e, y_i\}$ and local swaps for $\{\delta_{k,b}\}$; and a lightweight repair operator enforces (2). Each candidate layout is decoded and evaluated by the lower-level tour simulation to compute feasibility and costs in (1). GA is a widely adopted choice for large-scale facility siting and EV infrastructure planning when exact solvers become intractable (Akbari, Brenna, and Longo 2018; Vazifeh et al. 2019; Cintrano, Toutouh, and Alba 2021; Seilabi and coauthors 2025); the bi-level framing follows integrated static–dynamic charging studies (Sun, Chen, and Yin 2020; Zeng and Zhang 2020). Practical run settings (population, generations, operators, termination, and parallel evaluation strategy) are summarized in the Appendix together with sensitivity checks.

3.6 Scope and appendix contents

Equations (1)–(5) are intentionally minimal and illustrative. The Appendix provides the complete model: sets and indices; parameter definitions; all decision variables; the full upper-level objective with policy/budget constraints; the detailed SOC and SCS timing/power constraints; optional features (e.g., speed–power coupling on ERS, charger queueing abstractions); cost accounting; and solver pseudocode with parameter values used in experiments. This separation keeps the main text focused on design principles while maintaining full reproducibility off the critical path.

4 Case Study

This section documents the study area, the data pipeline that transforms micro-level freight records into a corridor-consistent tour dataset, the construction of the nationwide corridor network used for planning, and the implementation details of the heuristic solver. Figures reproduced below visualize the data abstractions we rely on for the national case. The parameter table for the Genetic Algorithm (GA) and key technology/cost inputs is reported in the Appendix to keep the main text concise.

4.1 Study Area and Data Overview

We use micro-level freight records that contain hierarchical identifiers (carrier, tour, trip), zonal references (NRM, BG, VAM), RD coordinates, timestamps, vehicle/commodity descriptors, distances, and CO_2 measures. From these fields we retain only what is required for planning: identifiers to reconstruct ordered tours, zonal/coordinate fields to embed tours in the corridor network, temporal fields to detect charging opportunities, and distance/emission fields for energy and penalty accounting. A compact list of variables and their roles is summarised in the data table (see Appendix). This subset supplies the core inputs for the tour-based feasibility checks and the investment–operation trade-off in our model.

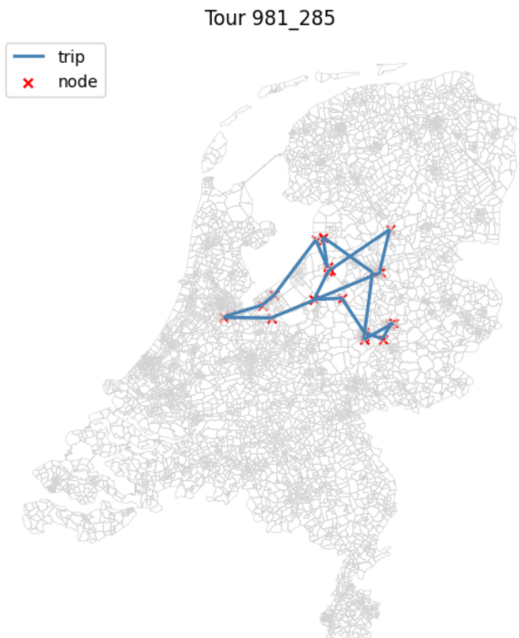


Figure 1: Example tour illustration

Figure 1 (tour demo) shows how a single tour is visualised as an ordered chain of OD legs. The raw data provide endpoints rather than path geometry; we therefore reconstruct routes on the VAM corridor network to obtain a graph-consistent distance metric used consistently in both operations and infrastructure accounting.

4.2 Zonal Framework and Flow Aggregation

For national-scale planning we operate at the level of the 542 Dutch VAM zones. Compared with node-level siting, zoning exposes shared corridor structure, keeps the model size tractable, and aligns with freight-demand practice. Figure 2 displays the spatial distribution of VAM zones used as SCS candidates; Figure 3 contrasts the raw node-level trip mesh with zone-level corridor flows used for planning. In the zonal abstraction, any SCS inside a zone is accessible to all tours traversing that zone, while ERS is deployed on inter-zonal links.

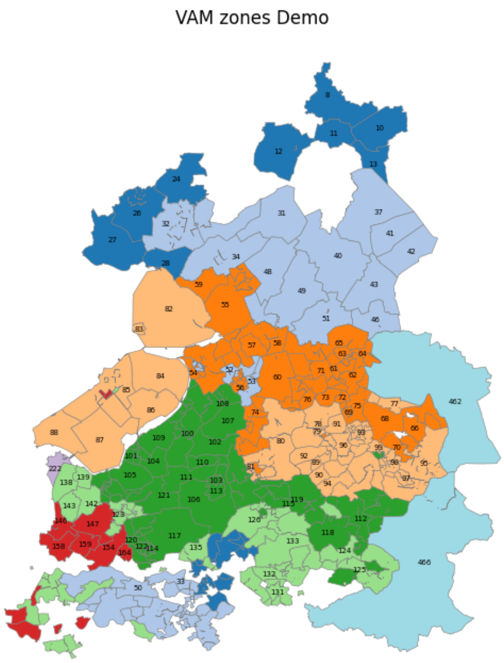


Figure 2: VAM zoning across the study area

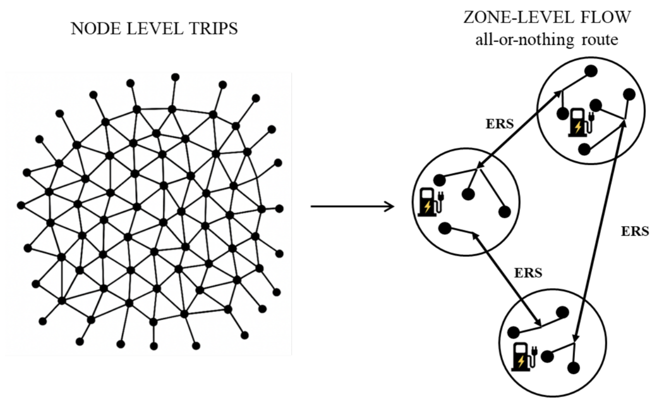


Figure 3: From node-level trips (left) to zone-level corridor flows (right)

4.3 Corridor Network Construction and Tour Routing

We construct a sparse, well-connected inter-zonal corridor graph $G = (V, E)$ by a Delaunay triangulation over VAM centroids (RD New, EPSG:28992). The resulting network has $|V| = 542$ nodes and $|E| = 1613$ links; link lengths are Euclidean centroid–centroid distances and serve as the common metric for energy and costs. Compared with k-NN or radius graphs, Delaunay naturally encodes proximate neighbours, avoids crisscrossing, and preserves single-component connectivity with a compact number of links—well suited for corridor-level ERS on links and SCS on nodes. Figure 4 shows the network.

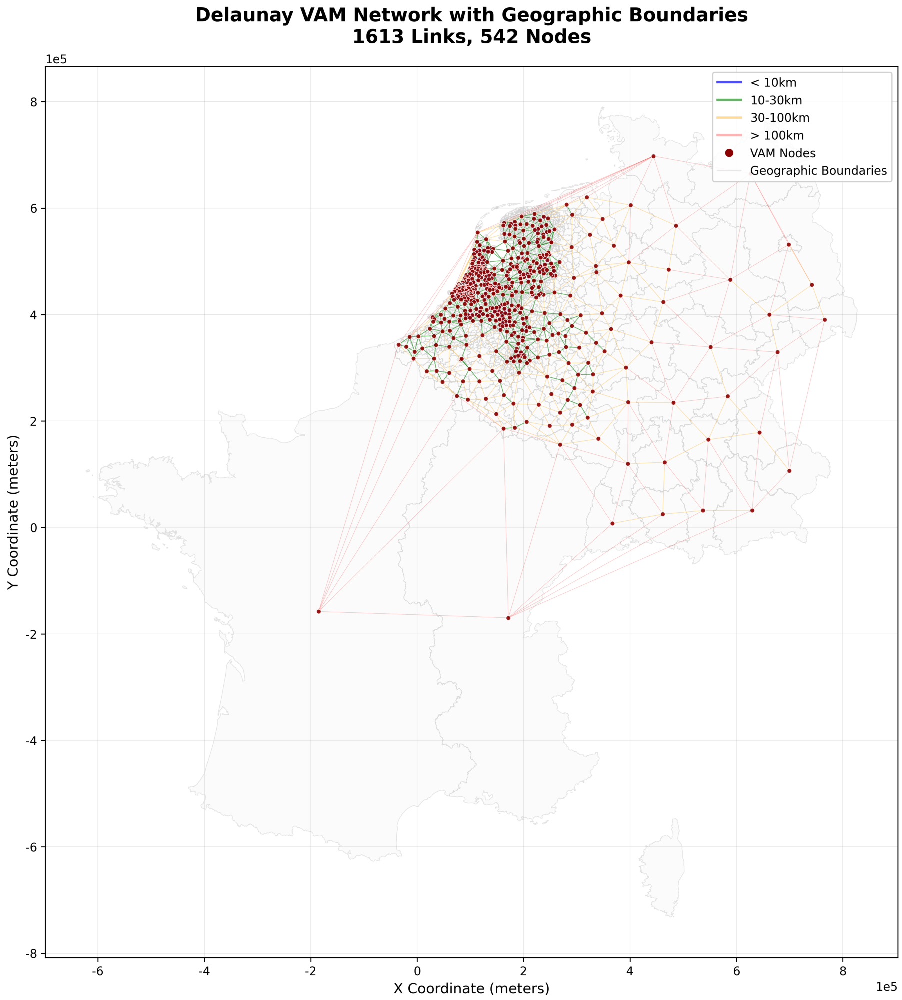


Figure 4: Delaunay-based VAM corridor network ($|V| = 542$, $|E| = 1613$)

Because raw records contain only OD endpoints, every trip is re-routed by Dijkstra's shortest path on G, and multi-leg paths are concatenated to form clean zone-by-zone tours; duplicates are removed. Figure 5 shows one reconstructed tour path. The model consistently uses reconstructed corridor length (sum of link lengths along the path) for energy/toll calculations, keeping operations and infrastructure on a common graph metric.

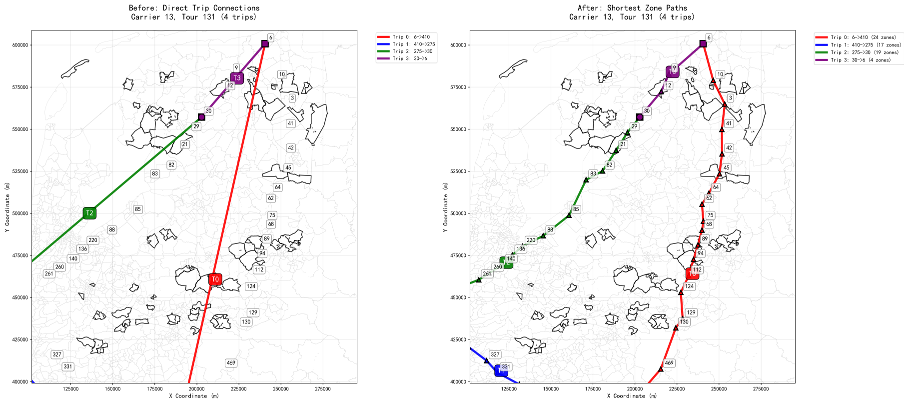


Figure 5: Example of a reconstructed tour on the VAM corridor network

Pipeline artefacts. The data pipeline outputs: (i) a tour table with ordered segments and corridor links; (ii) a link table with canonical edge IDs and lengths for ERS siting and energy/cost evaluation; (iii) a node table of VAM zones for SCS; and (iv) a truck catalogue mapping vehicles to tours for per-vehicle battery assignment. These artefacts provide a clean handoff to the optimisation layer.

4.4 Heuristic Solver and Workflow

The optimisation problem—jointly siting SCS, deploying ERS, and assigning heterogeneous battery classes per vehicle under tour-feasibility checks—is NP-hard at national size. We therefore adopt a tailored Genetic Algorithm (GA) with a three-segment chromosome (SCS / ERS / batteries), tournament selection, block-aligned two-point crossover, sparsity-aware mutation (battery-biased downward), elitism, diversity control by Hamming distance, and periodic restarts. Each chromosome is decoded to decisions and evaluated by a deterministic tour-energy simulation aligned with the corridor graph; the layered fitness prioritises feasibility (Ω penalty on unserved tours) before infrastructure/battery CAPEX and then OPEX. Figure 6 summarises the workflow. (GA hyperparameters and technology/cost scalars are reported in the Appendix.)

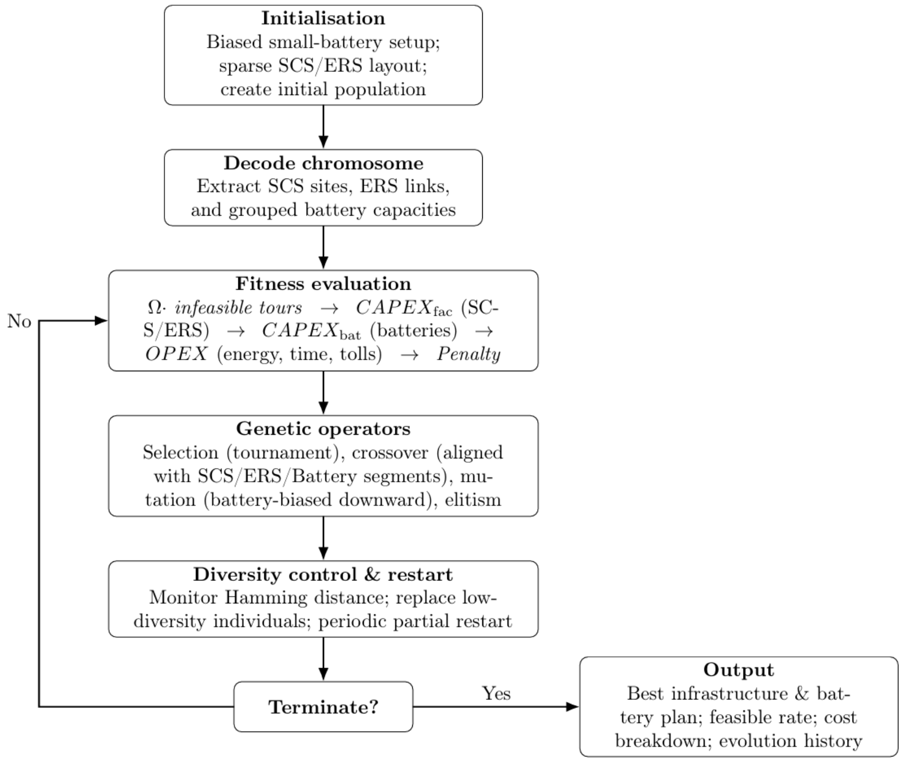


Figure 6: Workflow of the GA for the bi-level optimisation

Rationale and validation (concise). GA is suitable for mixed discrete decisions, nonlinear feasibility checks, and multi-component costs, with embarrassingly parallel fitness evaluation. Prior EV-infrastructure studies report strong performance of GA at city scale; our validation confirms correctness on small instances by matching exact MILP optima and shows scalability on larger sets where MILP is intractable. (Convergence and MILP-benchmark figures are placed in the Results section; detailed settings and unified test parameters appear in the Appendix.)

4.5 What is reported where

To preserve readability and the page budget, this chapter presents (i) the data abstractions and corridor construction with figures, and (ii) the solver workflow at a high level. All numerical parameter values (GA hyperparameters, Ω, unit costs for ERS/SCS/batteries, energy prices, and core energy-intensity assumptions) and the full GA parameter table are moved to the Appendix (see Table 1).

5 Results

5.1 Convergence and overall performance

The nationwide optimisation converges rapidly under the tailored GA: the best fitness improves sharply during the first dozen generations and then plateaus, while the electric service rate rises from the initial baseline to a high, stable level (Fig. 7). Capital expenditure (CAPEX) increases as the algorithm invests in corridor energy and nodal redundancy, whereas operating expenditure (OPEX) declines as more tours are served electrically. The average assigned battery capacity stabilises rather than drifting upward, indicating that the search settles on a heterogeneous portfolio instead of “max-battery everywhere” (Fig. 7).

5.2 National deployment pattern

The optimised layout exhibits a layered design that combines in-motion charging on high-volume inter-urban corridors with static fast charging at strategically located hubs. ERS segments form a long-haul backbone along major spines, while SCS sites cluster around distribution centres and gateway nodes with dense tour interactions (Figs. 8a–8b). This pairing provides corridor continuity and last-mile recovery without excessive dwell.

5.3 Residual infeasibility: locations and drivers

Despite large-scale deployment, a small share of tours remains infeasible. These residuals are geographically concentrated: a limited set of nodes accounts for a disproportionate number of infeasible tours (Fig. 9a), and a corresponding map view reveals clusters near peripheral and cross-border areas where long legs and sparse siting options restrict energy headroom (Fig. 9b). These locations are high-leverage targets for incremental corridor extensions or backup SCS.

5.4 Fleet battery portfolio and charging behaviour

The resulting fleet mix is bimodal, with sizable shares at a small-pack class and a large-pack class, and the remainder spread across intermediate capacities (Fig. 10a). At the tour level, most missions complete without en-route charging; when charging occurs, ERS is the dominant modality, with SCS used sparingly as conditional top-ups to secure the next energy-critical legs (Fig. 10b). This confirms the substitution–complementarity between corridor energy and on-board storage.

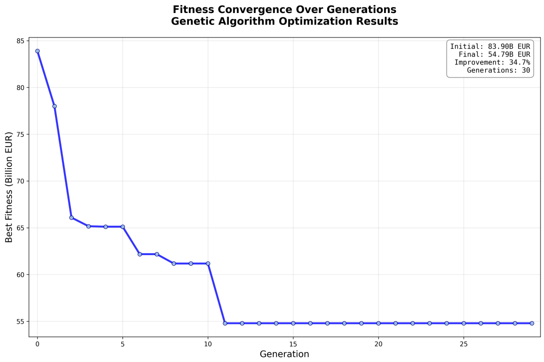
5.5 Operating cost composition

OPEX is dominated by ERS energy purchases among served tours; SCS energy costs are modest, while ERS corridor fees and time costs remain secondary. Penalty costs for the residual infeasible tours (diesel and outsourcing surcharge) are substantial on a per-tour basis and therefore mark the value of targeted investments that convert those penalties into electric operating costs (Fig. 11).

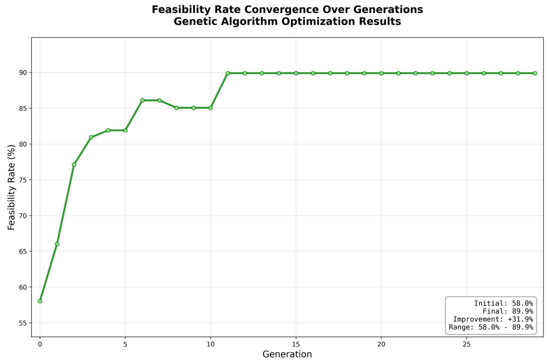
5.6 Takeaways

The hybrid design—ERS on long spines, SCS at high-interaction nodes, and heterogeneous battery assignment—achieves high electric service rates at competitive system cost. The joint trajectories of feasibility, CAPEX, and OPEX (Fig. 7) demonstrate the underlying trade-off: higher upfront corridor investment reduces recurring penalties and energy costs while avoiding blanket oversizing of batteries. The residual infeasibility analysis (Figs. 9a–9b) provides a clear, actionable shortlist for incremental expansion.

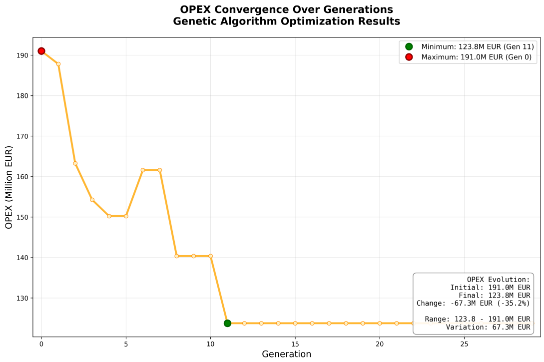
Figures for Chapter 5



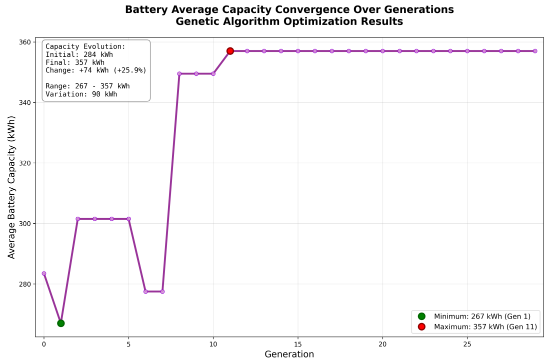
(a) Fitness convergence



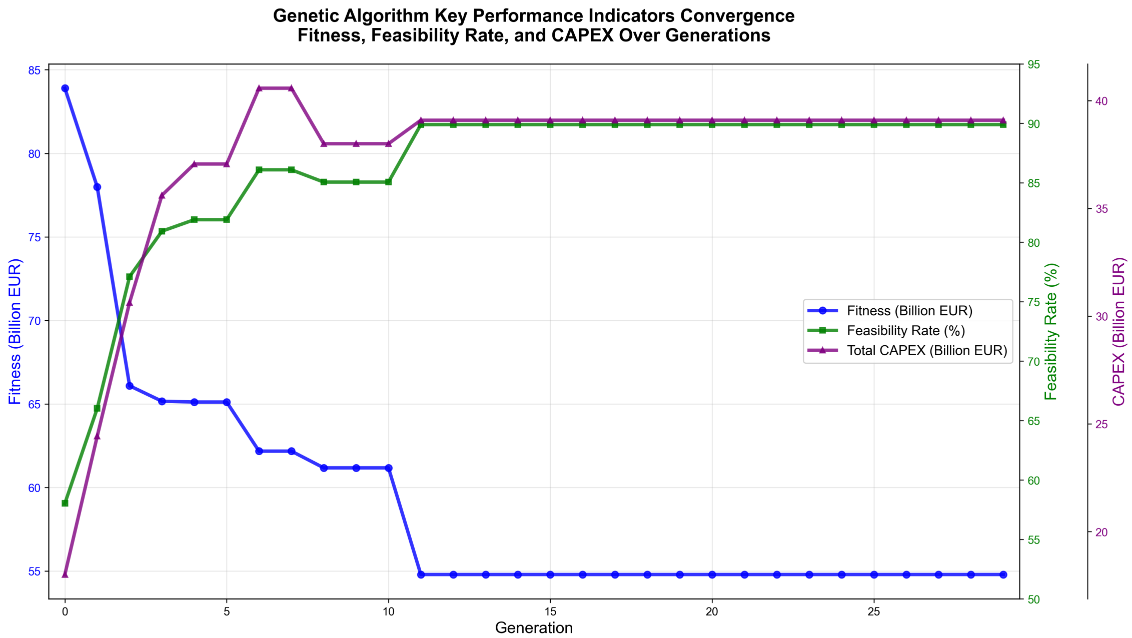
(b) Feasibility rate



(c) OPEX convergence

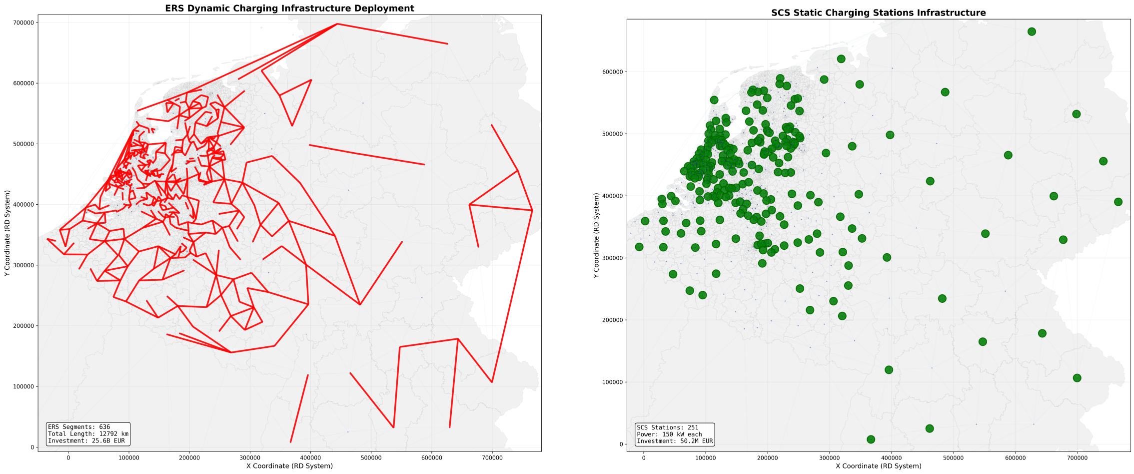


(d) Average battery capacity



(e) Joint KPIs: fitness, feasibility, and CAPEX

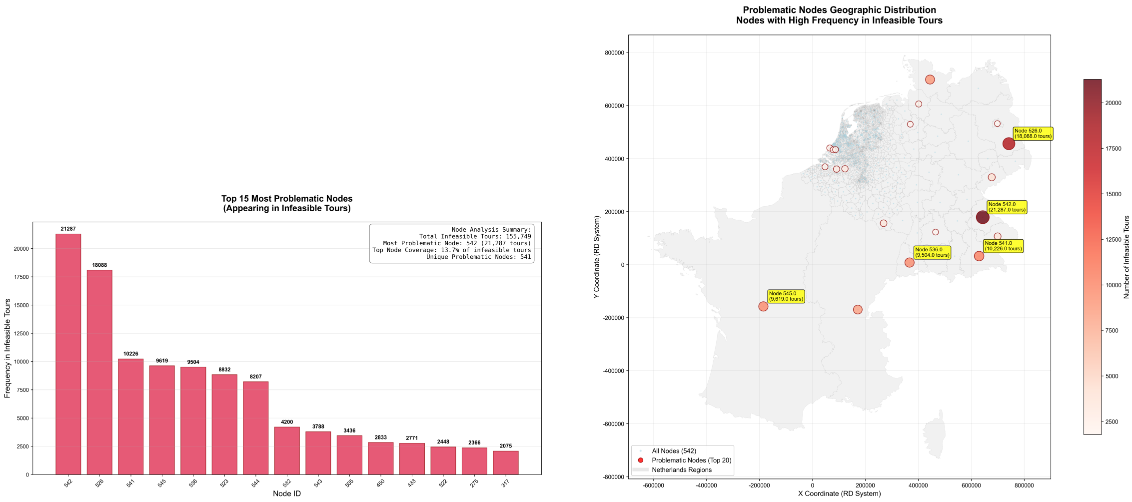
Figure 7: Genetic Algorithm convergence: fitness, feasibility, CAPEX/OPEX, and battery capacity across generations



(a) ERS deployment on corridors

(b) SCS locations at hubs and gateways

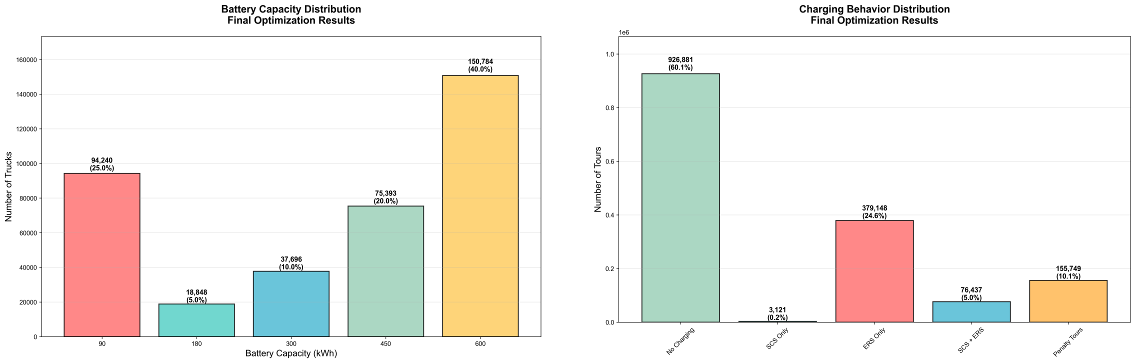
Figure 8: Optimised national infrastructure: ERS (left) and SCS (right).



(a) Top-N problematic nodes by unserved tours

(b) Spatial distribution of infeasibility

Figure 9: Residual infeasibility: node ranking (left) and geographic clusters (right).



(a) Fleet battery capacity distribution

(b) Tour-level charging behaviour

Figure 10: Post-optimisation fleet composition and operating behaviour.

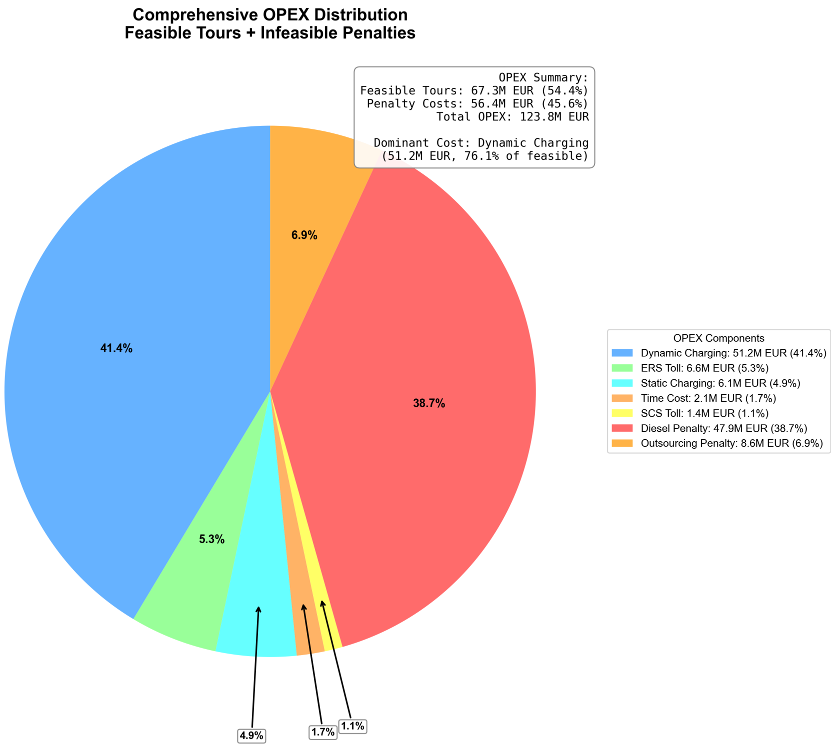


Figure 11: Operating cost breakdown under the optimised layout

6 Discussion

6.1 Interpretation of Key Findings

The national-scale optimisation reveals a consistent mechanism linking investment, feasibility, and cost. By expanding the ERS backbone and adding a limited number of SCS at high-interaction hubs, the GA raises the electric service rate from 58.0% to 89.9% while improving the objective by 34.7%. CAPEX increases to €39.1 bn, with €25.7 bn (65.6%) devoted to ERS and less than 1% to SCS, indicating that corridor energy—not extensive nodal hardware—bears most of the capital burden. Battery assignment converges to an average of 357 kWh, yet the fleet is intentionally heterogeneous: 25% of trucks operate with 90 kWh while 40% carry 600 kWh, and the remainder populate intermediate classes. This mix confirms that corridor energy substitutes for on-board storage on long spines, while selective upsizing covers ERS-sparse missions. Operating expenditures are modest: feasible-tour OPEX is dominated by ERS energy and corridor fees, whereas penalties for residual infeasible tours (10.1%) remain costly on a per-tour basis, marking the value of targeted extensions at peripheral and cross-border bottlenecks. Together, these patterns show that higher upfront CAPEX is traded for fewer infeasible tours and lower recurring costs, yielding a lower long-run system cost.

6.2 Trade-offs Between Cost, Feasibility, and Investment

The joint trajectories of fitness, feasibility, and CAPEX demonstrate a clear trade-off (Fig. 7e). Large-scale ERS investment reduces the number of tours that require very large batteries, thereby lowering battery CAPEX relative to a homogeneous-pack baseline, while SCS provides low-cost redundancy where SOC headroom would otherwise be binding. The persistence of a small infeasible share concentrates in a handful of nodes that together account for more than 40% of all unserved tours; incremental ERS links that shorten the longest (or second-longest) legs into these gateways, or a single SCS near the top-ranked nodes, can convert penalties into operating cost at favourable rates. In short, corridors buy systemic feasibility, hubs buy local robustness, and heterogeneous batteries prevent over-investment.

6.3 Comparison with Literature

These findings align with and extend recent work on integrated charging strategies. Prior models show that co-optimising static and dynamic charging can lower total cost and reduce the need for oversized batteries (Sun, Chen, and Yin 2020). Corridor-focused studies indicate that ERS enables substantial battery downsizing for long-haul missions, conditional on sufficient traffic density (Liao et al. 2024). Our tour-based results corroborate both points while adding operational realism: by preserving SOC continuity across chained legs, we identify a tail of duties that still needs very large packs even with ERS, and we quantify cost components (CAPEX/OPEX/penalties) at national scale. Methodologically, moving from trip-based to tour-based inputs addresses the known optimism bias of trip-wise siting models and is consistent with the urban freight and tour-synthesis literature (Boerkamps and Binsbergen 1999; Sánchez-Díaz, Holguín-Veras, and Ban 2015; Thoen et al. 2020). Policy-facing simulations similarly argue that ERS is a no-regrets lever when used at scale; our evidence shows where it should be complemented by SCS to close residual gaps (Rogstadius et al. 2025; Inez 2024).

6.4 Limitations

Several modelling choices trade realism for tractability. First, the corridor graph aggregates routes to VAM links and uses shortest paths; geography outside the Netherlands is coarser, which inflates single-leg distances on cross-border tours and partly explains residual infeasibility. Second, lower-level operations use deterministic energy balances without explicit queueing, time windows, or power-sharing; these effects are conservatively bounded rather than simulated. Third, prices and tolls are applied as national averages; spatially differentiated tariffs could sharpen siting priorities. Fourth, the GA provides heuristic solutions at full scale (validated against exact MILP on small instances), so the national run is near-optimal rather than proven optimal. These caveats are transparent and, in our tests, do not overturn the qualitative insights about corridor–node complementarity and battery heterogeneity. :contentReference[oaicite:2]index=2

6.5 Future Work

Two extensions are most promising. (i) **Operational realism:** embed time windows, charger capacity, and stochastic dwell into the lower level to quantify queueing-induced detours, and test simple congestion surrogates for peak periods. (ii) **Dynamic planning:** couple multi-year investment with learning curves for batteries and ERS, and endogenise adoption so that technology shares, duty allocation, and charging demand co-evolve. Additional priorities include finer corridor resolution in peripheral regions, explicit grid connection costs for SCS/ERS, and robust optimisation under energy-price and toll uncertainty. These steps would refine, rather than overturn, the central conclusion: ERS as the long-haul backbone, SCS as nodal redundancy, and heterogeneous batteries as cost optimisers at fleet level.

7 Conclusion

This paper develops a nationwide, tour-based planning framework that jointly optimises in-motion ERS, SCS, and heterogeneous battery capacities at the vehicle level. By linking strategic investments to tour-level feasibility, the approach attains a high electric service rate at competitive system cost and identifies where targeted expansions yield the largest marginal gains. In the national case study, the optimised layout deploys approximately 12,792 km of ERS and 251 SCS sites, serves 89.9% of tours electrically, and converges to an average assigned battery of 357 kWh while reducing operating expenditure through greater reliance on ERS energy.

Main contributions

1. Tour-based electrification at national scale. The model evaluates feasibility on complete tours rather than isolated trips, preserving state-of-charge continuity across chained legs and empty returns. This closes the optimism gap of trip-based siting and reveals where energy headroom is genuinely binding for operations.

2. Integrated design of ERS, SCS, and fleet batteries. Corridor energy, nodal redundancy, and vehicle storage are co-optimised in a single problem, capturing their substitution and complementarity. The solution uses ERS as the long-haul backbone, SCS as low-cost redundancy at hubs and gateways, and a heterogeneous battery mix aligned with duty cycles.

3. Bi-level architecture with a full mathematical specification. Strategic siting and battery assignment are coupled to a lower-level tour-feasibility and operating-cost evaluation. The main text presents the modelling principles and illustrative equations, and the complete sets, variables, objective, and constraints are documented in the appendix for reproducibility.

4. National-scale optimisation via a tailored genetic algorithm. A three-block chromosome (ERS, SCS, batteries) with elitist selection, block-aligned crossover, sparsity-aware mutation, light repair, and diversity maintenance enables efficient exploration. Fitness evaluation is parallelised at the tour level, providing a practical recipe where exact methods are not tractable at national size.

5. Validation and interpretability. On small instances, the heuristic matches exact MILP optima, establishing correctness of the modelling stack. At full scale, convergence traces and decomposed key performance indicators clarify how improvements arise rather than treating the optimiser as a black box.

6. Actionable, map-based outputs. The results provide a layered national layout, a transparent battery portfolio for procurement, and a shortlist of bottleneck nodes and corridors where incremental links or single SCS sites convert penalty costs into electric operating expenditure.

7. Reproducibility and transfer. The pipeline delivers a clean handoff from data to optimisation: tour reconstruction on a sparse corridor graph, canonical link and node tables for siting, and figure-ready outputs. The framework is portable to other regions given tour data or credible tour synthesis.

Closing remark

Electrifying heavy freight at scale is a design problem that aligns corridor energy, nodal redundancy, and fleet heterogeneity with how tours actually run. The framework and evidence presented here offer a practical path for that alignment at national scale.

References

Akbari, Milad, Morris Brenna, and Michela Longo. 2018. "Optimal Locating of Electric Vehicle Charging Stations by Application of Genetic Algorithm." *Sustainability* 10 (4): 1076.

BloombergNEF. 2025. "Lithium-Ion Battery Pack Prices See Largest Drop Since 2017, Falling to $115/kWh." Accessed 2025-08-24. https://about.bnef.com/blog/lithium-ion-battery-pack-prices-2024/.

Boerkamps, Jeroen, and Arjan van Binsbergen. 1999. "GOODTRIP: A New Approach for Modelling and Evaluation of Urban Goods Distribution." *City Logistics* 1.

Cargopedia. 2025. "European Fuel Prices." Accessed 2025-08-24. https://www.cargopedia.net/europe-fuel-prices.

Cintrano, Christian, Jamal Toutouh, and Enrique Alba. 2021. "Citizen-Centric Optimal EV Charging Station Locations in a Full City: Case of Malaga." In *Spanish Conference on Metaheuristics and AI (CAEPIA/Assoc. for AI),* 247–257. Springer.

Decisio. 2022. *Cost-Effectiveness of Electric Road Systems in the Netherlands.* Technical report. Decisio. https://decisio.nl/wp-content/uploads/Cost-effectiveness-ERS-in-the-Netherlands-Decisio-March-2022.pdf.

DG MOVE. 2014. *Update of the Handbook on External Costs of Transport.* Technical report. Accessed 2025-08-24. European Commission. https://ec.europa.eu/transport/themes/sustainable/studies/doc/2014-handbook-external-costs-transport.pdf.

European Transport Safety Council (ETSC). 2024. "Europe-wide Truck Speed Limits around 80 km/h." Accessed 2025-08-24. https://etsc.eu/speed-data/.

Heliox Energy. 2025. "Rapid 150 kW EV Charging Station (Modular)." Accessed 2025-08-24. https://www.heliox-energy.com/products/rapid-150kw-modular.

Honda. 2021. "Electric Road System (ERS): Technology Overview." *Honda Global ITS,* https://global.honda/en/safety/ITS/2021/pdf/07_Honda_Electric_Road_System.pdf.

Honda R&D Co., Ltd. 2021. "Honda Electric Road System – Charging Power Capabilities." Accessed 2025-08-24. https://global.honda/en/safety/ITS/2021/pdf/07_Honda_Electric_Road_System.pdf.

Inez, A. 2024. *Hybrid SCS–ERS Planning in the Netherlands: A Bi-level MILP.* Nationwide case study (trip-based); add publisher/DOI when available.

International Council on Clean Transportation (ICCT). 2018. "Comparison of Fuel Consumption and Emissions for Representative HDVs in Europe." Accessed 2025-08-24. https://theicct.org/wp-content/uploads/2021/06/HDV-EU-fuel-consumption_ICCT-Fact-Sheet_08042018_vF.pdf.

Kunawong, Kittipong, and coauthors. 2025. "Two-layer MILP for Long-distance Bus Networks: Charger Siting and Scheduling." As cited in thesis; update venue when available, *To appear / venue TBD.*

Liao, Ximeng, Mahnam Saeednia, Maria Nogal, and Lóránt Tavasszy. 2024. "Scaling Up Dynamic Charging Infrastructure: Significant Battery Cost Savings." *Transportation Research Part D: Transport and Environment* 129:104128.

Nelder, Chris, and Emily Rogers. 2019. *Reducing EV Charging Infrastructure Costs.* Technical report 10. Accessed 2025-08-24. Rocky Mountain Institute. https://rmi.org/insight/reducing-ev-charging-infrastructure-costs/.

NL Times. 2024. "Electric Car Charging Costs Vary Drastically Across Dutch Cities." Accessed 2025-08-24. https://nltimes.nl/2024/03/11/electric-car-charging-costs-vary-drastically-across-dutch-cities.

Piedel, Emma, Enrico Lauth, Alexander Grahle, and Dietmar Göhlich. 2024. "Review and Evaluation of Automated Charging Technologies for Heavy-Duty Vehicles." *World Electric Vehicle Journal* 15 (6).

PTV Logistics. 2025. "Introduction to Truck Toll Costs in Logistics." Accessed 2025-08-24. https://www.ptvlogistics.com/en/solutions/truck-toll-cost.

Rogstadius, Jakob, Hampus Alfredsson, Henrik Sällberg, and Karl-Filip Faxén. 2023. *Electric Road Systems: A No-Regret Investment with Policy Support.* Unpublished manuscript.

———. 2025. "Correcting Market Failure for No-Regret Electric Road Investments Under Uncertainty." *Nature Communications* 16 (1): 1–19.

Sánchez-Díaz, Iván, José Holguín-Veras, and Xuegang Jeff Ban. 2015. "A Time-dependent Freight Tour Synthesis Model." *Transportation Research Part B: Methodological* 78:144–168.

Saxe, Christopher de, Daniel Ainalis, John Miles, Phil Greening, Adam Gripton, Christopher Thorne, and David Cebon. 2023. "Vehicle and Charging Requirements for an Electrified Road Freight System Demonstrator in the UK." *Transportation Research Procedia* 72:4033–4040.

Seilabi, Saeid, and coauthors. 2025. "A GA-based Bi-level Framework for Sustainable EV Charging Station Planning." As cited in thesis; update venue when available, *To appear / venue TBD.*

Shoman, Wasim, Sten Karlsson, and Sonia Yeh. 2022. "Benefits of an Electric Road System for Battery Electric Vehicles." *World Electric Vehicle Journal* 13 (11): 197.

Shoman, Wasim, Sonia Yeh, Frances Sprei, Patrick Plötz, and Daniel Speth. 2023. "Battery Electric Long-Haul Trucks in Europe: Public Charging, Energy, and Power Requirements." *Transportation Research Part D: Transport and Environment* 121:103825. https://doi.org/10.1016/j.trd.2023.103825.

Sun, Xiaotong, Zhibin Chen, and Yafeng Yin. 2020. "Integrated Planning of Static and Dynamic Charging Infrastructure for Electric Vehicles." *Transportation Research Part D: Transport and Environment* 83:102331.

Tap Electric. 2023. "How Much Does It Cost to Charge an Electric Car in the Netherlands in 2025?" Accessed 2025-08-24. https://www.tapelectric.app/blog/cost-to-charge-an-electric-car-netherlands/.

Thoen, Sebastiaan, Lóránt Tavasszy, Michiel de Bok, Gonçalo Correia, and Ron van Duin. 2020. "Descriptive Modeling of Freight Tour Formation: A Shipment-based Approach." *Transportation Research Part E: Logistics and Transportation Review* 140:101989.

Vazifeh, Mohammad M., Hongmou Zhang, Paolo Santi, and Carlo Ratti. 2019. "Optimizing the Deployment of Electric Vehicle Charging Stations Using Pervasive Mobility Data." *Transportation Research Part A: Policy and Practice* 121:75–91.

Wikipedia. 2025a. "Carbon Price." Accessed 2025-08-24. https://en.wikipedia.org/wiki/Carbon_price.

———. 2025b. “LKW-Maut (Heavy Goods Vehicle Toll in Germany).” Accessed 2025-08-24. https://en.wikipedia.org/wiki/LKW-Maut.

Zeng, Bo, and Yu Zhang. 2020. “Bilevel Optimization for Transportation and Energy Systems: A Survey.” Use actual volume/issue/pages if different in your source, *IISE Transactions* 52 (4): 1–24.

Appendix A — Mathematical Model

A.1 Sets and Indices

$\mathcal{N}$ VAM zones / planning nodes (candidates for SCS), $n \in \mathcal{N}$.

$\mathcal{L}$ Undirected inter-zonal links (candidates for ERS), $\ell \in \mathcal{L}$.

$\mathcal{K}$ Trucks, $k \in \mathcal{K}$.

$\mathcal{T}$ Tours (ordered chains of trips), $t \in \mathcal{T}$.

$\mathcal{B}$ Battery-capacity classes, $b \in \mathcal{B}$.

$\mathcal{I}_t$ Ordered segments of tour t along its reconstructed corridor path, $i \in \mathcal{I}_t$.

Auxiliary mappings for each tour t and segment $i \in \mathcal{I}_t$:

$$\kappa(t) \in \mathcal{K} \text{ (truck assigned to tour } t), \quad n(t,i) \in \mathcal{N} \text{ (start node of segment } i), \quad \ell(t,i) \in \mathcal{L} \text{ (link traversed in } i), \quad d_{t,i} =$$

All routing is on the VAM corridor graph; distances $d_{t,i}$ are corridor (shortest-path) lengths used consistently for operations and investment accounting.

A.2 Parameters

c_n^{SCS}	CAPEX of an SCS at node n (€/site)
c_ℓ^{ERS}	CAPEX of ERS on link ℓ per km (€/km)
d_ℓ	Length of link ℓ (km); $d_{t,i}$ is the segment length (km)
c_b^{Bat}	Battery CAPEX for class b (€/truck)
Q_b	Battery capacity of class b (kWh)
α	Initial SOC fraction at tour start ($\mathrm{SOC}_{t,1} = \alpha\, Q_{\kappa(t)}$)
β	Traction energy intensity (kWh/km)
P^{SCS}	SCS charge power (kW); efficiency η^{SCS}
P^{ERS}	ERS in-motion power (kW) or per-km gain $\gamma_\ell = P^{\mathrm{ERS}}/v_{\mathrm{avg}}$; efficiency η^{ERS}
p^{stat}, p^{dyn}	Electricity prices at SCS / on ERS (€/kWh)
τ^{SCS}, τ_ℓ^{ERS}	SCS session fee (€/charge), ERS corridor fee per km (€/km)
c^{time}	Value of time used for SCS dwell pricing (€/h) (optional)
p_{diesel}, f_{diesel}	Diesel price (€/L), diesel intensity (L/km)
$e_{\mathrm{diesel}}^{\mathrm{CO2}}$, λ_{CO2}	Diesel CO_2 factor (g/L), carbon price (€/g)
φ	Additional outsourcing surcharge (€/km)
Ω	Feasibility weight on unserved tours (unitless; solver guidance)
M_{SOC}, M_{E}	Big-M constants (tight upper bounds; see A.7)

A.3 Decision Variables

Upper level

$$x_n \in \{0,1\} \text{ (build SCS at node } n), \quad y_\ell \in \{0,1\} \text{ (deploy ERS on link } \ell), \quad \delta_{k,b} \in \{0,1\} \text{ (truck } k \text{ uses battery class } b).$$

Lower level (per tour t, segment i)

$\mathrm{SOC}_{t,i} \geq 0$	state of charge at segment start (kWh)
$e^{\mathrm{SCS}}_{t,i} \geq 0,\ e^{\mathrm{ERS}}_{t,i} \geq 0$	charged energy on SCS/ERS (kWh)
$z^{\mathrm{SCS}}_{t,i},\ z^{\mathrm{ERS}}_{t,i} \in \{0,1\}$	activation of SCS/ERS at (t,i)
$s_t \in \{0,1\},\ r_t \in \{0,1\}$	served electrically vs. outsourced, with $s_t = 1 - r_t$.

Define truck capacity $Q_k = \sum_{b\in\mathcal{B}} Q_b\,\delta_{k,b}$.

A.4 Objective Function

$$\min Z \;=\; \underbrace{\Omega \sum_{t\in\mathcal{T}} r_t}_{\text{service-first guidance}} \;+\; \underbrace{\sum_{n\in\mathcal{N}} c^{\mathrm{SCS}}_n x_n + \sum_{\ell\in\mathcal{L}} c^{\mathrm{ERS}}_\ell d_\ell y_\ell}_{\text{facility CAPEX}} \;+\; \underbrace{\sum_{k\in\mathcal{K}} \sum_{b\in\mathcal{B}} c^{\mathrm{Bat}}_b\,\delta_{k,b}}_{\text{battery CAPEX}} \;+\; \sum_{t\in\mathcal{T}} \Big[s_t\,\mathrm{OPEX}_t + r_t\,C^{\mathrm{pen}}_t \Big]. \tag{6}$$

OPEX for served tours (priced energy, time, and corridor fees):

$$\mathrm{OPEX}_t = \sum_{i\in\mathcal{I}_t} \Big(p^{\mathrm{stat}} e^{\mathrm{SCS}}_{t,i} + p^{\mathrm{dyn}} e^{\mathrm{ERS}}_{t,i} + c^{\mathrm{time}} \frac{e^{\mathrm{SCS}}_{t,i}}{P^{\mathrm{SCS}}} + \tau^{\mathrm{SCS}} z^{\mathrm{SCS}}_{t,i} + \tau^{\mathrm{ERS}}_{\ell(t,i)}\, d_{t,i}\, z^{\mathrm{ERS}}_{t,i} \Big). \tag{7}$$

Penalty for unserved tours (outsourcing / diesel cost + carbon + surcharge):

$$C^{\mathrm{pen}}_t = \Big(\sum_{i\in\mathcal{I}_t} d_{t,i} \Big) \cdot \Big(p_{\mathrm{diesel}} f_{\mathrm{diesel}} + \lambda_{\mathrm{CO2}}\, e^{\mathrm{CO2}}_{\mathrm{diesel}} + \varphi \Big). \tag{8}$$

A.5 Core Constraints

(B) Battery assignment (upper level)

$$\sum_{b\in\mathcal{B}} \delta_{k,b} = 1 \qquad \forall k\in\mathcal{K}, \tag{9}$$

$$\delta_{k,b} \in \{0,1\} \qquad \forall k, b, \tag{10}$$

$$Q_k = \sum_{b\in\mathcal{B}} Q_b\,\delta_{k,b} \qquad \forall k. \tag{11}$$

(S) Service–infeasibility link

$$s_t + r_t = 1 \qquad \forall t\in\mathcal{T}. \tag{12}$$

(I) Initial SOC and bounds

$$\mathrm{SOC}_{t,1} = \alpha\, Q_{\kappa(t)} \qquad \forall t, \tag{13}$$

$$0 \;\leq\; \mathrm{SOC}_{t,i} \;\leq\; Q_{\kappa(t)} + M_{\mathrm{SOC}}\, r_t \qquad \forall t,\ \forall i\in\{1,\ldots,|\mathcal{I}_t|+1\}. \tag{14}$$

(E) Segment energy balance (gated by infeasibility)

$$-M_{\mathrm{SOC}}\, r_t \;\leq\; \mathrm{SOC}_{t,i+1} - \mathrm{SOC}_{t,i} + \beta\, d_{t,i} - \eta^{\mathrm{SCS}} e^{\mathrm{SCS}}_{t,i} - \eta^{\mathrm{ERS}} e^{\mathrm{ERS}}_{t,i} \;\leq\; M_{\mathrm{SOC}}\, r_t, \quad \forall t,\ \forall i\in\mathcal{I}_t. \tag{15}$$

(C) Charging feasibility and headroom

$$
\begin{aligned}
&\text{SCS headroom/time:} && e^{\mathrm{SCS}}_{t,i} \le \big(Q_{\kappa(t)} - \mathrm{SOC}_{t,i}\big) + M_{\mathrm{E}}\, r_t, && \forall t, i, && (16)\\
& && e^{\mathrm{SCS}}_{t,i} \le P^{\mathrm{SCS}} \cdot \Delta t^{\mathrm{SCS}}_{t,i} + M_{\mathrm{E}}\, r_t, && \forall t, i, && (17)\\
&\text{ERS headroom/time:} && e^{\mathrm{ERS}}_{t,i} \le \big(Q_{\kappa(t)} - \mathrm{SOC}_{t,i}\big) + M_{\mathrm{E}}\, r_t, && \forall t, i, && (18)\\
& && e^{\mathrm{ERS}}_{t,i} \le \gamma_{\ell(t,i)}\, d_{t,i} + M_{\mathrm{E}}\, r_t, && \forall t, i. && (19)
\end{aligned}
$$

Here $\Delta t^{\mathrm{SCS}}_{t,i}$ is the available charging time at the stop (if queueing/windows are not modeled, one may use a policy-driven upper bound in evaluation).

(A) Activation and siting consistency

$$
\begin{aligned}
e^{\mathrm{SCS}}_{t,i} \le M_{\mathrm{E}}\, z^{\mathrm{SCS}}_{t,i}, \quad e^{\mathrm{ERS}}_{t,i} \le M_{\mathrm{E}}\, z^{\mathrm{ERS}}_{t,i} \qquad & \forall t, i, && (20)\\
z^{\mathrm{SCS}}_{t,i} \le x_{n(t,i)}, \quad z^{\mathrm{ERS}}_{t,i} \le y_{\ell(t,i)} \qquad & \forall t, i, && (21)\\
z^{\mathrm{SCS}}_{t,i},\ z^{\mathrm{ERS}}_{t,i} \le 1 - r_t \qquad & \forall t, i, && (22)\\
z^{\mathrm{SCS}}_{t,i},\ z^{\mathrm{ERS}}_{t,i} \in \{0, 1\}. \qquad & && (23)
\end{aligned}
$$

(N) Non-negativity

$$
e^{\mathrm{SCS}}_{t,i} \ge 0, \qquad e^{\mathrm{ERS}}_{t,i} \ge 0 \qquad \forall t, i. \tag{24}
$$

A.7 Implementation Notes and Big-M Choices

- **Service-first guidance:** Ω is a solver-internal weight that discourages infeasibility (outsourcing). It steers the search but is *not* counted in reported economic totals.
- **Tight Big-M bounds:** choose $M_{\mathrm{SOC}} = \max_k Q_k$, $\quad M_{\mathrm{E}} = \max\{\eta^{\mathrm{SCS}}\, Q_{\max},\ \eta^{\mathrm{ERS}} \cdot (P^{\mathrm{ERS}}/v_{\min}) \cdot d_{\max}\}$ to stabilize relaxations and avoid numerical issues.
- **Consistent metric:** all d_ℓ and $d_{t,i}$ are corridor (shortest-path) distances on the same graph used for siting and for energy/cost evaluation.
- **OPEX triggers:** fees τ may be triggered by activation indicators z; in implementation, one may also trigger by positive energy $e > 0$ without extra binaries.

Appendix B — Model Parameters

Table 1: Key parameters value used in the case study

Category	Symbol	Base value	Notes / Source
Feasibility weight	Ω	100,000	Solver-only guidance to discourage outsourcing; excluded from reported economic totals.
ERS CAPEX	c^{ERS}	2.0 M EUR/km	Within reported ERS cost ranges (1.7–3.1 M EUR/km for catenary; 0.4–2.7 M EUR/km for other types) [Shoman, Karlsson, and Yeh 2022].
SCS CAPEX	c^{SCS}	200,000 EUR/site	150 kW DC hardware typically $75.6k–100k; full-site installs 3–5× higher (RMI) [Nelder and Rogers 2019].
Battery CAPEX	c^{Bat}	100 EUR/kWh	Li-ion pack price around $115/kWh in 2024 with significant YoY decline (adopt 100 EUR/kWh) [BloombergNEF 2025].
Energy intensity	β	1.6 kWh/km	Heavy-duty BEVs reported 1.08–1.30 kWh/km; urban/highway 1.2–1.8 kWh/km (choose 1.6) [Shoman, Karlsson, and Yeh 2022; Shoman et al. 2023].
Static electricity price	p^{stat}	0.73 EUR/kWh	NL public-charging benchmarks; fast DC averages around 0.67–0.86 EUR/kWh, mean ≈ 0.76 EUR/kWh [Tap Electric 2023; NL Times 2024].
ERS electricity price	p^{dyn}	0.36 EUR/kWh	Aligned with typical non-fast public-charging averages [Tap Electric 2023].
Road toll (static)	$c^{\mathrm{toll}}_{\mathrm{stat}}$	0.15 EUR/km	Germany HGV toll (LKW–Maut) about 0.15 EUR/km; NL per-km charging from 2026 similar [Wikipedia 2025b; PTV Logistics 2025].
Road toll (ERS)	$c^{\mathrm{toll}}_{\mathrm{dyn}}$	0.10 EUR/km	Lower fee reflecting dynamic-charging efficiencies / marginal infrastructure costs [PTV Logistics 2025].
SCS power	P^{SCS}	150 kW	Typical continuous rating for ultra-fast DC stations (e.g., Heliox) [Heliox Energy 2025].
ERS power (effective)	P^{ERS}	200 kW	Conservative dynamic-charging assumption; conductive overhead up to 450 kW in demos [Honda R&D Co., Ltd. 2021].
Average speed	v^{avg}	80 km/h	European HGV legal/typical motorway speeds [European Transport Safety Council (ETSC) 2024].
Value of driver time	c^{time}	38 EUR/h	Converted from UK COBA values and adjusted for freight operations [DG MOVE 2014].
Diesel price	p_{diesel}	1.60 EUR/L	Representative EU retail prices (e.g., BE 1.603 EUR/L; DK 1.797 EUR/L) [Cargopedia 2025].
Diesel use	f_{diesel}	0.35 L/km	Slightly above ICCT long-haul average 0.326 L/km [International Council on Clean Transportation (ICCT) 2018].
CO_2 cost	λ_{CO2}	0.00008 EUR/g	EU-ETS benchmark around 100 EUR/ton CO_2 (= 0.0001 EUR/g); adopt 0.00008 EUR/g [Wikipedia 2025a].
Outsourcing surcharge	φ	0.10 EUR/km	Conservative per-km surcharge capturing marginal external/operational costs.