International Journal of Intelligent Transportation Systems Research (2026) 24:1925–1961
https://doi.org/10.1007/s13177-026-00715-9

An Optimization Framework for Automated and Human-Operated Fleet Routing

Andrea Appolloni[1] · **Massimiliano Caramia**[2]

Received: 1 September 2025 / Revised: 24 July 2026 / Accepted: 14 August 2026 / Published online: 23 September 2026
© The Author(s) 2026

Abstract
This paper presents a mathematical model to determine the optimal routes for a heterogeneous fleet comprising autonomous and human-driven vehicles operating in freight distribution. The model is formulated as a mixed-integer programming problem with multiple objectives: minimize total emissions, energy consumption, and operational costs, while maximizing the reliability associated with autonomous vehicle deployment. A dedicated parameter models the probability that an autonomous vehicle can successfully traverse a given arc, enforcing a minimum reliability threshold. The proposed optimization framework is embedded within a higher-level decision support system that enables route adjustments based on discrete-time monitoring of network reliability conditions. Since the multi-objective model is solved by means of scalarization, a specific algorithm is designed to adjust the weights of the objectives. A case study is provided to demonstrate the effectiveness of the model and its practical applicability. To further enhance the performance of the proposed model in solving larger instances, three families of valid inequalities are presented, and a computational comparison between the basic model and the improved model is shown in a single-objective environment (operational cost minimization). We also implement a constructive heuristic and an Adaptive Large Neighborhood Search metaheuristic to further appreciate the reduced duality gaps achieved by the enhanced model.

Keywords Industry 5.0 · Autonomous vehicle · Mathematical programming · Routing

1 Introduction

Industry 5.0 represents the next evolution in industrial paradigms, highlighting the synergy between advanced technologies and human expertise to develop sustainable and efficient solutions [5]. In the context of transportation and logistics, this transition promotes collaboration between artificial intelligence (AI) systems and human-operated fleets, thus improving decision-making processes and operational outcomes.

Transportation 5.0 extends these principles to mobility systems, advocating for a user-centered approach that leverages AI technologies and human judgment to improve operational efficiency, safety, and environmental sustainability [31]. The core components include connected and autonomous vehicles (AVs), intelligent infrastructure, and integration of renewable energy. AVs are equipped with AI-based systems capable of real-time data processing, enabling autonomous decision-making while still allowing for human intervention when necessary. Intelligent infrastructure comprises sensors and IoT devices deployed across the network to monitor and manage traffic conditions, facilitating seamless communication between vehicles and infrastructure nodes.

The coexistence of AI-driven and human-driven vehicles (HDVs) requires the development of advanced decision-making models to support optimal fleet management. These models address key operational aspects, including route planning, energy management, and human-machine collaboration. In addition, AI algorithms are required to analyze

✉ Massimiliano Caramia
caramia@dii.uniroma2.it

Andrea Appolloni
andrea.appolloni@uniroma2.it

[1] Dipartimento di Management e Diritto, University of Rome Tor Vergata, Via Columbia, 2, 00133 Rome, Italy

[2] Dipartimento di Ingegneria dell'Impresa, University of Rome Tor Vergata, Via del Politecnico, 1, 00133 Rome, Italy

traffic patterns, forecast demand, and assess vehicle statuses to support optimal routing and task assignment decisions.

Despite its potential, the implementation of Industry 5.0 in transportation and logistics faces several challenges, such as technological integration, regulatory compliance, and societal acceptance. Industry 5.0 frames deployment as a socio-technical transition where compliance, legitimacy, and stewardship matter as much as performance. Regulatory readiness requires demonstrable conformance to an approved operational design domain, evidence of functional safety under representative edge cases, and auditable change control for software, data, and procedures; authorities increasingly expect traceable decisions, calibrated risk estimates, and documented incident handling that can be independently reviewed. Societal acceptance hinges on visible safety outcomes, perceived fairness of service across neighborhoods and times, and clear lines of accountability. Public trust grows when users can understand why decisions were made, when privacy and security protections are explicit, and when performance guarantees (e.g., service continuity under disruptions) are communicated and met. Operational governance connects these expectations to day-to-day practice: organizations should define roles and escalation paths, maintain incident response playbooks with bounded recovery times, monitor equity, safety, and environmental indicators alongside cost, and commit to periodic external audits. In the Industry 5.0 view, technology is adopted sustainably when regulators can verify compliance ex-ante and ex-post, communities see tangible benefits without disproportionate burdens, and operators operate with transparent rules, repeatable procedures, and measured results.

Ensuring compatibility among diverse systems and technologies is essential, as is the creation of policies that both accommodate emerging technologies and ensure public safety. Gaining stakeholder trust regarding the reliability and advantages of integrated AI-human solutions is also a critical requirement.

In this paper, we address the problem of defining optimal routing for a heterogeneous fleet of AVs and HDVs operating in a freight distribution scenario. The problem of routing a mixed AV/HDV fleet is especially relevant in transitional logistics environments, where autonomous vehicles can be deployed only under specific operational conditions rather than across the entire transportation network. Such conditions include the presence of suitable infrastructure, adequate communication support, regulatory admissibility, sufficiently high reliability levels, and compatibility with the operational design domain of the AV technology. At the same time, some delivery tasks may still require human-driven vehicles because of longer distances, charging limitations, low-reliability corridors, or operational contexts in which autonomous operation is not yet sufficiently robust. In these situations, a mixed fleet provides a realistic and practically meaningful compromise: AVs can be assigned to the portions of the network where their environmental and operational advantages can be exploited, while HDVs ensure service continuity, flexibility, and full network coverage.

The logistics network considered includes depots, customer locations, supply nodes, and parking facilities. The underlying graph is assumed to be complete in terms of arcs connecting all node types. The proposed mixed-integer programming (MIP) formulation captures multiple objectives: minimizing total emissions, energy consumption, and operational costs while simultaneously maximizing the reliability of AV assignments. Beyond classical routing constraints, the model accounts for AV recharge operations at specific parking nodes, tour-time limits, and compatibility requirements between vehicles and both (i) the arcs to be traversed and (ii) the operations to be performed at each node. In addition, the novelty of our proposal is related to the introduction of a specific parameter to model the reliability of an AV traversing a given arc, enabling the integration of probabilistic risk into routing decisions. In fact, the model is embedded in an optimization framework that allows routing adjustments based on real-time network conditions related to the changing reliability of using AVs in the network. Further pursuing this aspect, we note that, despite the growing literature on autonomous freight systems, electric vehicle routing, and uncertainty-aware logistics optimization, several research gaps remain. First, most available studies either focus on fully autonomous or fully conventional fleets, while limited attention has been devoted to routing problems in which AVs and HDVs coexist within the same operational system and must be allocated jointly. Second, when uncertainty or risk is considered, it is usually modeled through cost, travel time, or stochastic service terms, whereas the explicit integration of AV arc reliability as an enforceable routing-admissibility condition remains largely unexplored. Third, existing studies rarely combine this reliability dimension with energy constraints, node/arc compatibility, and a multi-objective optimization framework including economic, environmental, and operational criteria. These gaps motivate the present work, which develops a mixed AV/HDV routing framework in which AV deployment is explicitly conditioned on route reliability, while HDVs provide flexible fallback coverage in a realistic transitional logistics setting. In this sense, the paper is positioned at the intersection of autonomous-fleet routing, mixed-fleet logistics, and reliability-aware optimization.

However, our contribution is not a mere parameterization of reliability. In fact, this work contributes to the state-of-the-art along four axes.

(i) We operationalize AV reliability as a binary admissibility rule on arcs, controlled by a single threshold $\Lambda^{\min}$

computed from a route-level target with a feasibility cap. Unlike treating reliability as a soft objective or chance constraint, this creates policy-ready, geo-fenceable corridors while retaining HDV fallback.

(ii) We derive LP-relaxation-based stability indicators for the arc reliabilities. These provide a computationally cheap screening rule to decide when parameter drifts are large enough to warrant re-optimization. The dual multipliers are computed on the LP relaxation (where they are well-defined) and are used as an event-trigger for warm-started rerouting rather than as a certificate that the MIP optimum is unchanged.

(iii) We provide a lightweight, feedback-driven scheme to adapt the weights of emissions, energy, cost, and reliability across decision points, steering exploration toward well-balanced Pareto solutions without changing the solver or adding new variables.

(iv) We introduce three valid-inequality families (load propagation, state-of-charge recursion with tight big-M, and short infeasible energy-path cuts) that significantly reduce runtimes and gaps. Also, a constructive heuristic and an Adaptive Large Neighborhood Search implementation are introduced to benchmark the model solution quality and confirm scalability benefits.

The organization of the paper is as follows. In Section 2, we give details on the logistics research scenario considered and review the state-of-the-art on the topic. Section 3 defines the proposed model and the general methodology. Section 4 is devoted to analyzing how to cope with variations in reliability parameters associated with the use of AVs in the network. In Section 5, we describe how to embed the model and the reliability sensitivity analysis in an optimization framework that can take into account the changing reliability conditions in the network. Experiments are presented in Section 6; in the same section, we present valid inequalities to improve the formulation and further computational tests to assess their effectiveness. A three-phase heuristic and an Adaptive Large Neighborhood Search are also provided to validate the quality of the duality gaps produced by the enhanced model. Finally, Section 7 presents conclusions.

2 Context and Literature Review

Rapid advances in technology have significantly increased interest in services based on autonomous vehicles. Self-driving vehicles are already operational in geofenced environments, demonstrating their technical and commercial feasibility. For example, Waymo, a leader in autonomous vehicle technology, operates 24/7 ride-hailing services in designated zones within San Francisco, California, and Phoenix, Arizona [34].

During the past decade, substantial research has investigated the potential impact of AVs, particularly those classified as SAE Level 5, which are capable of fully autonomous operation under all conditions [27]. AV-based transport systems that autonomously fulfill mobility or delivery requests are expected to drastically reduce operational costs compared to conventional systems based on human drivers. As such, AVs hold significant promise in both the passenger transit and freight distribution sectors.

In freight distribution, AVs represent a major advance in logistics. By integrating artificial intelligence, electric propulsion, and real-time connectivity, these vehicles improve efficiency and sustainability. AVs operate autonomously using LiDAR, radar, high-resolution cameras, and AI-powered navigation systems, enabling precise maneuvering within complex logistics hubs, urban networks, and industrial corridors. Vehicles equipped with SAE Level 4 capabilities can function without human supervision in most conditions, ensuring seamless integration with intelligent warehouses and automated supply chain infrastructures.

The technical specifications of AVs vary considerably depending on their application. For instance, as for light-duty AVs, we can cite Nuro's zero-occupant delivery vehicle that provides two insulated cargo bays totaling about 27 ft^3 ($\approx$ 0.76 m^3) and supports payloads on the order of 500 lb; public reports indicate a battery around 31 kWh [24]. These platforms are operated at urban speeds and are optimized for short trips with frequent stop–start duty cycles rather than long range. In contrast, medium- to heavy-duty AVs intended for industrial freight or long-haul transport require larger energy storage and high-power chargers to reduce downtime and maintain logistics efficiency. In the middle-mile, Class-6 segment, Gatik operates autonomous box trucks commercially, with vehicles built on battery-electric platforms such as Lion Electric's Lion6, which specifies a 252 kWh pack, 250 kW peak motor power, and up to 350 km range with DC fast charging to 80% in about 2–2.5 h [14]. In the yard/short-haul heavy category, the cab-less Einride Pod has been piloted by DB Schenker on public roads in Jönköping (Sweden), with reported specs around 200 kWh battery capacity, $\sim$200 km range, 15 euro-pallets and up to $\sim$26 t GVW [7]. For light/medium domestic logistics in China, JD Logistics has fielded SAE L4 electric light trucks for depot-to-station distribution, reporting $\approx$24 m^3 cargo volume and $\approx$400 km per charge in pilot operations [2]. At the heavy, long-haul end, DeepWay (Baidu–Lionel) unveiled the Xingtu autonomous Class-8 tractor with a 450 kWh pack, $\sim$300 km stated range and high-power battery-swap aimed at minimizing dwell times on trunk routes [22].

Modern AVs are increasingly equipped with Vehicle-to-Everything (V2X) communication systems, enabling dynamic interaction with infrastructure, traffic management systems, and autonomous and human-driven vehicles [1]. These capabilities support real-time routing adjustments, congestion avoidance, and predictive control strategies. Advanced fleet management software, supported by cloud computing and AI analytics, enhances AV coordination and deployment in response to fluctuating operational conditions. These vehicles are also designed with multiple safety redundancies, including emergency stop systems, collision avoidance sensors, and adaptive braking to navigate diverse environments securely.

When analyzing infrastructure challenges and developments in vehicle technology, a substantial body of literature is available. The impacts of automated vehicles and strategic recommendations to enable a successful transition are discussed in several key studies. For example, Shahandasht et al. [30] reviews the current state of automated truck technologies, analyzing their effects on critical aspects of the trucking industry at varying levels of automation. The authors also designed a survey instrument to identify the risks and opportunities associated with adopting automated trucks at different automation levels.

In Monios and Bergqvist [20], the authors focus on the geography of electric and autonomous road freight transport, outlining a research agenda that highlights emerging and unresolved issues. A complementary perspective is offered in Beirigo et al. [4], which introduces a model that integrates passenger and freight requests using compartmentalized shared AVs, tackling a variation of the people- and freight-integrated transportation problem.

Monios and Bergqvist [21] contributes to this body of work by providing insights into the evaluation of optimal shipment sizes for autonomous trucks, targeting logistics managers seeking to assess technological readiness and operational efficiency. Meanwhile, Csiszár and Földes [8] poses foundational research questions regarding the structure and operation of future autonomous freight systems. Their approach involves analytical, system-oriented modeling to describe the shift from traditional freight transport systems to those based on autonomy. They model both system components and their interconnections using a process-driven framework.

In the context of last-mile delivery, Jennings and Figliozzi [15] examines autonomous delivery robots in the United States, comparing their travel time, cost, and efficiency with traditional van-based deliveries. The study also explores relevant regulatory issues. The technical transition required for vehicles to become fully autonomous is analyzed in depth by Masood et al. [19]; here, the authors provide a comprehensive overview of the necessary hardware and software adaptations, including sensor placement and system architecture.

Despite this breadth, there is significantly less literature that addresses mathematical models that integrate AVs into human-driven vehicle fleets, particularly models that incorporate AV reliability as a system-level concern. Notable exceptions include Guo et al. [13], where the authors develop a multimodal distribution model in which autonomous vehicles serve as both distribution tools and goods. The model integrates self-distribution into traditional logistics rings, forming a hybrid ring-branch network that seeks to minimize costs and maximize efficiency. To solve this complex model, an adaptive large neighborhood search algorithm is proposed and validated using real-world automotive logistics data.

Further evidence of AV potential is provided by Gružauskas et al. [12], who simulate a food logistics network incorporating AVs. Their results indicate a 22% reduction in CO_2 emissions and notable cost savings, facilitated by AVs eliminating the need for rest stops during trips. Similarly, De Maio et al. [9] introduces a routing problem involving autonomous delivery robots that leverage both road and public transit networks to extend operational range given limited battery capacity. The authors propose a destroy-and-repair heuristic embedded in a neighborhood search algorithm to solve large-scale instances. Their method yields cost reductions of up to 7.5% compared to conventional approaches.

In Wu et al. [35], the authors explore a future mobility scenario featuring a hybrid road network, where autonomous vehicle expressways coexist with traditional, manually driven local streets. Trips in this setting may involve both autonomous and human-driven segments. The study adopts a corridor-based modeling approach in which AV expressways and conventional roads run in parallel, connected through access and exit points. This structure represents a transitional phase in which AV expressways are first introduced along major routes. The research focuses on modeling travelers' route selection behavior in this dual-network environment and evaluates how the design and integration of AV expressways with surrounding local streets impact overall transportation system performance.

Lee and An [18] introduces a decision-making algorithm grounded in deep reinforcement learning to enhance fallback strategies for autonomous vehicles. These fallback mechanisms are designed to detect and respond to hazardous driving scenarios through timely evasive actions, which are critical for enabling safe and reliable operation in Level 4 and higher autonomous driving systems.

A branch of the research on VRP (Vehicle Routing Problem) and EVRP (Electric Vehicle Routing Problem) treats routing as a multi-objective and uncertainty-aware problem. On the multi-objective side, modern MIP and matheuristic models explicitly trade off operating cost with environmental

and service metrics and explore Pareto sets; for example, surveys and recent applications document MIP formulations that co-optimise distance, emissions, and service quality and report Pareto-efficient schedules in dynamic logistics settings [23, 37]. On the risk side, uncertainty is handled through two-stage stochastic programs with risk aversion, chance constraints, CVaR-type objectives, and robust and adaptive-robust sets. Illustrative contributions include a risk-averse latency location–routing model with stochastic travel times (two-stage with explicit risk trade-offs), chance-constrained CVRP (Constrained Vehicle Routing Problem) approximations via robust reformulations, and adaptive-robust EVRP formulations that hedge energy-consumption uncertainty with decomposition-based algorithms [16, 25, 32]. Against this backdrop, our reliability-gated mixed AV/HDV formulation addresses a distinct gap: reliability is operationalized not as a distributional risk term in the objective or as a probabilistic service-level constraint, but as an auditable admissibility threshold on AV arcs that can be tuned by policy while remaining compatible with standard MIP machinery.

To position the present contribution more explicitly with respect to the existing literature, Table 1 compares the proposed framework with a selected set of related studies already discussed in this section. The comparison highlights that the present paper differs from previous contributions in that it simultaneously considers a mixed AV/HDV fleet, an explicit reliability-gating mechanism for AV arc admissibility, energy-related constraints, node/arc compatibility restrictions, a multi-objective formulation, and a dynamic rerouting framework under changing reliability conditions. While several previous works address some of these aspects individually, their joint integration appears to be largely unexplored in the literature reviewed here.

3 Problem Setup and Mathematical Model

The logistics optimization problem considered occurs within a networked transportation system consisting of multiple depots, parking locations, customer locations, and supply nodes. The decision maker, a logistics operations manager, is responsible for managing this network efficiently by leveraging AI-driven and human-operated vehicle resources.

3.1 Network Structure

The transportation system is modeled as a directed graph $G = (N, A)$, where:

- N is the set of nodes, including:
  - D: Set of (capacitated) depots where vehicles are initially stationed or can be stationed at the end of a tour. Each vehicle in a depot starts and finishes the tour empty.
  - P: Set of (capacitated) parking locations where vehicles can be temporarily stored for maintenance or recharge.
  - C: Set of customer locations where deliveries must be made. Each customer is associated with a demand.
  - S: Set of (capacitated) supply nodes, i.e., storage points from which goods can be loaded to be delivered to customers.
- A: Set of directed arcs (i, j) representing the roads that connect nodes i, j (with $i \neq j$).
- d_{ij}: Distance between nodes $i, j \in N$, defined for each arc $(i, j) \in A$.

Let K^h be the set of HDVs and let K^a be the set of AVs. While AVs are electric, HDVs can have electric or thermal engines. In the case of thermal engine vehicles, we consider the refueling time to be negligible. Let $K^{he} \subseteq K^h$ be the subset of HDVs with an electric engine. To account for road and infrastructure constraints, we introduce the arc-vehicle compliance matrix:

- $\delta^a_{ijk}(\delta^h_{ijk})$: Binary parameter indicating whether arc $(i, j) \in A$ is compliant with vehicle $k \in K^a(K^h)$:

$$\delta^a_{ijk}(\delta^h_{ijk}) = \begin{cases} 1, & \text{if vehicle } k \text{ can traverse arc } (i,j) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Table 1 Comparison with selected related work

Reference	Mixed AV/HDV fleet	Explicit AV reliability	Energy constraints	Multi objective	Dynamic update/ rerouting
[9]	No	No	Yes	No	No
[12]	No	No	No	No	No
[13]	Yes	No	No	No	No
[16]	No	No	No	Yes	No
[25]	No	No	No	No	No
[32]	No	No	Yes	No	No
[35]	Yes	No	No	No	No
This paper	Yes	Yes	Yes	Yes	Yes

- $T^a_{ijk}(T^h_{ijk})$: Travel time required to traverse arc $(i, j) \in A$ using vehicle $k \in K^a(K^h)$, defined as:

$$T^a_{ijk}(T^h_{ijk}) = \frac{d_{ij}}{v_k}, \tag{2}$$

where v_k represents the average speed of vehicle k.

3.2 Mathematical Model

Parameters

- $T_{\max}$: Maximum time allowable for every tour.
- Q^a_k: Capacity of vehicle $k \in K^a$.
- Q^h_k: Capacity of vehicle $k \in K^h$.
- C^a_k: Carbon equivalent emissions per unit of distance traveled by vehicle $k \in K^a$.
- C^h_k: Carbon equivalent emissions per unit of distance traveled by vehicle $k \in K^h$.
- ρ_c: Demand at customer location $c \in C$.
- θ_s: Available supply at supply node $s \in S$.
- V_d: Maximum number of vehicles that can be stationed at depot $d \in D$.
- V_p: Maximum number of vehicles that can be temporarily stored at parking location $p \in P$.
- λ^k_{ij}: reliability of AV $k \in K^a$ to effectively traverse arc $(i, j) \in A$, set by the logistics operator and derived by AI elaboration. No analogous arc-reliability parameter is introduced for HDVs. The notation is kept vehicle-specific for modeling generality, although, in the case study of Section 6, a homogeneous AV setting is adopted, so that identical reliability values are assumed across all AVs on the same arc.
- Γ^a_k: Initial level of charge of vehicle $k \in K^a$ at the start of the planning horizon.
- Γ^{he}_k: Initial level of charge of vehicle $k \in K^{he}$ at the start of the planning horizon.
- π^a_k: Initial depot in which vehicle $k \in K^a$ is located at the start of the planning horizon.
- π^h_k: Initial depot in which vehicle $k \in K^h$ is located at the start of the planning horizon.
- M^a_k: Maximum charge for vehicle $k \in K^a$.
- M^{he}_k: Maximum charge for vehicle $k \in K^{he}$.
- m^a_k: Fixed recharge increment for vehicle $k \in K^a$ when stopping at a parking node $p \in P$.
- m^{he}_k: Fixed recharge increment for vehicle $k \in K^{he}$ when stopping at a parking node $p \in P$.
- $\Lambda^{\min}$: Minimum allowed arc reliability for AVs.
- R^a_k: Energy consumption for a unit of distance traveled by vehicle $k \in K^a$.
- R^{he}_k: Energy consumption for a unit of distance traveled by vehicle $k \in K^{he}$.
- τ^a_k: Time to recharge vehicle $k \in K^a$ by an amount $m^a_k \le M^a_k$. In our model, we assume that the state-of-charge (SoC) of a vehicle $k \in K^a$ increases by a fixed amount $m^a_k \le M^a_k$ when a recharge is operated at a parking area.
- τ^{he}_k: Time to recharge vehicle $k \in K^{he}$ by an amount $m^{he}_k \le M^{he}_k$. Similarly to vehicles $k \in K^a$, we assume that the SoC of a vehicle $k \in K^{he}$ increases by a fixed amount $m^{he}_k \le M^{he}_k$ when a recharge is operated at a parking area.
- w^a_k: Cost per unit of distance using vehicle $k \in K^a$.
- w^h_k: Cost per unit of distance using vehicle $k \in K^h$.
- ω^h_{ki}: 0-1 parameter indicating compliance between vehicle $k \in K^h$ and operations to be performed at node $i \in N$.
- ω^a_{ki}: 0-1 parameter indicating the compliance between vehicle $k \in K^a$ and operations to be performed at node $i \in N$.
- δ^h_{ijk}: 0-1 parameter indicating the compliance between vehicle $k \in K^h$ and arc $(i, j) \in A$.
- δ^a_{ijk}: 0-1 parameter indicating the compliance between vehicle $k \in K^a$ and arc $(i, j) \in A$.

Assumption on HDV Reliability As specified in the parameter description, human-driven vehicles are not associated with arc reliability. This mainly reflects the HDV role as the operational backup for corridors that fail the AV reliability threshold and the fact that reliability, in this paper, is used to model the capability of traversing arcs using an autonomous guide rather than the chance of overcoming factors like human errors of fatigue.

Decision Variables

- $h^k_{ij} \in \{0, 1\}$: Binary variable, 1 if vehicle $k \in K^h$ uses arc $(i, j) \in A$, 0 otherwise.
- $a^k_{ij} \in \{0, 1\}$: Binary variable, 1 if vehicle $k \in K^a$ uses arc $(i, j) \in A$, 0 otherwise.
- $s^a_{pk} \in \{0, 1\}$: Holds 1 if vehicle $k \in K^a$ stops at parking location $p \in P$ to recharge.
- $s^{he}_{pk} \in \{0, 1\}$: Holds 1 if vehicle $k \in K^{he}$ stops at parking location $p \in P$ to recharge.

- $e^a_{ijk} \geq 0$: The energy consumed by vehicle $k \in K^a$ on arc $(i, j) \in A$; is zero if the arc is not used.
- $e^{he}_{ijk} \geq 0$: The energy consumed by vehicle $k \in K^{he}$ on arc $(i, j) \in A$; is zero if the arc is not used.

- $\eta^a_{jk} \geq 0$: The remaining energy level (SoC) of vehicle $k \in K^a$ at node $j \in N$.
- $\eta^{he}_{jk} \geq 0$: The remaining energy level of vehicle $k \in K^{he}$ at node $j \in N$.

- $q^a_{ijk} \geq 0$: The amount of product transported by vehicle $k \in K^a$ on arc $(i, j) \in A$; is zero if the arc is not used.
- $q^h_{ijk} \geq 0$: The amount of product transported by vehicle $k \in K^h$ on arc $(i, j) \in A$; is zero if the arc is not used.

- $\Delta^a_{sk} \geq 0$: Amount of product loaded at supply node $s \in S$ by vehicle $k \in K^a$.
- $\Delta^h_{sk} \geq 0$: Amount of product loaded at supply node $s \in S$ by vehicle $k \in K^h$.
- $\Theta^a_{ck} \geq 0$: Amount of product unloaded at customer node $c \in C$ by vehicle $k \in K^a$.
- $\Theta^h_{ck} \geq 0$: Amount of product unloaded at customer node $c \in C$ by vehicle $k \in K^h$.

Objective Function The logistics operator aims to minimize total emissions, total energy consumption, total cost, and maximize total reliability of AVs, as follows:

$$\begin{aligned}\min\ & \alpha_1\left(\sum_{k\in K^a}\sum_{(i,j)\in A} C^a_k d_{ij} a^k_{ij} + \sum_{k\in K^h}\sum_{(i,j)\in A} C^h_k d_{ij} h^k_{ij}\right) + \\ & \alpha_2\left(\sum_{k\in K^a}\sum_{(i,j)\in A} e^a_{ijk} + \sum_{k\in K^{he}}\sum_{(i,j)\in A} e^{he}_{ijk}\right) + \\ & \alpha_3\left(\sum_{k\in K^a}\sum_{(i,j)\in A} w^a_k d_{ij} a^k_{ij} + \sum_{k\in K^h}\sum_{(i,j)\in A} w^h_k d_{ij} h^k_{ij}\right) - \\ & \alpha_4\left(\sum_{k\in K^a}\sum_{(i,j)\in A} \lambda^k_{ij} a^k_{ij}\right).\end{aligned} \tag{3}$$

where α_ℓ, with $\ell = 1, \ldots, 4$ are positive scalars.

Constraints The following constraints define the operational rules that govern the system. Each constraint ensures feasibility within the transportation network and enforces operational, logistics, and technical limitations. In Table 2, after the model, we summarize all of them.

Depot departure constraints

$$\sum_{j\in N} h^k_{\pi^h_k, j} \leq 1, \quad \forall k \in K^h, \tag{4}$$

$$\sum_{j\in N} a^k_{\pi^a_k, j} \leq 1, \quad \forall k \in K^a. \tag{5}$$

These constraints ensure that each vehicle (human-driven or automated) can start at most one route from its designated depot (position).

Since vehicles are explicitly indexed and each vehicle can depart at most once from its initial depot, the total number of activated routes cannot exceed the available fleet. Constraints 6–7

$$\sum_{k\in K^a}\sum_{j\in N} a^k_{\pi^a_k j} \ \leq\ |K^a|, \tag{6}$$

$$\sum_{k\in K^h}\sum_{j\in N} h^k_{\pi^h_k j} \ \leq\ |K^h|. \tag{7}$$

are implied by the per-vehicle departure constraints and are included only to make the fleet upper bound explicit.

Depot restriction constraints

$$\sum_{j\in N} h^k_{dj} = 0, \quad \forall k \in K^h, \forall d \in D : d \neq \pi^h_k, \tag{8}$$

$$\sum_{j\in N} a^k_{dj} = 0, \quad \forall k \in K^a, \forall d \in D : d \neq \pi^a_k. \tag{9}$$

These restrictions prevent vehicles (human-driven or automated) from departing from depots where they were not initially stationed, maintaining operational integrity.

$$\sum_{(i,j)\in A} a^k_{ij} \ \leq\ |A| \sum_{j\in N} a^k_{\pi^a_k, j}, \qquad \forall k \in K^a, \tag{10}$$

$$\sum_{(i,j)\in A} h^k_{ij} \ \leq\ |A| \sum_{j\in N} h^k_{\pi^h_k, j}, \qquad \forall k \in K^h. \tag{11}$$

The above constraints define the route activation constraints (no depot-free cycles).

Flow conservation constraints

$$\sum_{j\in N} h^k_{ij} - \sum_{j\in N} h^k_{ji} = 0, \quad \forall k \in K^h, \forall i \in P \cup S \cup C, \tag{12}$$

$$\sum_{j\in N} a^k_{ij} - \sum_{j\in N} a^k_{ji} = 0, \quad \forall k \in K^a, \forall i \in P \cup S \cup C, \tag{13}$$

$$\sum_{j\in N} a^k_{ij} \leq 1, \quad \sum_{j\in N} a^k_{ji} \leq 1 \qquad \forall k \in K^a,\ \forall i \in P \cup S \cup C, \tag{14}$$

$$\sum_{j\in N} h^k_{ij} \leq 1, \quad \sum_{j\in N} h^k_{ji} \leq 1 \qquad \forall k \in K^h,\ \forall i \in P \cup S \cup C. \tag{15}$$

Table 2 Synthesis of the constraints

Constraint family	Description
Depot departure (AV/HDV)	Each vehicle can start *at most one* route from its own depot.
Depot restriction (AV/HDV)	No vehicle may depart from a depot other than its assigned one.
Flow conservation (AV/HDV)	If a vehicle enters a non-depot node, it must also leave it (route continuity).
Depot capacity	The number of vehicles stationed/parked at depot d never exceeds capacity V_d.
Energy consumption on arcs	Arc energy is proportional to the travelled distance; zero if the arc is not used.
SoC propagation on non-parking arcs	State-of-charge decreases along traversed arcs due to the consumed energy.
Recharge at parking	At parking/charging nodes, SoC increases by a recharge increment $m_k^{\cdot}$ when $s_{pk}^{\cdot} = 1$.
Initial SoC from depot	Departure from depot sets initial SoC $\Gamma_k^{\cdot}$.
Zero load at depots (AV/HDV)	No load leaves or enters depots (vehicle load at depots is zero).
Load conservation at parking	Through a parking node, incoming load equals outgoing load.
Loading at supplies	Vehicles pick up $\Delta_{sk}^{\cdot}$ units at supply nodes s.
Unloading at customers	At customers c, vehicles unload $\Theta_{ck}^{\cdot}$ units (deliveries).
Arc load capacity (AV/HDV)	The load on any used arc never exceeds the vehicle capacity.
Customer service & demand	Each customer is visited by ≥ 1 vehicle and its demand ρ_c is met.
Supply capacity	Total loaded quantity at supply s does not exceed availability θ_s.
Route & node compatibility	Only allowed arcs/nodes per vehicle type can be selected.
Reliability gate (AV)	AV can use an arc only if its reliability exceeds the threshold $\Lambda^{\min}$.
Tour time limits (AV/HDV)	Travel plus charging time must not exceed the route horizon $T_{\max}$.
Parking capacity	Number of vehicles parked at node p cannot exceed capacity V_p.

These restrictions ensure that for every parking, supply, and customer node that a vehicle enters, it must also leave, following exactly one route. This guarantees continuity and prevents inconsistencies in the route.

Depot capacity constraints

$$\sum_{k\in K^h:\,\pi_k^h=d}\Big(1-\sum_{j\in N}h_{dj}^k\Big)+\sum_{k\in K^a:\,\pi_k^a=d}\Big(1-\sum_{j\in N}a_{dj}^k\Big)+\sum_{k\in K^h}\sum_{i\in N}h_{id}^k+\sum_{k\in K^a}\sum_{i\in N}a_{id}^k\ \leq\ V_d,\qquad \forall d\in D. \tag{16}$$

This constraint ensures that the number of vehicles stationed at any depot does not exceed the maximum available capacity of the depot.

Energy consumption constraints

$$e_{ijk}^a = R_k^a d_{ij} a_{ij}^k,\quad \forall k\in K^a,\quad \forall (i,j)\in A, \tag{17}$$

$$e_{ijk}^{he} = R_k^{he} d_{ij} h_{ij}^k,\quad \forall k\in K^{he},\quad \forall (i,j)\in A. \tag{18}$$

The energy consumed by an electric vehicle on a generic arc of a network must be zero if that arc is not used by that vehicle and is proportional to the distance traveled along the arc.

$$\eta_{\pi_k^a,k}^a = \Gamma_k^a,\qquad \forall k\in K^a, \tag{19}$$

$$\eta_{\pi_k^h,k}^{he} = \Gamma_k^{he},\qquad \forall k\in K^{he}. \tag{20}$$

Inequalities 19-20 define the initial SoC at the starting depot for vehicles in K^a and K^{he}.

$$\eta_{jk}^a \leq \eta_{ik}^a - e_{ijk}^a + M_k^a(1-a_{ij}^k),\quad \forall k\in K^a,\ \forall (i,j)\in A:\ j\notin P, \tag{21}$$

$$\eta_{jk}^a \geq \eta_{ik}^a - e_{ijk}^a - M_k^a(1-a_{ij}^k),\quad \forall k\in K^a,\ \forall (i,j)\in A:\ j\notin P, \tag{22}$$

$$\eta_{jk}^{he} \leq \eta_{ik}^{he} - e_{ijk}^{he} + M_k^{he}(1-h_{ij}^k),\quad \forall k\in K^{he},\ \forall (i,j)\in A:\ j\notin P, \tag{23}$$

$$\eta_{jk}^{he} \geq \eta_{ik}^{he} - e_{ijk}^{he} - M_k^{he}(1-h_{ij}^k),\quad \forall k\in K^{he},\ \forall (i,j)\in A:\ j\notin P. \tag{24}$$

Inequalities 21-22 and 23-24 model SoC propagation on non-parking arcs.

$$\eta_{pk}^a \leq \eta_{ik}^a - e_{ipk}^a + m_k^a s_{pk}^a + M_k^a(1-a_{ip}^k),\quad \forall k\in K^a,\ \forall (i,p)\in A: p\in P, \tag{25}$$

$$\eta_{pk}^a \geq \eta_{ik}^a - e_{ipk}^a + m_k^a s_{pk}^a - M_k^a(1-a_{ip}^k),\quad \forall k\in K^a,\ \forall (i,p)\in A: p\in P, \tag{26}$$

$$s_{pk}^a \leq \sum_{i\in N:(i,p)\in A} a_{ip}^k,\quad \forall p\in P,\ \forall k\in K^a, \tag{27}$$

$$\eta_{pk}^{he} \leq \eta_{ik}^{he} - e_{ipk}^{he} + m_k^{he} s_{pk}^{he} + M_k^{he}(1-h_{ip}^k),\quad \forall k\in K^{he},\ \forall (i,p)\in A: p\in P, \tag{28}$$

$$\eta_{pk}^{he} \geq \eta_{ik}^{he} - e_{ipk}^{he} + m_k^{he} s_{pk}^{he} - M_k^{he}(1-h_{ip}^k),\quad \forall k\in K^{he},\ \forall (i,p)\in A: p\in P, \tag{29}$$

$$s_{pk}^{he} \leq \sum_{i\in N:(i,p)\in A} h_{ip}^k,\quad \forall p\in P,\ \forall k\in K^{he}. \tag{30}$$

In a parking node, the electric vehicle can be recharged with a consequent increase in energy by $m_k^a \leq M_k^a$ or $m_k^{he} \leq M_k^{he}$, depending on the type of vehicle. Note that, due to the domains of Constraints 63 and 64, η_{pk}^a and η_{pk}^{he} cannot exceed M_k^a and M_k^{he}, respectively.

Vehicle load constraints

$$\sum_{j\in N:(d,j)\in A} q^a_{djk} = 0, \quad \forall k \in K^a, \quad \forall d \in D, \tag{31}$$

$$\sum_{j\in N:(d,j)\in A} q^h_{djk} = 0, \quad \forall k \in K^h, \quad \forall d \in D, \tag{32}$$

$$\sum_{j\in N:(j,d)\in A} q^a_{jdk} = 0, \quad \forall k \in K^a, \quad \forall d \in D, \tag{33}$$

$$\sum_{j\in N:(j,d)\in A} q^h_{jdk} = 0, \quad \forall k \in K^h, \quad \forall d \in D. \tag{34}$$

These constraints impose that each vehicle load in a depot must be zero. In particular, Eqs. 31 and 32 refer to the quantities existing in the depots, and Eqs. 33 and 34 those entering the depots.

$$\sum_{i\in N:(i,p)\in A} q^a_{ipk} = \sum_{j\in N:(p,j)\in A} q^a_{pjk}, \quad \forall k \in K^a, \ \forall p \in P, \tag{35}$$

$$\sum_{i\in N:(i,p)\in A} q^h_{ipk} = \sum_{j\in N:(p,j)\in A} q^h_{pjk}, \quad \forall k \in K^h, \ \forall p \in P. \tag{36}$$

These constraints ensure that the total amount of load transported remains conserved when flowing through a parking node.

$$\sum_{i\in N:(s,i)\in A} q^a_{sik} = \sum_{i\in N:(i,s)\in A} q^a_{isk} + \Delta^a_{sk}, \quad \forall k \in K^a, \forall s \in S, \tag{37}$$

$$\sum_{i\in N:(s,i)\in A} q^h_{sik} = \sum_{i\in N:(i,s)\in A} q^h_{isk} + \Delta^h_{sk}, \quad \forall k \in K^h, \forall s \in S. \tag{38}$$

These constraints ensure that vehicles are properly loaded at supply nodes before departing.

$$\sum_{i\in N:(c,i)\in A} q^a_{cik} = \sum_{i\in N:(i,c)\in A} q^a_{ick} - \Theta^a_{ck}, \quad \forall k \in K^a, \quad \forall c \in C, \tag{39}$$

$$\sum_{i\in N:(c,i)\in A} q^h_{cik} = \sum_{i\in N:(i,c)\in A} q^h_{ick} - \Theta^h_{ck}, \quad \forall k \in K^h, \quad \forall c \in C. \tag{40}$$

In Eqs. 39 and 40, we impose that visiting a client may require the unloading of a certain amount Θ^h_{ck} of product to fully or partially satisfy the demand of the client.

$$q^a_{ijk} \le a^k_{ij} Q^a_k, \quad \forall k \in K^a, \quad \forall (i,j) \in A, \tag{41}$$

$$q^h_{ijk} \le h^k_{ij} Q^h_k, \quad \forall k \in K^h, \quad \forall (i,j) \in A. \tag{42}$$

Constraints 41 and 42 say that the load of a vehicle cannot exceed its maximum capacity on a generic arc of the network.

Customer demand constraints

$$\sum_{k\in K^h} \sum_{i\in N:(i,c)\in A} h^k_{ic} + \sum_{k\in K^a} \sum_{i\in N:(i,c)\in A} a^k_{ic} \ge 1, \quad \forall c \in C, \tag{43}$$

$$\sum_{k\in K^h} \Theta^h_{ck} + \sum_{k\in K^a} \Theta^a_{ck} = \rho_c, \quad \forall c \in C. \tag{44}$$

Inequality 43 says that each customer must be visited by at least one vehicle, and Inequality 44 guarantees that the overall fleet meets the customer's demand.

Supply capacity constraints

$$\sum_{k\in K^a} \Delta^a_{sk} + \sum_{k\in K^h} \Delta^h_{sk} \le \theta_s, \quad \forall s \in S. \tag{45}$$

These constraints impose that the overall quantity loaded by vehicles in a supply node does not exceed the product availability of that supply node.

Route and node compatibility constraints

$$a^k_{ij} \le \delta^a_{ijk}, \quad \forall (i,j) \in A, \forall k \in K^a, \tag{46}$$

$$h^k_{ij} \le \delta^h_{ijk}, \quad \forall (i,j) \in A, \forall k \in K^h, \tag{47}$$

$$a^k_{ij} \le \omega^a_{ki}, \quad \forall (i,j) \in A, \forall i \in N, \forall k \in K^a, \tag{48}$$

$$h^k_{ij} \le \omega^h_{ki}, \quad \forall (i,j) \in A, \forall i \in N, \forall k \in K^h, \tag{49}$$

$$a^k_{ij} \le \omega^a_{kj}, \quad \forall (i,j) \in A, \forall j \in N, \forall k \in K^a, \tag{50}$$

$$h^k_{ij} \le \omega^h_{kj}, \quad \forall (i,j) \in A, \forall j \in N, \forall k \in K^h. \tag{51}$$

These restrictions ensure that vehicles travel only on allowed roads and nodes according to infrastructure compatibility.

Reliability constraints

$$\lambda^k_{ij} \ge \Lambda^{\min} a^k_{ij}, \quad \forall (i,j) \in A, \forall k \in K^a. \tag{52}$$

Inequality 52 ensures that automated vehicles are employed on arc $(i,j) \in A$ only if their reliability satisfies an assigned threshold $\Lambda^{\min}$. These constraints are for AVs only.

Time constraints

$$T_{\max} \ge \sum_{(i,j)\in A} T^a_{ijk} a^k_{ij} + \sum_{p\in P} \tau^a_k s^a_{pk}, \quad \forall k \in K^a, \tag{53}$$

$$T_{\max} \geq \sum_{(i,j)\in A} T^h_{ijk} h^k_{ij} + \sum_{p\in P} \tau^{he}_k s^{he}_{pk}, \quad \forall k \in K^{he}, \tag{54}$$

$$T_{\max} \geq \sum_{(i,j)\in A} T^h_{ijk} h^k_{ij}, \quad \forall k \in K^h \setminus K^{he}. \tag{55}$$

Inequalities 53, 54, and 55 impose that the travel and charging time should not exceed $T_{\max}$, that is, the maximum duration of the tour.

Parking capacity constraints

$$\sum_{k\in K^a} s^a_{pk} + \sum_{k\in K^{he}} s^{he}_{pk} \leq V_p, \quad \forall p \in P. \tag{56}$$

This restriction enforces parking limits, ensuring that the number of vehicles parked does not exceed the available space.

The model uses an explicit vehicle index k for every routing and parking/recharge decision. Hence, any vehicle counted as "parked" at a parking node is necessarily one of the vehicles in the given fleet (there is no mechanism in the formulation that can create additional vehicles). Therefore, a global constraint limiting the total number of parked vehicles by the total fleet size would be redundant. In particular, the parking-capacity constraints are meant to limit *space* at each parking location; the total number of vehicles involved in the plan is already limited by the fact that routes and stops are defined per vehicle, starting from the vehicles initially located at their assigned depots.

Domains of the variables

$$h^k_{ij} \in \{0,1\}, \quad \forall k \in K^h, \quad \forall (i,j) \in A, \tag{57}$$

$$a^k_{ij} \in \{0,1\}, \quad \forall k \in K^a, \quad \forall (i,j) \in A, \tag{58}$$

$$s^a_{pk} \in \{0,1\}, \quad \forall k \in K^a, \quad \forall p \in P, \tag{59}$$

$$s^{he}_{pk} \in \{0,1\}, \quad \forall k \in K^{he}, \quad \forall p \in P, \tag{60}$$

$$e^a_{ijk} \geq 0, \quad \forall k \in K^a, \quad \forall (i,j) \in A, \tag{61}$$

$$e^{he}_{ijk} \geq 0, \quad \forall k \in K^{he}, \quad \forall (i,j) \in A, \tag{62}$$

$$0 \leq \eta^a_{jk} \leq M^a_k, \quad \forall k \in K^a, \quad \forall j \in N, \tag{63}$$

$$0 \leq \eta^{he}_{jk} \leq M^{he}_k, \quad \forall k \in K^{he}, \quad \forall j \in N, \tag{64}$$

$$q^a_{ijk} \geq 0, \quad \forall k \in K^a, \quad \forall (i,j) \in A, \tag{65}$$

$$q^h_{ijk} \geq 0, \quad \forall k \in K^h, \quad \forall (i,j) \in A, \tag{66}$$

$$\Delta^a_{sk} \geq 0, \quad \forall k \in K^a, \quad \forall s \in S, \tag{67}$$

$$\Delta^h_{sk} \geq 0, \quad \forall k \in K^h, \quad \forall s \in S, \tag{68}$$

$$\Theta^a_{ck} \geq 0, \quad \forall k \in K^a, \quad \forall c \in C, \tag{69}$$

$$\Theta^h_{ck} \geq 0, \quad \forall k \in K^h, \quad \forall c \in C. \tag{70}$$

3.3 Minimum Arc Reliability Definition

In the model, AV arcs are allowed only if their reliability exceeds a threshold $\Lambda^{\min}$ (Constraint 52). In practice, we set

$$\Lambda^{\min} = \beta^{1/L},$$

where $\beta \in (0,1)$ is a target route-level reliability chosen by the operator and L is a data-driven proxy for the number of arcs on an AV route that we generated by the proposed Algorithm 1. This choice equates the geometric mean of the arc reliabilities along a length-L route to β. Empirically, we can set $\beta \in [0.85, 0.95]$ and L taken as a high percentile (e.g., the 90th) of the arc-counts produced by Algorithm 1. This construction implicitly relies on an independence approximation across arcs when translating a route-level reliability target into an arc-level admissibility threshold. More precisely, it assumes that route reliability can be approximated by the product of marginal arc reliabilities, so that a geometric-mean threshold can be used as an operational proxy. In real transportation networks, however, arc reliabilities are likely to be positively correlated because they are jointly affected by spatially structured factors such as weather, congestion, and infrastructure quality. Under positive correlation, the product of marginal reliabilities may fail to reproduce the true route-level success probability exactly and should therefore not be interpreted as an exact probabilistic guarantee. For this reason, the rule $\Lambda^{\min} = \beta^{1/L}$ is used in this paper as a transparent engineering approximation that converts a route-level reliability target into an auditable arc-level admissibility threshold.

In practical terms, this means that $\Lambda^{\min}$ should be interpreted as a policy-oriented operating threshold rather than as a statistically exact safety certificate. If a decision-maker wishes to hedge against possible positive dependence across

arcs, a more conservative choice of β can be adopted, or route-level correlation-aware reliability models can be introduced in future extensions.

Algorithm 1 Computation of the route-length proxy L.

1: For all $(i, j) \in A$ and $k \in K^a$, set energy lower bounds $c_{ij}^k \leftarrow R_k^a \cdot d_{ij}$ with $k \in K^a$.
2: Assign each $i \in C$ to its nearest depot $d(i) \in D$ by distance $d_{d(i),i}$ (ties are broken arbitrarily).
3: Initialize a pool of singleton AV routes for each assigned customer that is round-trip energy-feasible for at least one AV:

$$\mathcal{R}_d \leftarrow \{ r = (d \to i \to d) : i \in C,\ d = d(i),\ \exists k \in K^a \text{ s.t. } c_{di}^k + c_{id}^k \le M_k^a \wedge \rho_i \le Q_k^a \}.$$

4: **for** each depot $d \in D$ **do**
5: Clarke and Wright [6] savings. For assigned $i \neq j$ with $d(i) = d(j) = d$, compute

$$s(i, j) = d_{id} + d_{dj} - d_{ij}, \qquad s(j, i) = d_{jd} + d_{di} - d_{ji},$$

and create two candidate pairs $(i \to j)$ and $(j \to i)$ accordingly.
6: Sort all directed pairs by nonincreasing savings.
7: **for** each directed pair $(i \to j)$ in that order **do**
8: Let r_i be a route in $\mathcal{R}_d$ that currently *ends* at i, and r_j a route that currently *starts* at j (if such routes exist and are distinct).
9: Capacity check. Check $\exists k \in K^a$ with $q(r_i) + q(r_j) \le Q_k^a$, where $q(\cdot)$ is the sum of customer demands on the route.
10: Boundary-leg energy check. Check $\exists k \in K^a$ such that all boundary legs to be created remain feasible on some $k \in K^a$:

$$c_{\text{last}(r_i),\, j}^k \le M_k^a, \quad c_{i,\, \text{first}(r_j)}^k \le M_k^a, \quad c_{d,\, \text{first}(r_i)}^k \le M_k^a, \quad c_{\text{last}(r_j),\, d}^k \le M_k^a.$$

11: **if** both checks pass **then**
12: Merge $r_i \leftarrow r_i \circ r_j$ (concatenate) and remove r_j from $\mathcal{R}_d$.
13: **end if**
14: **end for**
15: **end for**
16: Let $\mathcal{R} \leftarrow \bigcup_{d \in D} \mathcal{R}_d$.
17: **if** $\mathcal{R} = \varnothing$ **then**
18: Set $L \leftarrow 2$ and go to Step 22.
19: **end if**
20: For each $r \in \mathcal{R}$, define its length as the number of arcs $H(r)$ along r (including depot legs).
21: Compute the empirical τ-quantile (e.g., $\tau = 0.9$) of $\{H(r) : r \in \mathcal{R}\}$ with the nearest-rank rule: if $n = |\mathcal{R}|$ and $H_{(1)} \le \cdots \le H_{(n)}$ are the sorted values, then $L \leftarrow \lceil H_{(\lceil n\tau \rceil)} \rceil$.
22: **return** L and set $\Lambda^{\min} \leftarrow \beta^{1/L}$.

The feasible set of our model is monotone nonincreasing in $\Lambda^{\min}$: increasing $\Lambda^{\min}$ can only remove AV arcs, never add them. The full model remains feasible provided that every required customer is still connected to some depot by a feasible path in the union of (i) all HDV arcs and (ii) AV arcs with $\lambda_{ij} \ge \Lambda^{\min}$ (and subject to time/energy/compatibility constraints). Therefore, infeasibility may arise only if a customer is effectively AV-only (by compatibility) or HDV routing is otherwise blocked (e.g., by time limits), and all incident AV (i, j) arcs to/from that customer have $\lambda_{ij} < \Lambda^{\min}$.

Before solving, we run a connectivity/feasibility check on $G(\Lambda^{\min})$: for each customer c, we verify (by the same Algorithm 1) the existence of at least one depot-reachable path that respects (i) vehicle-type compatibility and (ii) energy/time lower bounds (i.e., $t_{ij}^{\min}$ and c_{ij}^k used in Algorithm 1). If any customer fails this test, either lower $\Lambda^{\min}$ or allow the customer to be served by HDVs. In practice, a safe cap Λ^{cap} is

$$\Lambda^{\min} \ \le \ \min_{c} \ \max_{\text{feasible path } r:\ \text{depot} \to c} \Big(\prod_{(i,j) \in r} \lambda_{ij} \Big)^{1/|r|}, \tag{71}$$

i.e., do not set $\Lambda^{\min}$ above the best geometric-mean reliability of any feasible depot–customer route. This cap should be interpreted consistently with the independence approximation discussed above: it provides an operational upper bound for threshold selection within the model, but it is not intended as an exact route-level probabilistic guarantee when arc reliabilities are correlated.

4 Computing Arc Reliability Ranges to Replan Routes When Reliability Changes Over Time

In this and the next section, we analyze how the model can be used under changing network conditions that imply variations in the λ_{ij}^k values over time. We start by analyzing the theoretical framework.

Consider a general constrained optimization problem where the objective function and some constraints depend on a parameter ζ:

$$\min_{x} \quad f(x, \zeta) \tag{72}$$

$$\text{s.t.} \quad g_i(x, \zeta) \le 0, \quad i \in I_\zeta, \tag{73}$$

$$g_i(x) \le 0, \quad i \in I. \tag{74}$$

where:

- $f(x, \zeta)$ is the objective function dependent on ζ,
- $g_i(x, \zeta)$ are inequality constraints that explicitly depend on ζ,
- $g_i(x)$ are inequality constraints independent of ζ,
- I_ζ is the set of indexes of the constraints parametrized in ζ
- I is the set of indexes of the constraints not parametrized in ζ
- x represents the decision variables.

The Lagrangian function associated with this problem is given by:

$$\mathcal{L}(x, \zeta, \phi, \mu) = f(x, \zeta) + \sum_{i \in I_\zeta} \phi_i g_i(x, \zeta) + \sum_{i \in I} \mu_i g_i(x), \tag{75}$$

where $\phi_i \geq 0$ are the dual variables for the inequality constraints.

The first-order necessary conditions (Karush-Kuhn-Tucker conditions) include the following:

$$\nabla_x \mathcal{L} = \nabla f(x,\zeta) + \sum_{i \in I_\zeta} \phi_i \nabla g_i(x,\zeta) + \sum_{i \in I} \mu_i \nabla g_i(x) = 0, \tag{76}$$

$$g_i(x,\zeta) \leq 0, \quad \forall i \in I_\zeta, \tag{77}$$

$$g_i(x) \leq 0, \quad \forall i \in I, \tag{78}$$

$$\phi_i g_i(x,\zeta) = 0, \quad \forall i \in I_\zeta, \quad \text{(Complementary Slackness)}, \tag{79}$$

$$\phi_i g_i(x) = 0, \quad \forall i \in I, \quad \text{(Complementary Slackness)}, \tag{80}$$

$$\phi_i \geq 0, \quad \forall i \in I_\zeta \cup I. \tag{81}$$

To measure how changes in ζ affect the optimal objective function Z^*, we differentiate the Lagrangian with respect to ζ:

$$\frac{d\mathcal{L}}{d\zeta} = \frac{\partial f(x,\zeta)}{\partial \zeta} + \sum_{i \in I_\zeta} \phi_i \frac{\partial g_i(x,\zeta)}{\partial \zeta}. \tag{82}$$

Thus, the sensitivity of the optimal objective function Z^* to ζ is:

$$\frac{\partial Z^*}{\partial \zeta} = \frac{\partial f(x^*,\zeta)}{\partial \zeta} + \sum_{i \in I_\zeta} \phi_i^* \frac{\partial g_i(x^*,\zeta)}{\partial \zeta}. \tag{83}$$

This expression tells us how much the optimal cost Z^* changes with respect to small variations in ζ. Notice that constraints associated with sets I do not contribute to this sensitivity because they do not depend on ζ, which makes their derivatives zero.

Let us now customize the above analysis to evaluate the stability of the optimal solution of our model with respect to changes in λ_{ij}^k.

The routing formulation in Section 3 is a mixed-integer program, and dual multipliers and KKT-based sensitivity analysis are not defined in general. To retain a dual-based signal, we compute shadow prices on an *LP surrogate*, namely the continuous relaxation obtained by relaxing all binary variables in the proposed model. The resulting dual multipliers are well-defined and quantify first-order sensitivity of the optimal LP value (a valid lower bound for the MIP) with respect to small parameter changes. We use these values as a criterion for re-optimization under time-varying reliabilities (they do not represent, in general, a certificate that the MIP optimum is unchanged).

Assume that parameter ζ considered in the previous theoretical framework, in our model, is coded by λ_{ij}^k values and, therefore, constraints $g_i(x,\zeta)$ previously defined are the reliability gate constraints

$$\lambda_{ij}^k \geq \Lambda^{\min} a_{ij}^k, \qquad \forall (i,j) \in A, \ \forall k \in K^a, \tag{84}$$

which we rewrite as

$$g_{ij}^k(x,\lambda) := \Lambda^{\min} a_{ij}^k - \lambda_{ij}^k \leq 0.$$

Let $\phi_{ij}^{k,\mathrm{LP}*} \geq 0$ denote the dual multiplier of $g_{ij}^k \leq 0$ in the LP relaxation optimum. Since $\partial g_{ij}^k / \partial \lambda_{ij}^k = -1$, the constraint contribution in Eq. 82 is $-\phi_{ij}^{k,\mathrm{LP}*}$.

Since the objective includes the reliability term $^{-\sum_{k,(i,j)} \lambda_{ij}^k a_{ij}^k}$, then $\partial f / \partial \lambda_{ij}^k = -a_{ij}^{k,\mathrm{LP}*}$ (up to the fixed scalarization weight α_4 of that term), and the total derivative becomes:

$$\frac{\partial Z^{\mathrm{LP}*}}{\partial \lambda_{ij}^k} = -\alpha_4 \cdot a_{ij}^{k,\mathrm{LP}*} - \phi_{ij}^{k,\mathrm{LP}*}. \tag{85}$$

Let $\epsilon > 0$ be a tolerance on the admissible local change in the LP objective value (or LP bound) between two decision points. Using the first-order approximation $Z^{\mathrm{LP}*}(\lambda + \Delta\lambda) \approx Z^{\mathrm{LP}*}(\lambda) + \left(\partial Z^{\mathrm{LP}*} / \partial \lambda_{ij}^k\right) \Delta\lambda_{ij}^k$, we define for each parameter λ_{ij}^k the screening interval

$$\begin{aligned} \lambda_{ij}^{k,\min} &= \max\left\{0,\ \lambda_{ij}^k - \frac{\epsilon}{\left|\frac{\partial Z^{\mathrm{LP}*}}{\partial \lambda_{ij}^k}\right| + \varepsilon_0}\right\}, \\ \lambda_{ij}^{k,\max} &= \min\left\{1,\ \lambda_{ij}^k + \frac{\epsilon}{\left|\frac{\partial Z^{\mathrm{LP}*}}{\partial \lambda_{ij}^k}\right| + \varepsilon_0}\right\}, \end{aligned} \tag{86}$$

where $\varepsilon_0 > 0$ is a small safeguard to avoid division by zero. These intervals are used as a *trigger* as described in the next section: if many updated λ_{ij}^k leave their intervals, we warm-start or re-solve the MIP; otherwise, we keep the current plan.

Independent of the screening rule, if the current MIP plan uses an AV arc ($a_{ij}^{k,\mathrm{MIP}*} = 1$) and an update yields $\lambda_{ij}^k < \Lambda^{\min}$, then the incumbent becomes infeasible, and rerouting is mandatory.

To extract the optimal dual variables ϕ_{ij}^{*k} of Constraints 52 after the solver runs (to solve the LP relaxation of the proposed MIP) using the Gurobi 13.0.1 solver, it can be done as follows:

```
rel = model.relax()
rel.optimize()
dual_values = rel.getAttr(GRB.Attr.Pi,
Reliability_Constraints)
```

Role and Selection of ϵ Parameter ϵ represents a predefined sensitivity threshold that determines how much variation in λ_{ij}^k is considered acceptable before requiring a model re-optimization. The choice of ϵ directly affects the computed stability range and is crucial for practical implementation. A higher ϵ allows for a broader range where changes in λ do not trigger re-optimization, while a lower ϵ results in a more conservative estimation, requiring frequent updates to the optimization model.

The selection of ϵ depends on operational flexibility, that is, if the system can tolerate small deviations in routing reliability, a higher ϵ is chosen, and computational constraints, that is, if frequent re-optimizations are computationally expensive, a higher ϵ is preferred.

5 An Optimization Framework for Dynamic Reliability Update and Rerouting

In the following, we propose an optimization framework that allows a logistic operator to dynamically take into account variations in the reliability parameters λ_{ij}^k in the network.

The procedure ensures that fleet operations remain optimal over time, minimizing the need for unnecessary recomputation using the stability ranges presented in the previous section while ensuring feasibility in response to significant changes in λ_{ij}^k.

Step 1: Initial model execution

- The logistic operator sets the initial set of reliability parameters λ_{ij}^k for each $k \in K^a$ and each arc $(i, j) \in A$ (e.g., relying on an AI-based prediction).

- The operator solves the optimization model presented in Section 3 with these values, obtaining the optimal routing and allocation of vehicles.

- The sensitivity analysis is performed to determine the stability range $[\lambda_{ij}^{k,\min}, \lambda_{ij}^{k,\max}]$ for each λ_{ij}^k, indicating that the solution is likely to remain valid within these limits.

Step 2: Periodic update of λ_{ij}^k

- Every Δt, reliability parameters λ_{ij}^k are updated based on new data and predictions.

- The operator checks whether all updated values remain within their respective stability intervals:

$$\lambda_{ij}^k \in [\lambda_{ij}^{k,\min}, \lambda_{ij}^{k,\max}] \quad \forall (i, j) \in A, \forall k \in K^a. \tag{87}$$

Step 3: Decision on recomputing the model

- If λ_{ij}^k remains within the stability range for all arcs and vehicles, no recomputation is triggered, and the current routing plan is kept.

- If any λ_{ij}^k falls outside its stability range, the model needs to be updated.

- To enhance computational efficiency, the logistic operator should determine whether the model can be partially updated: (i) if only a small subset of arcs/vehicles experiences significant changes, the previous solution can be used as a warm start, fixing variables for unchanged assignments and solving a reduced problem. (ii) If widespread changes occur, the model should be re-run from scratch to find a globally optimal new solution.

In Algorithm 2, we summarize the steps of the iterative procedure.

Algorithm 2 Dynamic reliability update and rerouting.

```
Initialize λ_ij^k.
Solve the optimization model with initial λ_ij^k and store optimal routes.
Compute stability intervals [λ_ij^{k,min}, λ_ij^{k,max}] as defined in Section 4.
while true do
    Wait for update interval Δt.
    Receive new λ_ij^k.
    if ∀(i, j), k : λ_ij^k ∈ [λ_ij^{k,min}, λ_ij^{k,max}] then
        Maintain current solution, no recomputation needed.
    else
        Identify affected arcs/vehicles where λ_ij^k is out of range.
        if only a small subset of arcs/vehicles are affected then
            Solve a reduced problem by fixing previous assignments where feasible.
            Recompute stability intervals [λ_ij^{k,min}, λ_ij^{k,max}] as defined in Section 4.
        else
            Re-run the full optimization model.
            Recompute stability intervals [λ_ij^{k,min}, λ_ij^{k,max}] as defined in Section 4.
        end if
        Communicate updated routing plan to AVs and HDVs.
    end if
end while
```

5.1 Step 6 of Algorithm 2: Estimating Reliability from Historical Traversal Data

The purpose of this subsection is to illustrate a possible data-driven pipeline for estimating the reliability parameters $\lambda_{ij}^k(t)$ in a real deployment. Accordingly, this part of the paper should be interpreted as a methodological sketch showing how arc reliabilities could be estimated from historical traversal data, rather than as an empirically calibrated predictive model validated on a specific dataset.

To estimate the reliability parameter $\lambda_{ij}^k(t)$, one may employ a supervised learning approach based on logistic regression. Such an approach models the probability that an autonomous vehicle (AV) $k \in K^a$ successfully traverses a given arc (i, j) under a specific set of conditions observed at time t.

In the illustrative setting considered here, the predictors are arc- and context-dependent, so the resulting reliability can be interpreted as an arc-level quantity common to all AVs operating under the same environmental conditions. A fully vehicle-specific reliability model would require the inclusion of AV-dependent predictors, such as sensor-suite configuration, software version, automation stack, or vehicle-specific performance indicators.

In a real deployment setting, each observation in the dataset would correspond to a historical traversal attempt of an arc (i, j) by an AV. A possible feature vector $\mathrm{x}_{ij}^{(t)}$ may include the following contextual variables:

- `travel_time` [min]: average observed travel time;
- `speed` [km/h]: average speed over the arc;
- `congestion` [0–1]: normalized traffic congestion level;
- `incidents`: number of recent override or intervention events;
- `infrastructure` [0–1]: infrastructure quality rating;
- `V2X`$\in \{0,1\}$: presence of vehicle-to-infrastructure communication;
- `weather`: a categorical variable with values in {*clear, rain, fog, snow*}.

It should be noted that some of the above predictors may be correlated in practice. In particular, `travel_time` and `speed` are mechanically related through distance and may therefore exhibit substantial collinearity depending on the aggregation level of the observations. In a real empirical implementation, this issue should be handled through standard preprocessing and model-selection procedures, such as feature elimination, regularization, or other collinearity diagnostics. Since the present subsection is illustrative and not based on a fitted dataset, these diagnostics are not reported here.

Each observation would be labeled with a binary outcome $y^{(i)}$ representing whether the traversal was successful:

$$y^{(i)} = \begin{cases} 1 & \text{if the AV completes the arc without intervention or failure,} \\ 0 & \text{if the AV fails, is overridden, or does not complete traversal.} \end{cases}$$

This label would then be used as the target variable in the training process.

Before model fitting, categorical variables can be encoded using one-hot encoding. For example, the `weather` variable can be transformed into:

- `weather_rain`, `weather_fog`, `weather_snow`$\in \{0,1\}$,

with `weather_clear` used as the reference category and excluded to avoid exact multicollinearity in the design matrix.

Assume that, in a real deployment setting, a dataset of N observations is available:

$$\mathcal{D} = \left\{(\mathrm{x}^{(i)}, y^{(i)})\right\}_{i=1}^{N}.$$

A logistic regression model could then be trained to estimate the probability of successful traversal:

$$\lambda^{(i)} = \mathbb{P}(y^{(i)} = 1 \mid \mathrm{x}^{(i)}, \boldsymbol{\beta}) = \frac{1}{1 + \exp\left(-(\beta_0 + \sum_{j=1}^{p} \beta_j x_j^{(i)})\right)}. \tag{88}$$

In such a setting, the model parameters $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]$ would be estimated by maximizing the log-likelihood function:

$$\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left[y^{(i)} \log(\lambda^{(i)}) + (1 - y^{(i)}) \log(1 - \lambda^{(i)})\right]. \tag{89}$$

Such an estimation problem can be solved numerically using standard optimization routines, such as L-BFGS or equivalent gradient-based solvers. In a full empirical study, one would also report the usual model-diagnostic information, such as calibration measures, out-of-sample performance, and uncertainty estimates for the fitted coefficients. These elements are outside the scope of the present paper because the current subsection is intended only to illustrate the estimation pipeline.

Once calibrated on operational data, the model could be used to compute the reliability of any arc (i, j) for an AV k at time t:

$$\lambda_{ij}^{k}(t) = \frac{1}{1 + \exp\left(-(\beta_0 + \sum_{j=1}^{p} \beta_j x_j^{(t)})\right)}. \tag{90}$$

Here, $\mathrm{x}_{ij}^{(t)}$ would contain the current values of traffic, weather, infrastructure, and other contextual variables observed for arc (i, j) at time t.

It is important to stress again that the reliability values used in the computational experiments of Section 6 are exogenous scenario parameters adopted for optimization and benchmarking purposes; they are not outputs of the illustrative logistic-regression pipeline described in this subsection.

In Table 3, we summarize the conceptual estimation pipeline.

Table 3 Conceptual pipeline to estimate $\lambda_{ij}^{k}(t)$ from data

Step	Description
1	Collect historical AV traversal logs
2	Label each attempt as success ($y = 1$) or failure ($y = 0$)
3	Encode categorical features and preprocess inputs
4	Estimate β through logistic regression
5	Use the fitted model to predict $\lambda_{ij}^{k}(t)$ in real time

5.1.1 Illustrative Numerical Example

For illustration, suppose that a logistic model calibrated on operational data has produced the following coefficients (with one-hot encoding for weather and `clear` as the reference category):

Parameter	Value
Intercept β_0	−3.20
`travel_time` (min) β_{tt}	−0.15
`speed` (km/h) β_{spd}	+0.04
`congestion`$[0,1]\beta_{\text{cong}}$	−2.10
`incidents` (count) β_{inc}	−0.60
`infrastructure`$[0,1]\beta_{\text{inf}}$	+1.80
`V2X`$\in\{0,1\}\beta_{\text{v2x}}$	+0.90
`weather_rain`β_{rain}	−0.80
`weather_fog`β_{fog}	−1.10
`weather_snow`β_{snow}	−1.70

Consider arc (i, j) at a decision time with the following contextual conditions: travel time = 3.5 min, speed = 40 km/h, congestion = 0.55, incidents = 1, infrastructure = 0.8, V2X = 1, weather = fog. With one-hot encoding, this implies (weather_rain,weather_fog, weather_snow)= $(0, 1, 0)$.

The linear predictor is then

$$\eta = \beta_0 + \beta_{\text{tt}} \cdot 3.5 + \beta_{\text{spd}} \cdot 40 + \beta_{\text{cong}} \cdot 0.55 + \beta_{\text{inc}} \cdot 1 + \beta_{\text{inf}} \cdot 0.8 + \beta_{\text{v2x}} \cdot 1 + \beta_{\text{fog}} \cdot 1.$$

Substituting the illustrative coefficients yields

$$\eta = -3.20 - 0.525 + 1.600 - 1.155 - 0.600 + 1.440 + 0.900 - 1.100 = -2.640.$$

Therefore, the estimated reliability is

$$\lambda_{ij}^{k}(t) = \sigma(\eta) = \frac{1}{1 + e^{-\eta}} = \frac{1}{1 + e^{2.640}} \approx 0.0667, \tag{91}$$

that is, approximately 6.7% probability of successful AV traversal under fog, relatively high congestion, and one recent incident.

For contrast, under more favorable conditions travel_time = 3.0, speed = 45, congestion = 0.20, incidents = 0, infrastructure = 0.9, V2X = 1, weather = clear, the same illustrative model gives

$$\eta' = -3.20 + (-0.15) \cdot 3.0 + 0.04 \cdot 45 - 2.10 \cdot 0.20 + 1.80 \cdot 0.9 + 0.90 = 0.25,$$

$$\lambda'^{k}_{ij}(t) = \sigma(\eta') = \frac{1}{1 + e^{-0.25}} \approx 0.562. \tag{92}$$

In this second situation, the predicted reliability is approximately 56.2%. The same type of computation would apply to any arc (i, j) and any $k \in K^a$, once the corresponding feature vector is instantiated from operational data.

In Fig. 1, we report the flow chart of this overall optimization framework.

6 Case Study on Sustainable Logistics in Northern Italy

In this section, we present computational experiments to demonstrate how the model and the optimization framework can serve as a reference point for managers and logistics distribution operators. The latter were implemented and solved using Python 3.14 and Gurobi 13.0.1 on a PC equipped with an Intel i9-13900K processor with 24 cores, 196 GB of RAM, and a Windows-based operating system.

6.1 Case Study Setup

The logistic network for the use case covers key industrial and commercial hubs in Northern Italy, including Milan, Turin, Bologna, Verona, Venice, Genoa, and Florence. Table 4 provides an overview of the network nodes within the logistics network, categorizing them as depots, parking locations, customer locations, and supply nodes. Each node is defined by its geographic coordinates and capacity, in terms of vehicle storage capacity or freight handling capability. Table 5 presents the distance matrix, which specifies the direct travel distances in kilometers between all nodes of the network. Table 6 describes the fleet composition, detailing vehicle types, payload capacities, and operational parameters. AVs are characterized by their energy consumption, maximum autonomy (range), emissions, charging

time, and cost per km. HDVs have all thermal engines and are characterized by their maximum autonomy, emissions, and cost per km. Table 7 defines the initial reliability values for each arc in the network, representing the chance that a given vehicle type can traverse a specific route without disruptions. Reliability parameters are critical to optimizing vehicle allocation and determining trade-offs between emissions, energy consumption, and operational costs. It is important to emphasize that the reliability values adopted in the computational study are exogenous scenario parameters used to calibrate the optimization experiments and are not the output of the illustrative logistic-regression pipeline described in Section 5.1. Moreover, it is important to further note that, in the numerical experiments, these reliability values are assumed to be identical across all AVs, so that reliability is effectively treated as an arc-level characteristic of the operating environment rather than as a vehicle-specific attribute. The more general notation λ_{ij}^{k} is retained in the formulation to allow future extensions with heterogeneous AV classes or vehicle-specific reliability profiles. Table 8 lists the initial depot locations and charge levels for each vehicle at the beginning of the planning horizon. Electric vehicles start with a specific charge level, which influences their initial range and the need for intermediate charging stops (thermal-engine vehicles are assumed to have negligible refueling time). Table 9 specifies the vehicle compatibility matrix, indicating which vehicles can access specific nodes. The depots are compatible with all vehicles, while the parking, customer, and supply nodes impose restrictions based on the type of vehicle and operational feasibility. In our case study, it is assumed that the arc compatibility is complete for all vehicles, that is, $\delta_{ijk}^{a} = 1, \forall (i,j) \in A,\ \forall k \in K^{a}$ and $\delta_{ijk}^{h} = 1, \forall (i,j) \in A,\ \forall k \in K^{h}$.

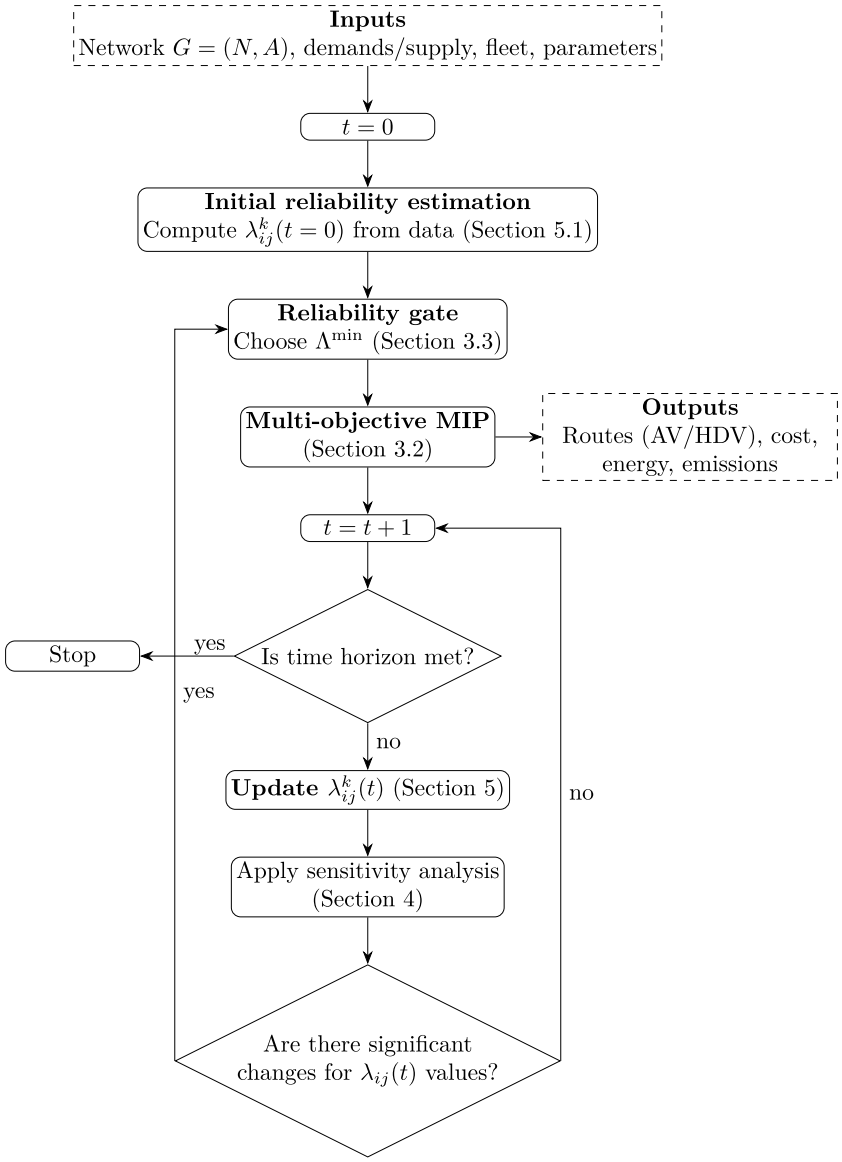


Fig. 1 Flow chart of the optimization framework.

Table 4 Network nodes with their geographic coordinates and capacities/demands

Node	Type	City	Latitude	Longitude	Capacity (vehicles)
D1	Depot	Milan	45.4642 N	9.1900 E	6 vehicles
D2	Depot	Turin	45.0703 N	7.6869 E	5 vehicles
D3	Depot	Verona	45.4384 N	10.9916 E	6 vehicles
P1	Parking	Brescia	45.5416 N	10.2118 E	4 vehicles
P2	Parking	Parma	44.8015 N	10.3279 E	5 vehicles
P3	Parking	Modena	44.6471 N	10.9252 E	5 vehicles
Node	Type	City	Latitude	Longitude	Demand (tons)
C1	Customer	Genoa	44.4056 N	8.9463 E	1.2 tons
C2	Customer	Bologna	44.4949 N	11.3426 E	1.5 tons
C3	Customer	Venice	45.4408 N	12.3155 E	1.8 tons
C4	Customer	Florence	43.7696 N	11.2558 E	1.4 tons
C5	Customer	Trieste	45.6495 N	13.7768 E	1.1 tons
C6	Customer	Padua	45.4064 N	11.8768 E	1.2 tons
Node	Type	City	Latitude	Longitude	Capacity (tons)
S1	Supply	Piacenza	45.0524 N	9.6928 E	6 tons
S2	Supply	Parma	44.8015 N	10.3279 E	8 tons
S3	Supply	Ferrara	44.8392 N	11.6198 E	8 tons
S4	Supply	Udine	46.0623 N	13.2345 E	6 tons

Generation and Calibration of the Synthetic Case-Study Parameters Although the logistics network considered in this paper is based on real geographic locations and real inter-city distances in Northern Italy, several operational parameters used in the computational study are synthetic. We therefore clarify here how the benchmark instance was generated and calibrated.

Depots, parking locations, customer nodes, and supply nodes correspond to actual cities in Northern Italy, and the distance matrix reported in Table 5 is derived from the corresponding road-network structure. Therefore, the spatial component of the benchmark is not randomly generated.

Second, customer demands and supply capacities are synthetic but *scenario-calibrated*. More precisely, the values reported in Table 4 were selected to represent plausible medium-scale regional freight-distribution volumes while preserving consistency with the adopted vehicle capacities and the overall size of the considered network. In this sense, "calibrated" means that the demand and supply values were designed to generate a realistic

Table 5 Distances in the network (km)

	D1	D2	D3	P1	P2	P3	C1	C2	C3	C4	C5	C6	S1	S2	S3	S4
D1	0	126	141	80	116	164	119	201	244	250	359	210	60	116	203	322
D2	126	0	263	205	211	260	124	296	366	319	481	331	158	211	311	447
D3	141	263	0	62	88	88	198	108	104	187	219	69	111	88	83	188
P1	80	205	62	0	83	114	161	147	165	214	278	131	68	83	135	242
P2	116	211	88	83	0	50	118	87	172	137	287	139	57	0	102	267
P3	164	260	88	114	50	0	160	37	141	101	250	113	107	50	59	240
C1	119	124	198	161	118	160	0	191	290	198	405	257	93	118	218	384
C2	201	296	108	147	87	37	191	0	130	81	231	110	145	87	44	229
C3	244	366	104	165	172	141	290	130	0	204	116	35	210	172	86	99
C4	250	319	187	214	137	101	198	81	204	0	289	188	189	137	122	299
C5	359	481	219	278	287	250	405	231	116	289	0	151	327	287	192	62
C6	210	331	69	131	139	113	257	110	35	188	151	0	176	139	66	128
S1	60	158	111	68	57	107	93	145	210	189	327	176	0	57	154	298
S2	116	211	88	83	0	50	118	87	172	137	287	139	57	0	102	267
S3	203	311	83	135	102	59	218	44	86	122	192	66	154	102	0	186
S4	322	447	188	242	267	240	384	229	99	299	62	128	298	267	186	0

Table 6 Fleet composition with capacity, range, energy consumption, emissions, time to recharge (electric vehicles), and cost per km

Vehicle Type	Engine Type	Q_k^a/Q_k^h (tons)	Range (km)	R_k^a/R_k^{he} (kWh/km)	C_k^a/C_k^h (kg CO_2/km)	τ_k^a/τ_k^{he} (h)	w_k^a/w_k^h ($/km)
AV1	Electric	1.5	435	0.39	0	0.92	1.1
AV2	Electric	1.5	435	0.39	0	0.92	1.1
AV3	Electric	1.5	435	0.39	0	0.92	1.1
AV4	Electric	1.5	435	0.39	0	0.92	1.1
AV5	Electric	1	275	0.36	0	0.58	0.8
AV6	Electric	1	275	0.36	0	0.58	0.8
AV7	Electric	1	275	0.36	0	0.58	0.8
AV8	Electric	1	275	0.36	0	0.58	0.8
HDV1	Thermal	3	550	N/A	0.28	N/A	2.2
HDV2	Thermal	3	550	N/A	0.28	N/A	2.2
HDV3	Thermal	3	550	N/A	0.28	N/A	2.2
HDV4	Thermal	3	550	N/A	0.28	N/A	2.2

mix of feasible and binding routing decisions rather than being drawn from an arbitrary random generator without operational interpretation.

Third, fleet parameters were calibrated from publicly available technical specifications and representative operational values. In particular, the capacities, ranges, energy-consumption coefficients, recharge times, emission factors, and distance-based operating costs of AVs and HDVs were assigned to reflect plausible differences between electric autonomous vehicles and conventional human-driven freight vehicles. The resulting fleet composition is therefore synthetic but built to preserve realistic scale relations among load, distance, energy, and cost.

Fourth, the arc-reliability matrix reported in Table 7 is also synthetic and should be interpreted as a *scenario-calibrated benchmark input*, not as the output of an empirical statistical estimation on a real AV traversal dataset. More precisely, the reliability values were assigned to reproduce a heterogeneous operating environment in which some corridors are highly favorable to AV deployment while others are closer to the admissibility threshold $\Lambda^{\min}$. This choice was made intentionally to test the role of the reliability-gating mechanism within the routing model. Therefore, Table 7 should not be read as an empirically fitted reliability map, but as a benchmark scenario designed to be operationally plausible and computationally informative.

Finally, the term "scenario-calibrated" is adopted in this paper to indicate that synthetic quantities were selected within plausible operational ranges and jointly tuned to generate a non-trivial mixed AV/HDV routing problem. This benchmark-construction approach is common in methodological optimization studies when real, fully integrated datasets are not available in a form suitable for reproducible experimentation. Nevertheless, we acknowledge that future work should validate the framework on empirically estimated demand and reliability data from real freight operations.

6.2 Analysis of Weighting Scenarios on Pareto-Optimal Solutions

To solve the multi-objective model, we employ a scalarization approach (weighted sum of the objectives), where we assign a weight to each of the four objectives to obtain a single-objective model. The evolution of weight vectors is illustrated below and summarized in Algorithm 3. Indeed, the proposed weight-adaptation mechanism is heuristic in nature, and its purpose is to guide the exploration of balanced trade-off solutions by adjusting the scalarization weights in response to observed objective deterioration. It is worth noting, in fact, that weighted-sum scalarization, while computationally convenient and easy to embed in the MIP framework, cannot recover non-convex portions of the Pareto front in discrete multi-objective problems. Therefore, the solution sets reported in this paper should be interpreted as informative approximations of the trade-off structure induced by the sampled scalarizations, rather than as an exhaustive reconstruction of the full Pareto set.

We begin with a uniform weight vector $\boldsymbol{\alpha}^{(1)} = (0.25, 0.25, 0.25, 0.25)$, where $\boldsymbol{\alpha}_i$ is associated with the i-th objective of the model. The scalarization of the four objectives via a convex combination is as follows:

$$\min_{x\in\mathcal{X}} \sum_{\ell=1}^{3} \alpha_\ell z_\ell(x) - \alpha_4 z_4 \quad \text{s.t.} \quad \alpha \in \Delta_4 := \left\{\alpha \in \mathbb{R}^4_{\geq 0} \mid \sum_{\ell=1}^{4} \alpha_\ell = 1\right\}, \tag{93}$$

where z_1, z_2, z_3 are the total emissions, total energy consumption, and total cost, respectively, while z_4 is the total reliability. $\mathcal{X}$ indicates the feasible region defined by the constraints of the model.

Table 7 Baseline reliability parameters for each arc. In the main case study, these values are assumed identical for all AVs, so that $\lambda_{ij}^k = \lambda_{ij}$ for every $k \in K^a$

	D1	D2	D3	P1	P2	P3	C1	C2	C3	C4	C5	C6	S1	S2	S3	S4
D1	–	1.00	1.00	1.00	0.87	0.87	0.83	1.00	1.00	1.00	0.82	1.00	1.00	0.89	0.88	0.88
D2	0.92	–	0.97	0.92	1.00	0.86	0.92	0.94	0.97	1.00	0.88	0.99	1.00	0.83	1.00	0.87
D3	0.84	1.00	–	1.00	0.92	0.85	1.00	0.97	0.86	0.99	0.83	1.00	0.90	1.00	0.92	1.00
P1	1.00	0.88	1.00	–	1.00	1.00	1.00	1.00	0.85	0.88	0.83	0.93	0.95	0.91	1.00	0.94
P2	0.91	1.00	0.86	1.00	–	1.00	1.00	0.88	0.82	1.00	1.00	1.00	1.00	0.84	0.94	0.86
P3	1.00	1.00	0.93	0.84	0.92	–	1.00	1.00	1.00	0.98	0.86	1.00	1.00	1.00	1.00	0.99
C1	1.00	0.96	0.82	0.85	0.83	1.00	–	0.99	1.00	0.90	0.96	1.00	0.89	0.84	0.92	0.87
C2	1.00	1.00	1.00	1.00	1.00	0.88	1.00	–	1.00	1.00	0.93	0.85	0.89	0.96	1.00	1.00
C3	0.82	0.99	0.96	0.89	0.86	0.93	1.00	0.93	–	1.00	0.94	1.00	1.00	0.90	0.99	0.92
C4	0.91	0.83	1.00	0.99	0.83	0.91	1.00	0.90	0.87	–	1.00	0.90	1.00	1.00	0.90	1.00
C5	0.94	1.00	1.00	1.00	0.85	1.00	0.93	0.88	0.83	1.00	–	0.82	0.99	0.89	1.00	0.88
C6	1.00	0.95	1.00	0.86	0.93	0.85	1.00	1.00	0.90	1.00	1.00	–	1.00	0.90	0.85	1.00
S1	1.00	1.00	0.93	0.94	1.00	1.00	1.00	1.00	1.00	0.84	0.87	1.00	–	0.82	0.85	1.00
S2	0.82	0.87	1.00	1.00	1.00	0.89	1.00	0.90	0.93	1.00	1.00	1.00	1.00	–	0.85	0.94
S3	0.91	0.90	1.00	0.95	1.00	1.00	1.00	0.99	1.00	0.99	0.88	1.00	0.91	0.82	–	0.88
S4	1.00	1.00	1.00	0.94	0.82	1.00	0.96	1.00	1.00	1.00	0.92	0.95	1.00	0.93	0.87	–

Normalized $\Delta_\ell^{(t)}$ At iteration t we solve Eq. 93 with weights $\alpha^{(t)}$ and obtain objective values $z_\ell^{(t)}$. For minimization objectives ($\ell \in \{1,2,3\}$), we define the normalized deterioration

$$\Delta_\ell^{(t)} := \frac{z_\ell^{(t)} - z_\ell^{(t-1)}}{|z_\ell^{(t-1)}| + \varepsilon},$$

while for the reliability term ($\ell = 4$), where larger is better, we flip the sign:

$$\Delta_4^{(t)} := \frac{z_4^{(t-1)} - z_4^{(t)}}{|z_4^{(t-1)}| + \varepsilon}.$$

Here $\varepsilon > 0$ prevents division by zero (we use $\varepsilon = 10^{-9}$).

Weight Update and Projection Let $g_\ell^{(t)} := \mathrm{clip}(\Delta_\ell^{(t)}, -\gamma, \gamma)$ with a clipping parameter $\gamma > 0$ to temper outliers. We update *pre-normalized* weights

$$\tilde{\alpha}_\ell^{(t+1)} = \alpha_\ell^{(t)} + \eta\, g_\ell^{(t)}, \qquad \ell = 1, \ldots, 4,$$

with stepsize $\eta > 0$, enforce lower/upper box safeguards $0 \le \tilde{\alpha}_\ell^{(t+1)} \le 1$, then project onto the simplex:

$$\alpha^{(t+1)} = \Pi_{\Delta_4}(\tilde{\alpha}^{(t+1)}) \quad \text{where} \quad \Pi_{\Delta_4}(u) := \arg\min_{v \in \Delta_4} \|v - u\|_2.$$

In implementation, we use the standard "shift-and-threshold" projection: subtract a common scalar τ so that $\sum_{\ell=1}^{4} \max\{u_\ell - \tau, 0\} = 1$ and then apply the positive part.

Algorithm 3 Objective weight adaptation.

Input: Initial weights $\alpha^{(1)} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$, stepsize $\eta > 0$, clip $\gamma > 0$, tolerance $\varepsilon > 0$, max iters T. Let $z^{(0)} = 0$.
for $t = 1, \ldots, T$ **do**
  Solve Eq. 93 with $\alpha^{(t)}$ and obtain $z_\ell^{(t)}$, with $\ell = 1, \ldots, 4$
  for $\ell \in \{1, 2, 3\}$ **do**
    $\Delta_\ell^{(t)} \leftarrow \dfrac{z_\ell^{(t)} - z_\ell^{(t-1)}}{|z_\ell^{(t-1)}| + \varepsilon}$
  end for
  $\Delta_4^{(t)} \leftarrow \dfrac{z_4^{(t-1)} - z_4^{(t)}}{|z_4^{(t-1)}| + \varepsilon}$
  for $\ell = 1$ **to** 4 **do**
    $g_\ell^{(t)} \leftarrow \mathrm{clip}(\Delta_\ell^{(t)}, -\gamma, \gamma)$
    $\tilde{\alpha}_\ell^{(t+1)} \leftarrow \min\{1, \max\{0, \alpha_\ell^{(t)} + \eta g_\ell^{(t)}\}\}$
  end for
  $\alpha^{(t+1)} \leftarrow \Pi_{\Delta_4}(\tilde{\alpha}^{(t+1)})$
  if stopping criterion satisfied **then**
    break
  end if
end for

Table 8 Initial depot location and initial state-of-charge of vehicles (where applicable)

Vehicle	Initial Depot	Initial Charge Level (kWh)
AV 1	D1	85
AV 2	D2	90
AV 3	D1	80
AV 4	D1	95
AV 5	D2	88
AV 6	D1	92
AV 7	D1	87
AV 8	D2	89
HDV 1	D3	–
HDV 2	D2	–
HDV 3	D1	–
HDV 4	D1	–

Table 9 Node-vehicle compatibility matrix associated with ω_{ki}^{h} and ω_{ki}^{a} parameters

Node	Compatible Vehicles
D1	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
D2	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
D3	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
P1	AV 1, AV 3, AV 4, AV 6, AV 7, HDV 3, HDV 4
P2	AV 2, AV 5, AV 8, HDV 1, HDV 2
P3	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
C1	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
C2	AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2
C3	AV 1, AV 2, AV 3, AV 4, HDV 1, HDV 2, HDV 3, HDV 4
C4	AV 1, AV 2, AV 3, AV 4, AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
C5	AV 5, AV 6, AV 7, AV 8, HDV 1, HDV 2
C6	AV 1, AV 2, AV 3, AV 4, HDV 1, HDV 2, HDV 3, HDV 4
S1	AV 1, AV 3, AV 4, AV 6, AV 7, HDV 1, HDV 2, HDV 3, HDV 4
S2	AV 1, AV 3, AV 4, AV 6, AV 7, HDV 1, HDV 2, HDV 3, HDV 4
S3	AV 2, AV 5, AV 8, HDV 1, HDV 2, HDV 3, HDV 4
S4	AV 2, AV 5, AV 8, HDV 1, HDV 2, HDV 3, HDV 4

Algorithm Parameters and Stopping Criteria We terminate when (i) the last W iterations produce relative changes $|\Delta_\ell^{(t)}| \le \delta$ for all ℓ, or (ii) an iteration budget T is met. Typical values are $\eta \in [0.05, 0.2]$, $\gamma \in [0.05, 0.1]$, $W = 3$, and $\delta = 10^{-3}$.

Example Consider iteration k with current weights:

$$\alpha^{(k)} = (0.40,\ 0.25,\ 0.15,\ 0.20), \qquad \sum_{\ell=1}^{4} \alpha_\ell^{(k)} = 1.$$

Assume that the objective values at $t = k-1$ and $t = k$ are:

$$\left(z_1^{(k-1)}, z_2^{(k-1)}, z_3^{(k-1)}, z_4^{(k-1)}\right) = (100,\ 200,\ 50,\ 0.80),$$

$$\left(z_1^{(k)}, z_2^{(k)}, z_3^{(k)}, z_4^{(k)}\right) = (103,\ 194,\ 51,\ 0.78).$$

Let $\varepsilon = 10^{-9}$, clip bound $\gamma = 0.05$, and stepsize $\eta = 0.1$.

Step 1 — normalized deltas. For minimization objectives $\ell \in \{1, 2, 3\}$,

$$\Delta_1 = \frac{103 - 100}{100} = 0.03, \qquad \Delta_2 = \frac{194 - 200}{200} = -0.03, \qquad \Delta_3 = \frac{51 - 50}{50} = 0.02.$$

For reliability (larger is better),

$$\Delta_4 = \frac{0.80 - 0.78}{0.80} = 0.025.$$

Step 2 — clipping. Since all $\Delta_\ell \in [-\gamma, \gamma] = [-0.05, 0.05]$,

$$g_\ell = \mathrm{clip}(\Delta_\ell, -\gamma, \gamma) = \Delta_\ell \quad \Rightarrow \quad g = (0.03,\ -0.03,\ 0.02,\ 0.025).$$

Step 3 — pre-projection update.

$$\tilde{\alpha}_\ell = \alpha_\ell^{(k)} + \eta\, g_\ell$$

gives

$$\tilde{\alpha}_1 = 0.403, \quad \tilde{\alpha}_2 = 0.247, \quad \tilde{\alpha}_3 = 0.152, \quad \tilde{\alpha}_4 = 0.2025,$$

with sum $0.403 + 0.247 + 0.152 + 0.2025 = 1.0045$.

Step 4 — simplex projection. Because all components are positive, the Euclidean projection onto $\Delta_4 = \{\alpha \geq 0 : \sum_{\ell=1}^{4} \alpha_\ell = 1\}$ subtracts a common shift

$$\tau = \frac{1.0045 - 1}{4} = 0.001125.$$

Thus,

$$\alpha^{(k+1)} = \tilde{\alpha} - \tau 1 = \big(0.401875,\ 0.245875,\ 0.150875,\ 0.201375\big),$$

which are nonnegative and sum to 1.

Figure 2 presents a graph that illustrates the evolution of the weight components $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ over 50 generations, as generated by the Objective Weight Adaptation algorithm. This visualization reveals the dynamic adjustment of the weight vector in response to the observed changes in the values of the objective function. At iteration 1 the algorithm starts from the uniform prior $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.25, 0.25, 0.25, 0.25)$. At each step, the feedback rule compares normalized objective changes and shifts toward objectives that deteriorated, while projecting back to the simplex.

6.3 Impact of the Network Reliability on Emissions

Figure 3 represents the trade-off between emissions (kg CO2) and reliability in the optimization model. The x-axis corresponds to the total emissions generated by the fleet, while the y-axis indicates the average reliability of the transportation network. The curve is piecewise linear, reflecting the discrete nature of optimal solutions derived from the optimization model.

The downward trend of the Pareto front suggests an inverse relationship between emissions and reliability. As emissions decrease, reliability tends to improve, but this relationship is not uniform across the entire range. The presence of flat segments and sharp transitions in the curve indicates that certain operational configurations maintain reliability while reducing emissions up to a certain threshold, after which further emissions reductions result in a steeper decline in reliability.

This behavior is likely a result of how the logistics system allocates AVs and HDVs in response to network conditions. In scenarios with low emissions, a higher proportion of AVs will likely be used as they produce fewer emissions compared to HDVs. However, AVs depend on network conditions for their effectiveness, meaning that, in deteriorating network links with lower reliability, HDVs might be preferred because of their ability to operate in less predictable environments. This substitution of AVs with HDVs leads to increased emissions, but improved reliability in certain cases.

In addition, the chart shows sharp declines in reliability at certain emission thresholds, which could indicate critical changes in vehicle allocation or routing strategies. For example, beyond a certain level of emissions, the model might begin to introduce more HDVs into the fleet to compensate for unreliable network conditions, leading to localized jumps in reliability.

If the logistic operator seeks to reduce emissions while maintaining high reliability, this chart can be used to determine threshold points where trade-offs become more severe. These insights could be used to make strategic investments in infrastructure, autonomous vehicle technology, or routing algorithms to balance emissions and reliability more effectively.

6.4 Analysis of Network Reliability, Stability, and Cost

In the following, we analyze how fluctuations in network reliability and the stability analysis proposed in Section 4 impact rescheduling costs, considering ten decision time points and the total operational cost as the objective function. Algorithm 2 proposed in Section 5 has been implemented starting from the proposed baseline instance, assuming random variations of λ_{ij}^k values over time. Parameter ϵ is set to 0.05. Figure 4 presents a four-panel analysis with four subplots, which represent, in order:

(1) The average reliability values λ_{ij}^k over all arcs $(i, j) \in A$ and $k \in K^a$ over time.
(2) The distribution of stability intervals for λ_{ij}^k values based on sensitivity analysis, presented as box plots.
(3) The total operational cost computed at each decision point.
(4) The percentage variation in total operational cost from one decision time to the next.

The first chart shows how the average reliability λ_{ij}^k changes over time. The horizontal dashed line at $\Lambda^{\min} = 0.90$ represents the critical threshold below which the reliability of AVs begins to deteriorate significantly (see Constraint 52). When reliability drops below this threshold, operational adjustments, such as increasing the share of HDVs, are required to maintain service stability. In general, these fluctuations can trigger rescheduling procedures if the change in λ_{ij}^k for certain arcs is outside the corresponding stability

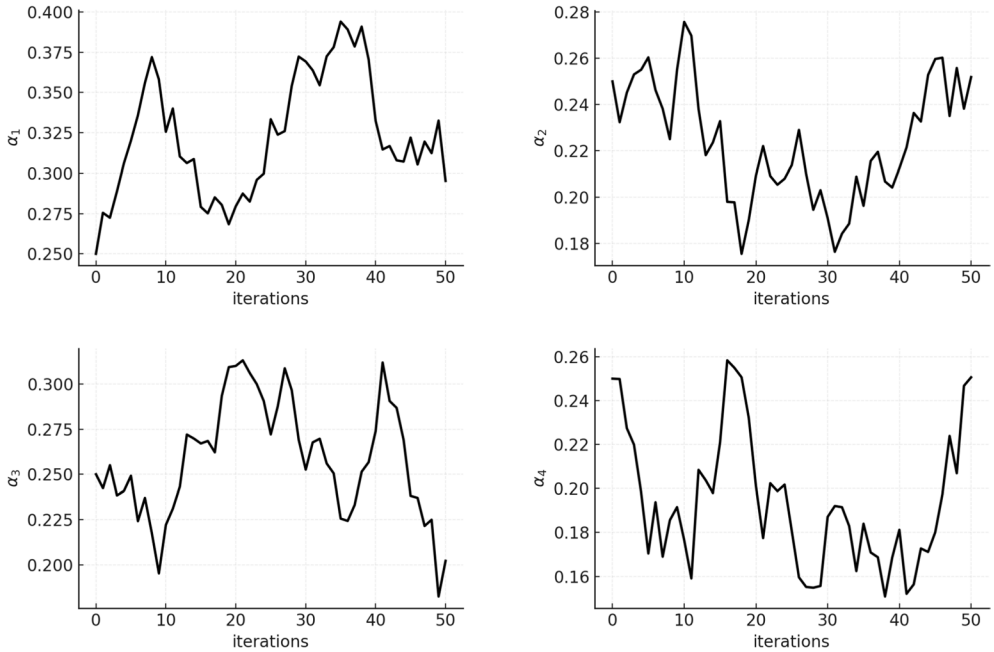
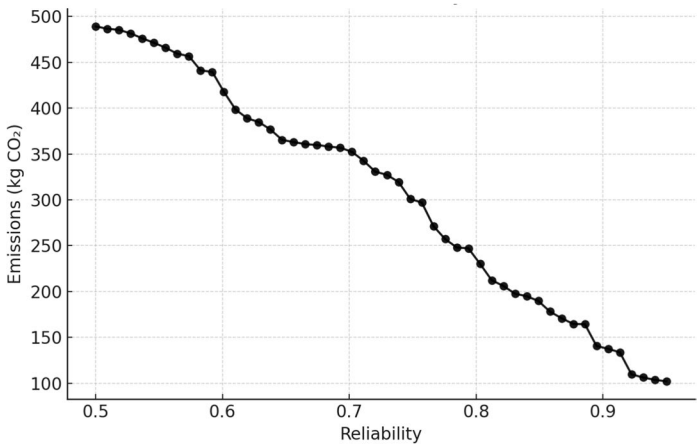


Fig. 2 Trends of the adaptive scalarization weights over iteration

Fig. 3 The trade-off between emissions and reliability



intervals. The second chart visualizes these stability intervals. Narrower boxes indicate more constrained intervals in which λ_{ij}^k can fluctuate without affecting optimal routing decisions. Wider intervals offer greater flexibility, reducing the need for rescheduling.

The third chart illustrates the total operational cost, which reflects how the system reacts to changes in reliability. When average λ_{ij}^k values fall below the threshold, and their deviation exceeds the stability range, rescheduling is activated. This leads to an increase in cost, as captured in the third and fourth charts.

The final chart displays the percentage change in operational cost relative to the previous decision point, highlighting spikes associated with rerouting and the deployment of HDVs. Together, the four panels demonstrate how network reliability and its allowed fluctuation bounds (stability) are directly linked to operational rescheduling and associated cost variation.

In Table 10, we report the actions and computing times associated with the 10 decision points. A decision point typically occurs every 10 to 15 minutes after the previous one.

To further analyze the role of $\Lambda^{\min}$, Fig. 5 reports the share of total distance covered by AVs at the 10 decision time points for five reliability thresholds, i.e., $\Lambda^{\min} = 0.90, 0.89, 0.88, 0.87, 0.86$. As can be observed, trends tend to improve and become flat as soon as $\Lambda^{\min}$ is relaxed $(0.90 \rightarrow 0.86)$.

6.5 Supplementary Consistency Experiments

In this section, two supplementary experiments are illustrated on the same Northern-Italy case study of Tables 4–9. The aim of these additional runs is twofold:

- to verify whether the vehicle-specific notation λ_{ij}^k becomes operationally relevant when AVs are assigned heterogeneous reliability profiles, and
- to quantify the LP-relaxation gap underlying the screening mechanism proposed in Section 4, together with a direct validation of the screening trigger against observed rerouting changes.

To make repeated MIP/LP re-solves computationally tractable in the present environment, the experiments were run on a *cost–reliability scalarized variant* of the original model, adopting the following objective function

$$\min \sum_{k\in K}\sum_{(i,j)\in A} w_k d_{ij} x_{ij}^k \; - \alpha_4 \sum_{k\in K^a}\sum_{(i,j)\in A} \lambda_{ij}^k x_{ij}^k,$$

with $\alpha_4 = 50$. Thus, operating cost remains the dominant term while AV reliability still contributes explicitly to the scalarized objective, which is important for the LP-based sensitivity analysis.

6.5.1 Heterogeneous AV Reliabilities

In the experiments conducted in the previous subsections, we assumed homogeneous AV reliabilities, i.e., $\lambda_{ij}^k = \lambda_{ij}$ for all $k \in K^a$. Now, AVs are partitioned into two classes,

$$K^{a,+} = \{AV1, AV2, AV3, AV4\}, \qquad K^{a,-} = \{AV5, AV6, AV7, AV8\},$$

and, for every arc $(i, j) \in A$, the baseline reliability λ_{ij} is transformed into vehicle-specific values according to

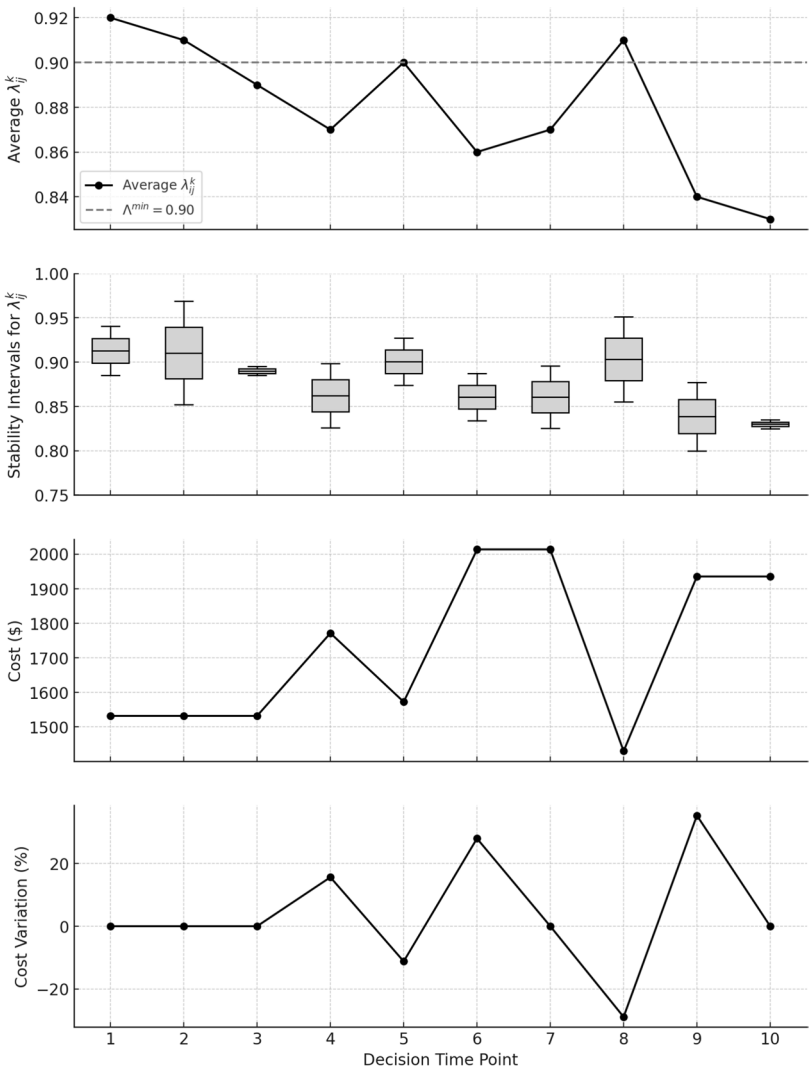


Fig. 4 Evolution of network metrics across ten decision time points

Table 10 CPU time per decision point

Decision point	Trigger on λ_{ij}^k	Action	Time (seconds)
1	initial plan	Full solve	20
2	within stability intervals; cost flat	—	0
3	active arcs still within ranges; cost flat	—	0
4	drift beyond ranges; cost ↑	Warm-start	8
5	local recovery; cost ↓	Warm-start	6
6	widespread reliability violation; cost ↑ sharply	Full re-solve	19
7	stable vs. decision point 6; cost flat	—	0
8	reliability rebound; cost ↓ sharply	Warm-start	7
9	reliability collapse; cost ↑ sharply	Full re-solve	20
10	within ranges; cost flat	—	0
Total over 10 points			80
Re-optimizations (incl. 1)			6
Avg time (seconds) when triggered			13.7

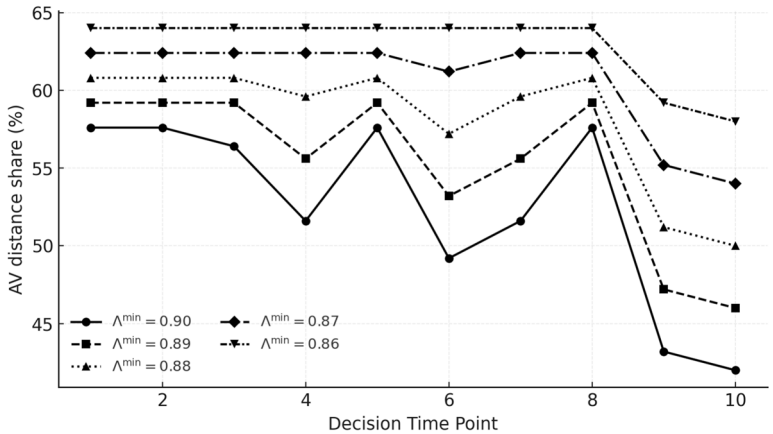


Fig. 5 Share of total distance covered by AVs over 10 decision time points for five reliability thresholds ($\Lambda^{\min} \in \{0.90, 0.89, 0.88, 0.87, 0.86\}$)

$$\lambda_{ij}^k = \begin{cases} \min\{1, \lambda_{ij} + 0.02\}, & \text{if } k \in K^{a,+}, \\ \max\{0, \lambda_{ij} - 0.03\}, & \text{if } k \in K^{a,-}. \end{cases}$$

This perturbation preserves the original network and demand settings while activating the vehicle-specific index in the reliability parameter.

Table 11 shows that the introduction of vehicle-specific reliability profiles materially changes the solution to the original benchmark. The best incumbent objective increases from 1952.40 to 2464.40, with a corresponding increase of 26.22% (see also Fig. 6). At the same time, the number of active AV arcs rises from 6 to 9, the number of AV routes rises from 2 to 3, and seven AV arcs differ from the homogeneous baseline. This confirms that the k -indexed formulation is operationally meaningful once AV heterogeneity is allowed, while the homogeneous assumption of the main experiments should be regarded as a simplifying benchmarking choice (Fig.).

6.5.2 LP Relaxation Gap and Validation of the Screening Mechanism

A second supplementary experiment was performed to clarify the practical interpretation of the LP-based screening mechanism introduced in Section 4. The analysis consists of two steps: first, computing the LP-relaxation gap on the original benchmark; second, testing whether the screening rule correctly predicts observed rerouting changes under a sequence of reliability degradations.

LP-Relaxation Gap The scalarized MIP defined above was solved on the original case study instance under the

Table 11 Homogeneous vs. heterogeneous AV reliability experiment

Metric	Homogeneous baseline	Heterogeneous AV reliabilities
Objective value	1952.40	2464.40
Variation (%)	—	+26.22%
Number of active AV arcs	6	9
Number of AV routes	2	3
Changed AV arcs	—	7
AV distance share (%)	39.71	46.67

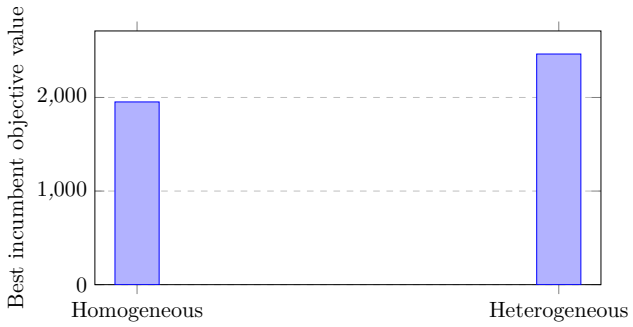


Fig. 6 Best incumbent objective value under homogeneous and heterogeneous AV reliability assumptions

homogeneous baseline reliability matrix, and the corresponding LP relaxation was then solved by relaxing all binary variables while preserving the same coefficient structure and parameter values (Table 12).

The resulting LP gap is large (93.22%), which confirms that the LP relaxation is a loose bound for this routing formulation on the original benchmark. Therefore, the sensitivity intervals derived from the LP relaxation should be interpreted as *screening bands* for rerouting decisions rather than as exact post-optimality certificates for the mixed-integer optimum.

Dynamic Screening-Validation Protocol To assess the practical usefulness of the screening mechanism, a short dynamic pilot was run over four decision points. Starting from the homogeneous baseline reliability matrix, three progressively degraded matrices were generated by multiplying all arc reliabilities by the factors

1.000, 0.9975, 0.9950, 0.9925.

At each decision point, the scalarized MIP was re-solved with a 10 s time cap. A rerouting change was classified as *significant* whenever either (i) the incumbent objective value changed by more than 1% with respect to the previous decision point, or (ii) the set of active AV arcs changed. The screening rule of Section 4 was then evaluated for

$$\epsilon \in \{0.01, 0.02, 0.05, 0.10\}.$$

In these runs, the dual multipliers of the reliability rows were numerically zero, so the first-order screening derivative was driven by the LP AV-arc usage term of the envelope formula. This is still consistent with the general sensitivity expression, but it further reinforces the interpretation of the screening mechanism as a practical rerouting trigger rather than as an exact dual-based certificate.
Tables 13 and 14, together with Fig. 7, show that all three transitions of the dynamic pilot correspond to meaningful rerouting changes according to the adopted significance criterion, and that the screening mechanism correctly flags all of them for every tested value of ϵ. Thus, on this original benchmark pilot, the screening rule behaves as an effective early-warning trigger.

Overall Interpretation The heterogeneous-reliability test confirms that the vehicle-specific notation λ_{ij}^{k} is meaningful once AV heterogeneity is activated. The LP-relaxation and dynamic pilot results show that the screening mechanism can provide useful rerouting signals on the original case study, while also making explicit that its theoretical support comes from a relatively loose LP surrogate. This is precisely why the screening bands are described as triggering intervals and not as exact post-optimality guarantees.

Table 12 Scalarized MIP incumbent, LP-relaxation value, and relative LP gap

Metric	Value
Best incumbent MIP objective	1952.40
LP-relaxation objective	132.46
Relative LP gap (%)	93.22

6.6 Mixed Fleet Cost vs HDV-Only Fleet Cost with Varying Client Nodes

Figure 8 examines the impact of increasing the number of client nodes on the total operational cost for a mixed fleet consisting of AVs and HDVs compared to a fleet with only HDVs. Basically, we considered the same setup of the case study in Tables from Tables 4 to 9, where clients are varied from 5 to 20 with steps of 5. Customers' demands are defined in a wraparound fashion as follows: customers from 1 to 5 have the same demands as in Table 4; customers from 6 to 10, from 11 to 15, and from 16 to 20 have the same

Table 13 Objective evolution over the dynamic decision points

Decision point	Scaling factor	Avg. λ	Best incumbent objective	Active AV arcs
1	1.0000	0.9451	1952.40	6
2	0.9975	0.9427	2242.72	18
3	0.9950	0.9404	2930.70	6
4	0.9925	0.9380	3220.02	10

Table 14 Screening accuracy for different values of ϵ in the dynamic pilot

ϵ	TP	FP	FN	TN	Precision	Recall
0.01	3	0	0	0	1.00	1.00
0.02	3	0	0	0	1.00	1.00
0.05	3	0	0	0	1.00	1.00
0.10	3	0	0	0	1.00	1.00

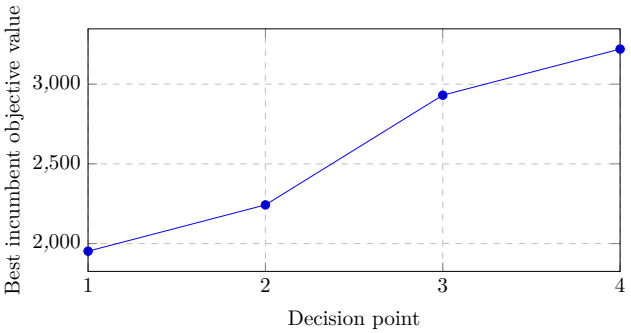


Fig. 7 Evolution of the best incumbent objective value across the four decision points of the dynamic pilot

demands as clients from 1 to 5, respectively. Customers from 1 to 5 are located in the same way as those in Table 4. Locations of customers from 6 to 10, from 11 to 15, and from 16 to 20 are generated at random in a range of 20 Km, 30 km, and 40 km from customers 1 to 5, respectively.

The x-axis in Fig. 8 represents the number of client nodes, while the y-axis denotes the total operational cost. The solid curve represents the cost associated with the mixed fleet, while the dashed curve represents the cost incurred when only HDVs are used for logistics operations. In these experiments, the reliability of the systems does not change over time and is assumed to be the same as that in Table 7.

The results indicate that as the number of client nodes increases, the operational cost increases for both fleet configurations. The mixed fleet consistently incurs lower costs than the HDV-only fleet, and the cost difference between the two fleets becomes more pronounced as the number of client nodes increases, demonstrating the advantage of incorporating AVs in larger-scale logistics operations.

The trend observed in the chart is the result of several operational factors. With more locations for clients, the complexity of the logistics network increases, leading to longer travel distances, higher fuel or energy consumption, and greater scheduling restrictions. The HDV-only fleet experiences a sharper increase in costs, and, in contrast, the mixed fleet benefits from AVs' lower per-mile operational costs, autonomy in navigation, and optimized scheduling, which contribute to its comparatively lower cost trajectory.

This analysis highlights the scalability benefits of deploying AVs in distribution logistics. Although the initial investment in AV infrastructure may be significant, long-term savings in operational costs justify the transition to automation. The results emphasize that the cost-effectiveness of AV integration becomes more apparent as logistics networks grow in complexity and size. Decision-makers can use this information to make inferences about the composition of the fleet, ensuring a balanced approach to cost optimization, service reliability, and sustainability in large-scale logistics operations.

6.7 Analysis on the Computational Scalability of the Model

The computational performance of the proposed model has been analyzed over a set of test instances characterized by varying network sizes. The setting of this test is the same as that of the previous section, with customers varying from 5 to 25 (Fig. 9). For smaller networks containing five client nodes, the model is solved in approximately 20 seconds. As the number of client nodes increases, the computational effort exhibits an exponential growth pattern. When reaching 20 client nodes, the solving time increases to 6852 seconds, indicating the computational burden imposed by increasing network complexity. For networks with 25 client nodes, the solver reaches the imposed time limit of 7200 seconds without obtaining an optimal solution. This behavior is expected as the number of decision variables and constraints grows significantly with increasing network size, particularly due to the combinatorial nature of vehicle routing and scheduling.

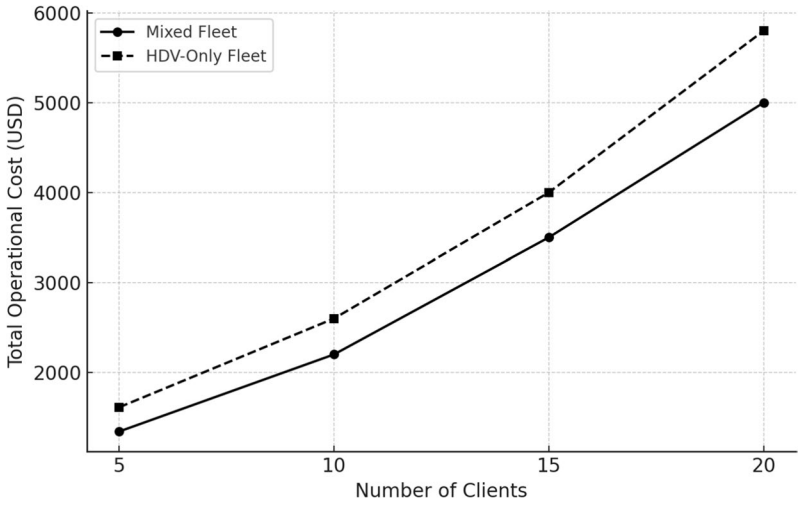


Fig. 8 Cost Comparison: Mixed fleet vs HDV-only fleet with varying client nodes

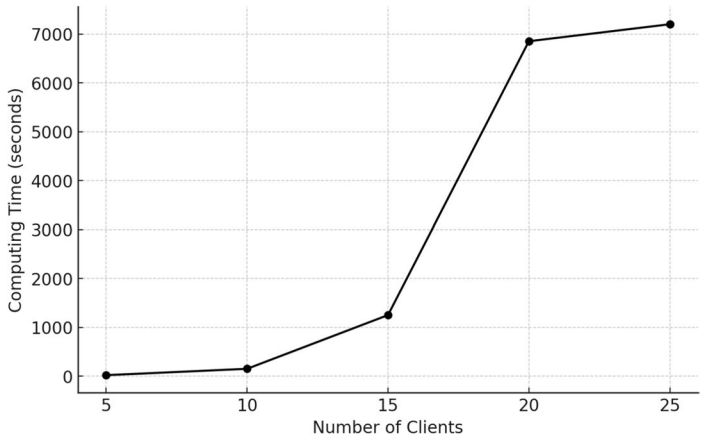


Fig. 9 Computing time as a function of the number of clients

This analysis provides insight into the scalability of the proposed optimization framework. Although the model performs efficiently for medium-scale networks, large-scale instances require advanced solution strategies, such as decomposition techniques, heuristic approaches, or parallel computing frameworks to ensure practical applicability in real-world logistics scenarios.

6.8 Strengthening the Proposed Formulation

In order to improve the model performance, in the following, we present three families of valid inequalities, i.e., T1, T2, and T3.

(T1) Load-based Constraints We enforce load-based MTZ (Miller–Tucker–Zemlin) subtour elimination constraints (see, e.g. Desrochers and Laporte [11], Kara et al. [17], Toth and Vigo [33], Yaman [36]). Define node departure load variables as follows:

$$u^a_{ik} := \sum_{\ell \in N} q^a_{i\ell k} \qquad \forall i \in N,\ \forall k \in K^a,$$

$$u^h_{ik} := \sum_{\ell \in N} q^h_{i\ell k} \qquad \forall i \in N,\ \forall k \in K^h,$$

with bounds $0 \le u^a_{ik} \le Q^a_k$, $0 \le u^h_{ik} \le Q^h_k$ and the standard activation $0 \le q^a_{ijk} \le Q^a_k a^k_{ij}$ (AV) or $0 \le q^h_{ijk} \le Q^h_k h^k_{ij}$ (HDV).

For every arc $(i, j) \in A$, the following inequalities enforce the exact load change at node j whenever the arc is used:

AVs ($k \in K^a$):

$$\begin{aligned} u^a_{jk} &\ge u^a_{ik} + \Delta^a_{jk} - \Theta^a_{jk} - Q^a_k (1 - a^k_{ij}), \\ u^a_{jk} &\le u^a_{ik} + \Delta^a_{jk} - \Theta^a_{jk} + Q^a_k (1 - a^k_{ij}). \end{aligned} \tag{94}$$

HDVs ($k \in K^h$):

$$\begin{aligned} u^h_{jk} &\geq u^h_{ik} + \Delta^h_{jk} - \Theta^h_{jk} - Q^h_k(1 - h^k_{ij}), \\ u^h_{jk} &\leq u^h_{ik} + \Delta^h_{jk} - \Theta^h_{jk} + Q^h_k(1 - h^k_{ij}). \end{aligned} \tag{95}$$

(At nodes j that are not supplies, set $\Delta^{\cdot}_{jk} = 0$; at nodes that are not customers, set $\Theta^{\cdot}_{jk} = 0$.)

If (i, j) is used by k (a^k_{ij}=1 or h^k_{ij}=1), Eqs. 94–95 reduce to $u^{\cdot}_{jk} = u^{\cdot}_{ik} + \Delta^{\cdot}_{jk} - \Theta^{\cdot}_{jk}$. If the arc is not used, both inequalities are implied by $0 \leq u^{\cdot}_{\cdot k} \leq Q^{\cdot}_k$. Summing these equalities along any cycle of customers (no supplies) strictly decreases u^k by $\sum \Theta^{\cdot}_{jk} > 0$ yet returns to the start, a contradiction; hence customer-only subtours are cut.

(T2) Tight Big-*M* for AV Energy Calculation with Lower Bounds The following constraints implement the standard state-of-charge recursion with activation and use node-specific residual-energy lower bounds (via reverse energy shortest paths). This aligns with common EVRP/EVRPTW modeling where state-of-charge conservation is enforced, and resource bounds (reachability to charging stations or depot) are used to restrict the state space and/or strengthen the master [10, 28].

For ease of presentation, the following analysis is performed for $k \in K^a$ (can be straightforwardly expanded to $k \in K^{he}$ adjusting the notation). Let η^k_i be the state-of-charge of k upon arrival at node i, bounded by $0 \leq \eta^k_i \leq M^a_k$. For each arc $(i, j) \in A$, let $c^k_{ij} \geq 0$ be a valid arc energy lower bound for k:

$$c^k_{ij} := R^a_k\, d_{ij} \qquad \text{or} \qquad c^k_{ij} := \left(\frac{M^{\cdot}_k}{\text{range}_k}\right) d_{ij}$$

($v^{\max}_k$ a conservative speed limit and range_k the maximum autonomy of the vehicle.) By construction $0 \leq c^k_{ij} \leq e^a_{ijk}$ for all admissible conditions. Define the node-specific state-of-charge lower bound

$$L^k_j := \min_{p \in P} \text{SP}^k_E(j \to p),$$

where $\text{SP}^k_E(j \to p)$ is the minimum state-of-charge needed by k to reach some $p \in P$ (a shortest path on the directed graph with arc costs c^k_{ij}). We replace the standard energy big-M by

$$\eta^k_j \geq \eta^k_i - c^k_{ij} - (M^a_k - L^k_j)(1 - a^k_{ij}), \qquad \forall (i, j) \in A,\ \forall k \in K^a, \tag{96}$$

together with $0 \leq \eta^k_i \leq M^a_k$ and $0 \leq \eta^k_j \leq M^a_k$.

If $a^k_{ij} = 1$, Eq. 96 reduces to $\eta^k_j \geq \eta^k_i - c^k_{ij}$. If $a^k_{ij} = 0$, the RHS equals $\eta^k_i - c^k_{ij} - (M^a_k - L^k_j) \leq L^k_j$, so Eq. 96 is implied by the bound $\eta^k_j \geq L^k_j$. Hence, it is valid and does not remove any integer-feasible solution.

To compute L^k_j, run one reverse Dijkstra per vehicle type on the graph with edge weights c^k_{ij}, taking every $p \in P$ as a source; the resulting distance to j is L^k_j as reported in Algorithm 4.

Algorithm 4 Reverse Dijkstra for node lower bounds L^k_j.

1: Create a new super-sink s; for each $p \in P$ add arc (p, s) with cost 0
2: Run Dijkstra on the *reverse* graph with edge weights c^k_{ij}, sources $\{s\}$ (or equivalently, multi-source Dijkstra from P on the forward graph)
3: **for** each node $j \in N$ **do**
4: $\quad L^k_j \leftarrow \text{distance}(j, s)$
5: **end for**

(T3) Energy-Feasible *m*-Path Cuts The following constraints are infeasible-path energy cuts that forbid short arc patterns whose cumulative energy exceeds the battery unless an intermediate charge occurs. These are local, compact instances of the infeasible path cuts introduced in recent EVRP branch-and-cut frameworks [29], and conceptually parallel the energy-feasibility pruning used in exact EVRP algorithms [10]. For ease of presentation, the following analysis is performed for $k \in K^a$ (can be straightforwardly expanded to $k \in K^{he}$ adjusting the notation).

Between two consecutive recharges, an AV cannot accumulate more than M^a_k energy consumption. Fix a node $j \notin P$ (no recharge) and an integer $m \geq 2$. Define the *best-case m-step energy through j*

$$\underline{C}^{(m),k}_j := \min\left\{ \sum_{r=1}^{m} c^k_{v_{r-1}v_r} \;\middle|\; v_0, \ldots, v_m \text{ a directed walk with } v_1 = j \text{ or } v_{m-1} = j \right\}.$$

If $\underline{C}^{(m),k}_j > M^a_k$, then no feasible no-recharge segment can pass through j with m consecutive arcs; hence, vehicle

k cannot both enter and leave j within such a segment. Enforce

$$\sum_{(i,j)\in A} a_{ij}^k + \sum_{(j,\ell)\in A} a_{j\ell}^k \le 1 \qquad \text{whenever } \underline{C}_j^{(m),k} > M_k^a. \tag{97}$$

Assume by contradiction that there exists a feasible AV route for k with a no-recharge segment that enters j from some i and leaves to some ℓ, forming an m-step subwalk as above. By definition of $\underline{C}_j^{(m),k}$, that subwalk costs at least $\underline{C}_j^{(m),k}$ energy, which exceeds M_k^a, contradicting feasibility of any no-recharge segment. Thus, at most one of the two events "enter j" / "exit j" can occur inside the same segment, proving Eq. 97.

When $m = 2$, $\underline{C}_j^{(2),k} = \min_{i,\ell}(c_{ij}^k + c_{j\ell}^k)$. If $c_{ij}^k + c_{j\ell}^k > M_k^a$ for a pair of neighbors (i, j) and (j, ℓ), we obtain the local 2–path cut

$$a_{ij}^k + a_{j\ell}^k \le 1,$$

which can be generated by scanning only the adjacency lists of j (complexity $O(\deg(j)^2)$ per j) as reported in Algorithm 5.

Algorithm 5 Generation of m=2 energy 2-path cuts $a_{ij}^k + a_{j\ell}^k \le 1$.

```
for each k ∈ K^a do
  for each j ∈ N \ P do
    for each (i, j) ∈ A with i ≠ j do
      for each (j, ℓ) ∈ A with ℓ ≠ j do
        if c^k_ij + c^k_jℓ > M^a_k then
          Add cut: a^k_ij + a^k_jℓ ≤ 1
        end if
      end for
    end for
  end for
end for
```

If $c_{ij}^k + c_{j\ell}^k > M_k^a$, then any no-recharge segment containing consecutive arcs $i \to j$ and $j \to \ell$ would exceed the battery budget, contradicting feasibility. Hence at most one of the two arcs can be chosen in a no-recharge segment, i.e., $a_{ij}^k + a_{j\ell}^k \le 1$. This is precisely the m=2 case of inequality Eq. 97 proved in T3.

Further Computational Results In Fig. 10, we compare the runtime performance of the two models, i.e., the baseline model and the strengthened model with T1-T3 inequalities, for the objective of total cost. To do this, we further enlarged the set of test instances from 25 clients to 100 clients (following the same implementation for the instance generation

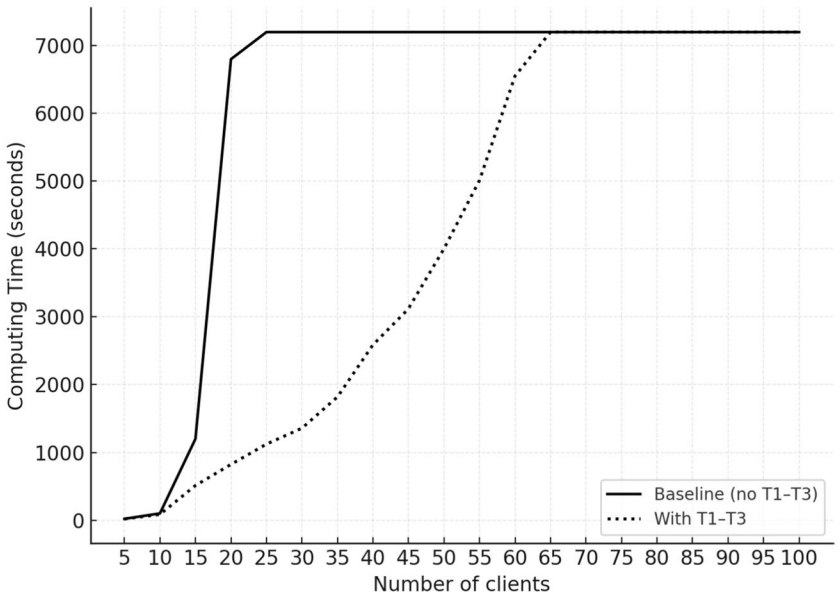


Fig. 10 Runtime vs. number of clients (5–100, step 5). Baseline (solid) and strengthened model (dotted)

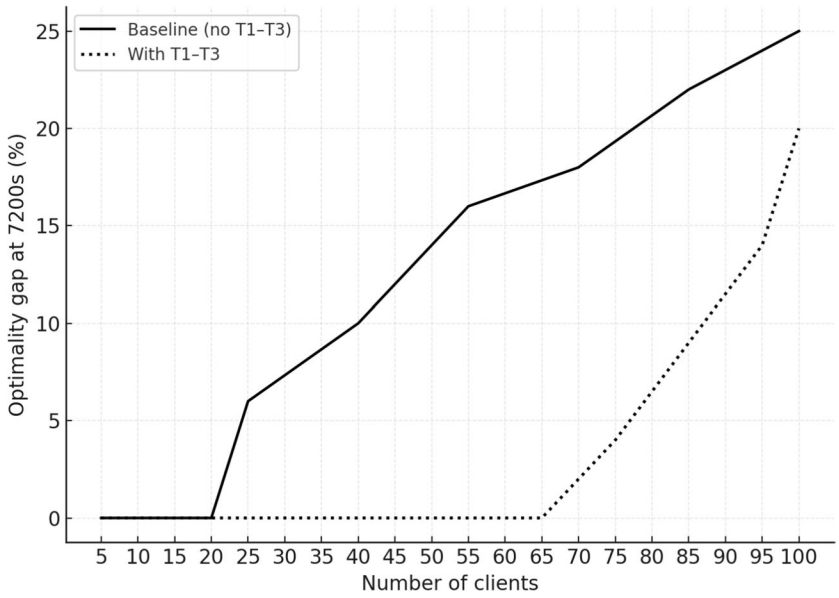


Fig. 11 Optimality gap at the 7,200 s time limit vs. number of clients. Baseline (solid) and strengthened model (dotted)

described in Section 6.6). The results show that we can enlarge the set of instances solvable to optimality within 7200 seconds to 65 clients.

In Fig. 11, for the same experiment setting, we show the optimality gap produced by the baseline model and the improved model (with T1-T3).

6.9 Comparison with a Three-Phase Heuristic

To deeply understand the performance of our model proposal, we propose a heuristic that constructs a mixed AV/HDV solution in three phases (the implementation has been carried out in Python). First, for each depot d it initializes singleton routes $(d \to c \to d)$ for those customers c whose round-trip legs satisfy the per-leg feasibility checks used in Algorithm 1 extended to vehicles $k \in K^a \cup K^{he}$, and, for AV candidates, the reliability gate $\lambda_{ij} \geq \Lambda^{\min}$ on both legs is verified. Customers that fail the checks are initialized as HDV in $K \setminus K^{he}$ singletons. Second, it performs a directed Clarke–Wright merge on each depot's pool: for any ordered pair $(i \to j)$ assigned to depot d, the directed saving is $s(i \to j) = d_{id} + d_{dj} - d_{ij}$, and pairs are considered in nonincreasing $s(i \to j)$ order. When a route r_i currently ends at i and a route r_j currently starts at j, the algorithm attempts the concatenation $r_i \leftarrow r_i \circ r_j$ provided the merged payload $q(r_i) + q(r_j)$ does not violate the capacity constraint and all boundary legs created by the merge are individually admissible under the same checks (the two new internal legs, and the two outer legs connecting to the depot). Whenever multiple vehicles can carry the merged load, the check is made against the most permissive admissible type (AV preferred if admissible, otherwise HDV). This phase terminates when no further admissible merge improves the depot's total distance. The third phase applies a route-improvement based on *2−opt* on each constructed route. Routes are represented as directed sequences $(v_0 = d, v_1, \ldots, v_m, v_{m+1} = d)$. For any pair of nonadjacent edges $(v_\alpha \to v_{\alpha+1})$ and $(v_\beta \to v_{\beta+1})$ with $0 \leq \alpha + 1 < \beta \leq m$, a 2−opt move proposes to remove these edges and reconnect as $(v_\alpha \to v_\beta)$ and $(v_{\alpha+1} \to v_{\beta+1})$, followed by reversing the intermediate segment $(v_{\alpha+1}, \ldots, v_\beta)$ to preserve a simple path. Because the network is directed, both the reconnection and the reversed segment are checked in their *actual* new directions. A move is accepted only if (i) it yields a strict decrease in travel cost, and (ii) *all* arcs in the modified route remain individually admissible for the current vehicle type: for AV routes, the reliability gate $\lambda_{ij} \geq \Lambda^{\min}$ holds on every arc; the state-of-charge is feasible for all AVs and HDVs in K^{he}; the payload on the route remains within capacity. Route improvement 2−opt

moves are repeated until no improving, admissible move exists (local optimum).

Algorithm 6 Three-Phase heuristic.

Input: Graph (N, A) with depots D, customers C, chargers/parkings P; distances d_{ij}; vehicle sets K^a (AV), K^{he} (electrified HDV if any), K^h (HDV thermal); capacities Q_k; SoC bounds M_k^a for $k \in K^a$; compatibility δ_{ij}^k, ω_v^k; reliability λ_{ij}; threshold $\Lambda^{\min}$.
Output: Feasible mixed-fleet solution x with routes and assigned vehicle types.

procedure HEURISTICMIXEDAVHDV($D, C, P, K^a, K^{he}, K^h$)
  $x \leftarrow \emptyset$;
  Phase I: Singleton initialization
  for all $d \in D$ **do**
    $\mathsf{Pool}(d) \leftarrow \emptyset$
  end for
  for all customer $c \in C$ **do**
    $d^\star \leftarrow \arg\min_{d \in D}(d_{dc} + d_{cd})$
    try AV: $\text{ok}^{av} \leftarrow \exists k \in K^a$: FEASIBLELEG($k, d^\star, c$) and FEASIBLELEG($k, c, d^\star$)
      if ok^{av} **then**
        pick any admissible $k \in K^a$; $\mathsf{r} \leftarrow (d^\star \rightarrow c \rightarrow d^\star)$ with vehicle k
      else
        fallback to HDV: pick $k \in K^h$ with FEASIBLELEG($k, d^\star, c$) and FEASIBLELEG($k, c, d^\star$)
        $\mathsf{r} \leftarrow (d^\star \rightarrow c \rightarrow d^\star)$ with vehicle k
      end if
      $\mathsf{Pool}(d^\star) \leftarrow \mathsf{Pool}(d^\star) \cup \{\mathsf{r}\}$
    end for

  Phase II: Directed Clarke–Wright merging (per depot)
  for all $d \in D$ **do**
    build ordered pair list $\mathcal{P}_d = \{(i \rightarrow j) : i \neq j, i, j \in C$ assigned to $d\}$ with savings $s(i \rightarrow j) = d_{id} + d_{dj} - d_{ij}$
    sort $\mathcal{P}_d$ by nonincreasing $s(i \rightarrow j)$
    for all $(i \rightarrow j) \in \mathcal{P}_d$ **do**
      let r_i be a route in $\mathsf{Pool}(d)$ that *ends* at i; let r_j be a route that *starts* at j (if they exist)
      if r_i and r_j exist and $\mathsf{r}_i \neq \mathsf{r}_j$ **then**
        r_{new}, ok $\leftarrow$ TRYMERGEDIRECTED($\mathsf{r}_i, \mathsf{r}_j, d$)
        if ok **and** COST(r_{new}) < COST(r_i) + COST(r_j) **then**
          replace $\mathsf{r}_i, \mathsf{r}_j$ in $\mathsf{Pool}(d)$ with r_{new}
        end if
      end if
    end for
  end for

  Phase III: Route improvement by directed 2-opt
  for all $d \in D$ **do**
    for all route $\mathsf{r} \in \mathsf{Pool}(d)$ **do**
      $\mathsf{r} \leftarrow$ TWOOPTIMPROVE(r)
    end for
  end for
  $x \leftarrow \bigcup_{d \in D} \mathsf{Pool}(d)$
  return x
end procedure

Algorithm 7 Feasibility checks and operators used by the heuristic.

function FEASIBLELEG(k, i, j)
  if $\delta_{ij}^k = 0$ **or** $\omega_i^k = 0$ **or** $\omega_j^k = 0$ **then**
    return false
  end if
  if $k \in K^a$ **and** $\lambda_{ij} < \Lambda^{\min}$ **then**
    return false
  end if
  return true
end function

function FEASIBLEROUTEAFTERINSERTION(r, k)
  Capacity: ensure total load on r is $\leq Q_k$
  Compatibility: $\delta_{uv}^k = 1$ for all arcs (u, v) in r and $\omega_v^k = 1$ for all visited nodes
  if $k \in K^a$ **then**
    Reliability: require $\lambda_{uv} \geq \Lambda^{\min}$ for all arcs (u, v) in r
    simulate SoC along r; if a leg (u, v) violates SoC, attempt to insert one charging node $p \in P$ near u; reset SoC to M_k^a after charge; re-simulate; if still infeasible, **return false**
  end if
  return true
end function

function TRYMERGEDIRECTED($\mathsf{r}_i, \mathsf{r}_j, d$)
  let k_i (resp. k_j) be the current vehicle types of r_i (resp. r_j)
  construct concatenation r_{cat} by linking the tail of r_i to the head of r_j, preserving depot d
  Pick vehicle type (most permissive):
  if $\exists k \in K^a$ s.t. FEASIBLEROUTEAFTERINSERTION($\mathsf{r}_{\text{cat}}, k$) **then** $k^\star \leftarrow$ such AV
  else if $\exists k \in K^h$ s.t. FEASIBLEROUTEAFTERINSERTION($\mathsf{r}_{\text{cat}}, k$) **then** $k^\star \leftarrow$ such HDV
  else
    return (r_{cat}, **false**)
  end if
  set vehicle of r_{cat} to $k^\star$
  return (r_{cat}, **true**)
end function

function TWOOPTIMPROVE(r)
  let $\mathsf{r} = (v_0 = d, v_1, \ldots, v_m, v_{m+1} = d)$ with vehicle k
  improved $\leftarrow$ **true**
  while improved **do**
    improved $\leftarrow$ **false**
    for $\alpha = 0$ **to** $m - 2$ **do**
      for $\beta = \alpha + 2$ **to** m **do**
        propose replacing edges $(v_\alpha \rightarrow v_{\alpha+1}), (v_\beta \rightarrow v_{\beta+1})$ with $(v_\alpha \rightarrow v_\beta)$ and $(v_{\alpha+1} \rightarrow v_{\beta+1})$ and reverse segment $(v_{\alpha+1}, \ldots, v_\beta)$
        if FEASIBLEROUTEAFTERINSERTION(proposed route, k) **and** COST(proposed) < COST(r) **then**
          $\mathsf{r} \leftarrow$ proposed; improved $\leftarrow$ **true**; **break**
        end if
      end for
      if improved **then**
        break
      end if
    end for
  end while
  return r
end function

In Algorithms 6 and 7, we report the pseudocode of the three-phase heuristic. In Fig. 12, we compare the optimality gap of the proposed improved model and the heuristic (the gaps are calculated using the same lower bounds as in the model gaps). Computing times of the heuristic are negligible with respect to the model running times, but the optimality gaps produced by the model are always lower than those produced by the heuristic.

6.10 Comparison with an Adaptive Large Neighborhood Search Algorithm

To further compare the modeling approach with the state-of-the-art heuristic approaches, we implemented (in Python) an Adaptive Large Neighborhood Search (ALNS) in the destroy/repair paradigm [26], enriched with EV/SoC feasibility [28] and an AV reliability gate. Candidate solutions encode a set of depot-rooted routes assigned to vehicles $k \in K^a \cup K^h$; AV routes carry an implicit state-of-charge process and allow on-the-fly insertion of charging stops at parking/charging nodes P. The scalarized objective is as in Eq. 93.

In the Initialization, we build a complete, feasible incumbent as follows. For each customer c, in random order, we scan all vehicles k and all insertion positions in their current route (d_k = depot) to minimize the objective increment Δz subject to feasibility: capacity, node/arc compatibility, and, for AV trial routes, (i) reliability gate $\lambda_{ij} \geq \Lambda^{\min}$ on every arc and (ii) SoC recursion with a single greedy charging insertion if a leg would deplete SoC (The charging insertion attempts one intermediate stop at the nearest charging node to the leg's tail; SoC is reduced by travel energy and reset to M_k^a upon charging). If no AV insertion is feasible for c, we try all HDVs; since HDVs have no SoC and no reliability gate, this fallback yields a complete feasible solution in our instances. Ties in Δz are broken by shorter travel distance.

The algorithm is iterative. At every iteration t, we maintain a set $\mathcal{D}$ of four destroy operators: *random removal* (uniformly remove q assigned customers), *Shaw/relatedness removal* (preferentially remove customers geographically close to a seed), *route removal* (drop all customers from a randomly chosen route), and *energy-critical removal* (remove customers far from the nearest charger). A destroy operator is selected by weighted roulette. Let $w_o^{(t)}$ be the current weight of operator $o \in \mathcal{D}$, then draw o with probability $w_o^{(t)} / \sum_{o'} w_{o'}^{(t)}$. The number of removed customers is $q = \max\{1, \lfloor \phi\, |C_{\text{assigned}}| \rfloor\}$ with $\phi \sim \text{Unif}[0.15, 0.35]$, truncated to $q \leq \min\{20, |C_{\text{assigned}}|\}$. The repair operator is fixed to *regret-2* insertion: for each to-be-inserted customer, we enumerate feasible insertions, rank by objective increment, and pick the one with the largest regret (second best − best); feasibility tests and on-the-fly charging follow the same rules as in initialization.

As for move acceptance and operator adaptation, let z_{curr} be the current value and z_{cand} the candidate after destroy/repair. We accept the candidate if it improves the objective, or with simulated annealing probability $\exp(-(z_{\text{cand}} - z_{\text{curr}})/T)$ otherwise. The temperature follows geometric cooling $T \leftarrow \gamma T$. Operator scores are updated as follows: if the candidate is accepted, the active destroy (and repair) operator receives a reward r_{acc}; if it also improves the global best, it receives r_{best}. Every I_w iterations, weights are smoothed by the reaction scheme $w_o^{(t+)} = (1-\rho)\, w_o^{(t)} + \rho \dfrac{\text{score}_o}{\text{count}_o}$, $o \in \{\text{random, Shaw, route, energy}\}$, and scores/counters are reset.

Feasibility checks (common to init/repair) takes into account (i) capacity: load on each route $\leq Q_k$; (ii) compatibility: node and arc masks ω, δ must be 1; (iii) AV reliability gate: $\lambda_{ij} \geq \Lambda^{\min}$ for every AV arc; (iv) AV energy/SoC: along each AV route, SoC decreases by leg energy; if a leg would violate SoC, insert at most one charging stop at the nearest $p \in P$ and resume; if still infeasible, the insertion is rejected.

The ALNS algorithm runs until a CPU time limit of 7200 seconds and returns the best solution found. In Algorithms

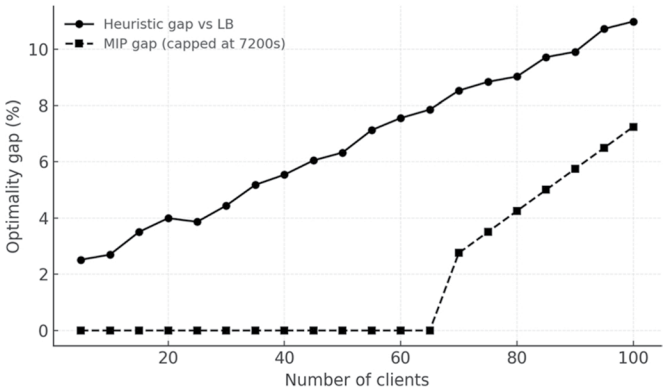


Fig. 12 Optimality gaps: Three-phase heuristic vs. MIP

8, 9, and 10, we report the whole pseudocode with the parameters used in the implementation.

Algorithm 8 ALNS.

Input: Instance $\mathcal{I} = \langle N, A, d_{ij}, K^a, K^h, \lambda_{ij}, \omega, \delta, P, \text{depots}\rangle$; weights $\alpha = (0.25, 0.25, 0.25, 0.25)$; reliability threshold $\Lambda^{\min} = 0.90$.
Output: Best solution $x^\star$ and value $z^\star$.
Parameters: SA temperature $T_0 = 0.01\,|z_{\text{init}}|$, cooling $\gamma = 0.9995$; operator reaction $\rho = 0.20$; weight update period $I_w = 200$ iterations; removal rate $\phi \sim \text{Unif}(0.15, 0.35)$; removal cap $q \le 20$.
Operator sets: Destroy $\mathcal{D}$ = {RANDOM, SHAW, ROUTEREM, ENERGYCRIT}; Repair $\mathcal{R}$ = {REGRET2}.
Initialization (Greedy Best-Position): build $x^{(0)}$ by inserting customers in random order into the best position of any vehicle route, minimizing Δz and enforcing feasibility via FEASIBLEAFTERINSERTION (Alg. 9).
Set current $x \leftarrow x^{(0)}$, $z \leftarrow z(x)$; best $x^\star \leftarrow x$, $z^\star \leftarrow z$; initialize operator weights $w_o \leftarrow 1$ for all $o \in \mathcal{D}$, scores $s_o \leftarrow 0$, counts $c_o \leftarrow 10^{-9}$; temperature $T \leftarrow T_0$; iteration $t \leftarrow 0$; start timer.
while elapsed time $< T_{\max}$ **do**
  $t \leftarrow t + 1$.
  Roulette destroy-pick: draw $u \sim \text{Unif}(0, 1)$ and select $o \in \mathcal{D}$ with $\mathbb{P}(o) = w_o / \sum_{o'} w_{o'}$.
  Removal size: let n_c be # assigned customers; draw $\phi \sim \text{Unif}(0.15, 0.35)$; set $q \leftarrow \min\{20, \max(1, \lfloor \phi n_c \rfloor)\}$.
  Destroy: $x^{\text{d}} \leftarrow \text{DESTROY}(x, o, q)$.
  Repair (Regret-2): $x^{\text{cand}} \leftarrow \text{REGRETTWOREPAIR}(x^{\text{d}})$ using feasibility from Alg. 9.
  if x^{cand} is infeasible **then**
    continue
  end if
  $z_{\text{cand}} \leftarrow z(x^{\text{cand}})$; $\Delta \leftarrow z_{\text{cand}} - z$.
  SA acceptance: accept if $\Delta < 0$ or $\exp(-\Delta/T) > \text{rand}()$.
  if accepted **then**
    $x \leftarrow x^{\text{cand}}, z \leftarrow z_{\text{cand}}; \quad r \leftarrow 1$.
    if $z < z^\star$ **then**
      $x^\star \leftarrow x, z^\star \leftarrow z; \quad r \leftarrow 5$
    end if
    Update $s_o \leftarrow s_o + r$, $c_o \leftarrow c_o + 1$ for the active destroy/repair.
  end if
  Cooling: $T \leftarrow \max(10^{-6}, \gamma T)$.
  if $t \bmod I_w = 0$ **then**
    for each $o \in \mathcal{D}$ **do**
      $w_o \leftarrow (1 - \rho)\, w_o + \rho\, (s_o / \max\{c_o, 10^{-9}\}); \quad s_o \leftarrow 0$, $c_o \leftarrow 10^{-9}$
    end for
  end if
end while
return $x^\star, z^\star$

Algorithm 9 Feasibility check function.

function FEASIBLEAFTERINSERTION(route, k)
  Capacity: cumulative delivered load on route $\le Q_k$.
  Compatibility: for all arcs (i, j) in route, $\delta_{ij}^k = 1$ and for all nodes v, $\omega_v^k = 1$.
  if $k \in K^a$ **then**
    Reliability gate: for all arcs (i, j), require $\lambda_{ij} \ge \Lambda^{\min}$.
    Energy/SoC: simulate SoC along the route; if a leg (i, j) violates SoC, attempt a *single* charging insertion at the nearest $p \in P$ to i; after charging, reset SoC to M_k^a and continue. If still infeasible, **return** infeasible.
  end if
  return feasible route
end function

Algorithm 10 Destroy and repair operators.

function DESTROY(x, o, q)
  if $o =$ RANDOM **then**
    remove q customers uniformly at random.
  else if $o =$ SHAW **then**
    pick a seed customer c; iteratively add the most related customer (smallest geometric distance to the current removed set) until q are removed.
  else if $o =$ ROUTEREM **then**
    select a random nonempty route; remove all its customers (truncate to q if needed).
  else if $o =$ ENERGYCRIT **then**
    score customers by distance to nearest charger (larger first); remove top q.
  end if
  return partially destroyed solution
end function
function REGRETTWOREPAIR(x)
  $U \leftarrow$ list of unassigned customers.
  while $U \neq \emptyset$ **do**
    for each $c \in U$ **do**
      enumerate all vehicles k and insertion positions; for each, build trial route and call FEASIBLEAFTERINSERTION; compute objective increment Δz.
      let $\Delta z_{(1)}(c) \le \Delta z_{(2)}(c) \le \ldots$ be the best increments; set $\text{regret}(c) = \Delta z_{(2)}(c) - \Delta z_{(1)}(c)$ (if only one feasible position, use $\text{regret}(c) = \Delta z_{(1)}(c)$; if none, mark c infeasible).
    end for
    if all c infeasible **then**
      return infeasible
    end if
    choose $c^\star = \arg\max_c \text{regret}(c)$ and perform the corresponding best insertion.
    update U and the incumbent routes.
  end while
  return repaired solution
end function

In Table 15, we report the overall comparison of the model performance with respect to the three-phase heuristic and the ALNS metaheuristic previously presented. The strengthened formulation (+T1–T3) markedly improves computational performance and scalability. It closes the optimality gap to 0% up to 65 clients (vs. 20 clients for the baseline MIP) and reduces runtime substantially at fixed size (e.g., 6,800 → 823 s at 20 clients, an ∼8.3× speedup; 1,200 → 512 s at 15 clients, ∼2.3×). Beyond 65 clients, +T1–T3 still yields tighter gaps than the baseline (e.g., 7.0% vs. 22.0% at 80 clients; 20.0% vs. 25.0% at 100 clients within the 7, 200 s cap). The constructive Clarke–Wright+2opt heuristic (three-phase heuristic) is fast but consistently farther from the lower bound (gaps grow from 2.5% to 11.0% as size increases). The ALNS metaheuristic dominates the simple heuristic across all sizes (e.g., 1.5% vs. 2.5% at 5 clients; 9.8% vs. 11.0% at 100 clients) and exhibits strong behavior: its best solution is typically found well before the time cap (e.g., 45 s at 10 clients; 4,300 s at 100). Notably, at the largest scales, ALNS produces better (primal) solutions than the MIP within the time limit (gap 9.8% vs. 20.0% at 100 clients against the same lower bound), whereas for small/medium instances, the strengthened MIP remains preferable due to provable optimality and shorter runtimes.

6.11 MIP and Heuristics Hypervolume Analysis

In the following, we assess the quality of the solution set through the *hypervolume* (HV) indicator computed in the full multi-objective space. Let the objective vector be $\mathrm{z}(x) = \big(z_1(x), z_2(x), z_3(x), z_4(x)\big)$, where z_1, z_2, z_3 are to be minimized (emissions, cost, energy), while z_4 is a reliability-related objective to be maximized.

For each method $m \in \{\text{MIP}, \text{three-phase heuristic}, \text{ALNS}\}$, we first extract the set of non-dominated solutions $\mathcal{P}^{m,\mathrm{ND}}$ from the produced solution pool $\mathcal{P}^m$.

To compute HV comparably across methods, we normalize objectives on the union of solution sets $\mathcal{P}^{\cup} = \cup_m \mathcal{P}^{m,\mathrm{ND}}$. Define, for $\ell \in \{1, 2, 3\}$ (minimization objectives),

$$\bar{z}_\ell(x) = \frac{z_\ell(x) - z_\ell^{\min}}{z_\ell^{\max} - z_\ell^{\min} + \varepsilon}, \qquad z_\ell^{\min} = \min_{y \in \mathcal{P}^{\cup}} z_\ell(y), \quad z_\ell^{\max} = \max_{y \in \mathcal{P}^{\cup}} z_\ell(y), \tag{98}$$

and for the reliability objective z_4 (maximization),

$$\bar{z}_4(x) = \frac{z_4^{\max} - z_4(x)}{z_4^{\max} - z_4^{\min} + \varepsilon}, \qquad z_4^{\min} = \min_{y \in \mathcal{P}^{\cup}} z_4(y), \quad z_4^{\max} = \max_{y \in \mathcal{P}^{\cup}} z_4(y). \tag{99}$$

With Eqs. 98–99, all normalized objectives $\bar{z}_\ell$ are *to be minimized* and lie (approximately) in [0, 1]. We set the hypervolume reference point to $\mathrm{r} = (1, 1, 1, 1)$.

Given a normalized non-dominated set $\bar{\mathcal{P}}^{m,\mathrm{ND}}$ for method m, its hypervolume can be defined [3] as

$$\mathrm{HV}\big(\bar{\mathcal{P}}^{m,\mathrm{ND}}\big) = \mathrm{vol}\left(\bigcup_{\bar{\mathrm{z}} \in \bar{\mathcal{P}}^{m,\mathrm{ND}}} [\bar{\mathrm{z}}, \mathrm{r}] \right), \tag{100}$$

where $[\bar{\mathrm{z}}, \mathrm{r}]$ denotes the axis-aligned hyper-rectangle with opposite corners $\bar{\mathrm{z}}$ and r. Higher HV indicates a better approximation of the (unknown) Pareto set, rewarding both convergence and diversity in the objective space.

To visualize how quickly each approach builds a high-quality Pareto approximation, we also plot cumulative HV values: ordering the non-dominated points of each method by the associated scalarization parameter/exploration index (the same order used to generate the solution pool), we compute HV_t on the first $t = 30$ non-dominated points.

Figures 13, 14, and 15 summarize the quality of the Pareto sets when all objectives are considered simultaneously.

To assess whether the hypervolume conclusions were sensitive to the number of sampled solutions, we extended the analysis beyond the initial pool of 30 points. The

Table 15 Summary of the performance of the MIP and the heuristics. Baseline denotes the original MIP; +T1–T3 is the strengthened formulation with valid inequalities. Heuristic gaps refer to the Clarke–Wright+`2opt` algorithm and are computed against the same lower bound used for MIP gaps. ALNS runs with a 7, 200 s cap; we report the best-found gap versus the same lower bound and the time (in seconds) when that best solution was first reached

# Clients	MIP Runtime (s)		MIP Gap at 7,200 s (%)		Heuristic	ALNS	
	Baseline	+T1–T3	Baseline	+T1–T3	Gap vs LB (%)	Gap vs LB (%)	Time-to-Best (s)
5	20	18	0.0	0.0	2.5	1.5	12
10	100	85	0.0	0.0	2.8	1.9	45
15	1,200	512	0.0	0.0	3.8	2.9	160
20	6,800	823	0.0	0.0	4.5	3.6	420
25	7,200	1,123	6.0	0.0	3.9	3.2	820
40	7,200	2,589	10.0	0.0	5.5	4.8	1,300
55	7,200	5,012	16.0	0.0	7.1	6.2	1,950
65	7,200	7,200	18.0	0.0	7.8	7.0	2,600
80	7,200	7,200	22.0	7.0	9.0	8.4	3,400
100	7,200	7,200	25.0	20.0	11.0	9.8	4,300

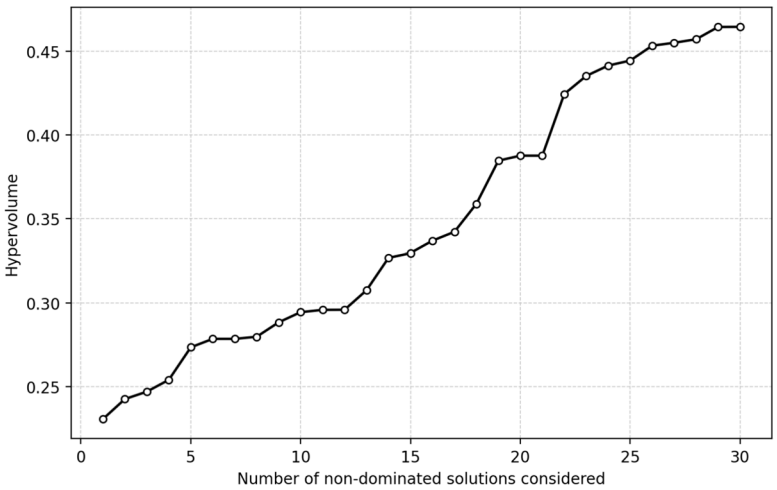


Fig. 13 MIP: cumulative hypervolume computed on its respective non-dominated set with the same normalization and reference point as in Eqs. 98–100

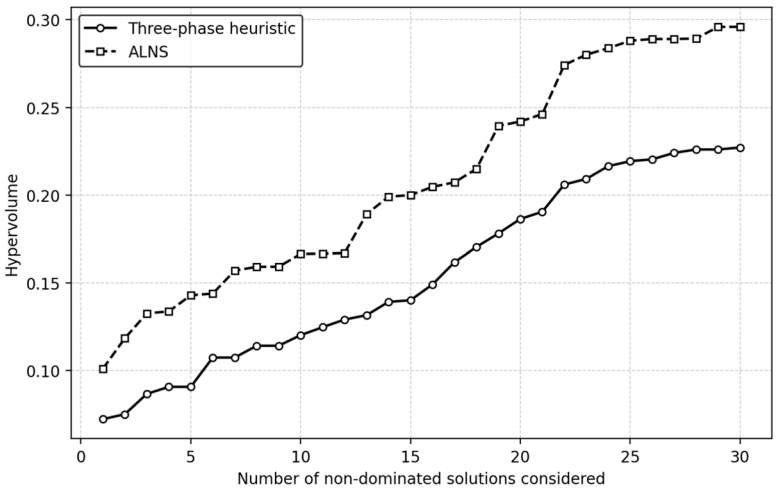


Fig. 14 Heuristics: cumulative hypervolume for the three-phase heuristic and ALNS, computed on their respective non-dominated sets with the same normalization and reference point as in Eqs. 98–100

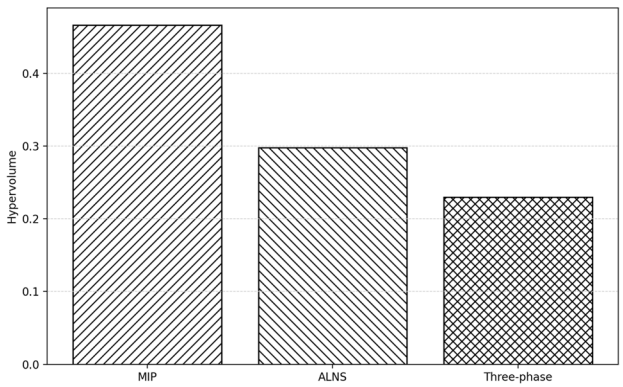


Fig. 15 Final hypervolume comparison after 30 non-dominated solutions

additional runs showed that the weighted-sum MIP benchmark already reaches a stable hypervolume at 30 sampled solutions, whereas ALNS exhibits a further increase up to about 150 sampled solutions and then remains flat up to the maximum tested budget. The three-phase heuristic remains unchanged throughout. Therefore, the main qualitative conclusions are stable, but the extended test confirms that 30 sampled solutions are fully sufficient only for the exact weighted-sum benchmark, while a somewhat larger pool is needed to observe the complete stabilization of ALNS.

Within the sampled solution pools considered here, the weighted-sum MIP approach provides an exact benchmark on the supported solutions generated by the scalarization scheme, but its hypervolume stabilizes very early. By contrast, ALNS explores a broader sampled trade-off region and achieves additional hypervolume gains before stabilizing. The three-phase heuristic remains the weakest method in terms of sampled hypervolume.

We conclude this analysis, noting that the hypervolume analysis reported should be interpreted as a comparative indicator of solution-set quality within the sampled pool of non-dominated solutions generated by each method. Since the problem is four-dimensional and the evaluation is based on a finite number of sampled non-dominated points, the resulting hypervolume values do not provide an exhaustive description of the entire Pareto structure. Rather, they offer a comparative measure of convergence and diversity for the solution sets effectively explored by the different methods under the adopted computational budget.

7 Conclusions

This paper introduced a mixed-integer programming framework for routing a heterogeneous fleet of autonomous vehicles (AVs) and human-driven vehicles (HDVs) in freight distribution. Beyond standard VRP/EVRP elements, four ingredients are central: *(i)* a reliability-gated arc admissibility rule for AVs controlled by a single threshold $\Lambda^{\min}$, *(ii)* LP-relaxation-based screening intervals (via dual multipliers of the reliability constraints) used as an event-trigger to decide when changes in λ_{ij}^{k} warrant re-optimization, *(iii)* a lightweight feedback-driven weight adaptation scheme for the multi-objective scalarization (emissions, energy, cost, reliability), and *(iv)* three families of valid inequalities (T1–T3) that strengthen the formulation by tightening the load/energy structure and pruning infeasible energy patterns. The resulting framework supports dynamic rerouting under time-varying network reliability while maintaining tractability and providing clear operational "corridor" rules for AV deployment.

We validated the approach on a Northern-Italy case study with real road distances and calibrated demands/reliabilities, and we embedded the model into a discrete-time decision framework in which arc reliabilities $\lambda_{ij}^{k}(t)$ are periodically updated (e.g., via the logistic-regression pipeline described in Section 5.1). The dynamic layer shows how stability intervals can prevent unnecessary re-optimizations while guaranteeing feasibility when updated reliabilities invalidate active AV arcs (i.e., when $\lambda_{ij}^{k} < \Lambda^{\min}$ on used arcs).

From a computational standpoint, the strengthening provided by T1–T3 yields substantial scalability gains. In the cost-minimization environment used for the computational comparison, the baseline formulation solves instances up to 20 clients within the 7,200 s budget and reaches the time cap starting at 25 clients (with a nonzero gap). The strengthened formulation (+T1–T3) attains 0% optimality gap up to 65 clients (with the first instances hitting the time cap at 65) and delivers markedly tighter gaps at larger sizes (e.g., improved performance at 80–100 clients within the same time limit). To complement exact optimization, we also implemented a constructive three-phase heuristic and an ALNS metaheuristic: both provide high-quality feasible solutions quickly, and ALNS in particular remains competitive at the largest instance sizes when the MIP is time-limited.

Managerially, the proposed framework provides actionable levers and deployment guidance. *(i)* $\Lambda^{\min}$ acts as a policy parameter converting safety/reliability targets into auditable AV corridors. The rule $\Lambda^{\min} = \beta^{1/L}$ translates a route-level reliability target β into an arc-level admissibility threshold using a data-driven proxy L, while the feasibility cap avoids over-stringent settings that could disconnect customers. *(ii)* The emissions–reliability trade-off and the adaptive weight update mechanism offer a practical way to explore balanced Pareto solutions without modifying the solver or introducing additional variables. *(iii)* The event-triggered rerouting procedure (based on stability intervals and feasibility checks) supports periodic planning with override actions under reliability shocks (e.g., weather/congestion disruptions), enabling operators to control recomputation frequency and response time. Finally, the framework naturally supports phased deployment strategies: AV-preferred corridors in high-reliability regions, mixed operations with HDV fallback, and explicit human-override policies when reliability drops below threshold.

As for limitations and future work, the current model uses fixed recharge increments and deterministic time/energy relationships, and the case study adopts simplifying assumptions (e.g., complete arc compatibility and homogeneous AV arc reliabilities). Extending the framework to richer charging representations (variable-rate charging, queueing), time windows and service times, correlated reliability across arcs, and more explicit uncertainty treatments (e.g., stochastic energy consumption) are natural directions. On the reliability side, a promising extension is to incorporate a complementary HDV risk gate (e.g., $\Lambda_{\mathrm{HDV}}^{\min}$) capturing human-factor risks (fatigue, incident likelihood) to support unified safety governance across mixed fleets. Continuous learning and calibration of $\lambda_{ij}^{k}(t)$, and tighter integration with real-time traffic and charging forecasts, would further improve responsiveness and operational realism. Another limitation is that the mapping from the route-level target β to the arc-level threshold $\Lambda^{\min}$ relies on an independence approximation across arcs. Although this yields a simple and auditable admissibility rule, it does not explicitly model spatial dependence in reliability conditions, which would be a valuable direction for future work.

Author Contributions A.P.: Conceptualization. M.C.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing.

Funding Open access funding provided by Università degli Studi di Roma Tor Vergata within the CRUI-CARE Agreement. The authors did not receive support from any organization for the submitted work.

Availability of Data and Material The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing Interests Authors have no conflict of interest to declare.

Ethics Approval and Consent to Participate Not applicable

Consent for Publication Not applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the

source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. 3GPP. 3GPP TR 22.886: Study on enhancement of 3gpp support for 5g v2x services, vol. 886 (2018)
2. AINews. Jd logistics launches level-4 autonomous trucks for automated delivery (2025). https://www.aibase.com/news/19469
3. Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. **19**(1), 45–76 (2011)
4. Beirigo, B. A., Schulte, F., Negenborn, R. R.: Integrating people and freight transportation using shared autonomous vehicles with compartments. IFAC-PapersOnLine, 51(9):392–397 (2018). 15th IFAC Symposium on Control in Transportation Systems CTS 2018
5. Breque, M., Nul, L.D., Petridis, A.: Industry 5.0: Towards a sustainable, human-centric and resilient European industry. European Commission, DG Research and Innovation (2021)
6. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
7. DB Schenker Consulting. Einride | db schenker collaboration (2019). https://www.dbschenker.com/co-en/innovation-/einride
8. Csiszár, C., Földes, D.: System model for autonomous road freight transportation. Promet-Traffic & Transportation **30**(1), 93–103 (2018)
9. De Maio, A., Ghiani, G., Laganà, D., Manni, E.: Sustainable last-mile distribution with autonomous delivery robots and public transportation. Transportation Research Part C: Emerging Technologies **163**, 104615 (2024)
10. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. **64**(6), 1388–1405 (2016)
11. Desrochers, M., Laporte, G.: Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. Oper. Res. Lett. **10**(1), 27–36 (1991)
12. Gružauskas, V., Baskutis, S., Navickas, V.: Minimizing the trade-off between sustainability and cost effective performance by using autonomous vehicles. J. Clean. Prod. **184**, 709–717 (2018)
13. Guo, Y., Chen, X., Yang, Y.: Multimodal transport distribution model for autonomous driving vehicles based on improved alns. Alex. Eng. J. **61**(4), 2939–2958 (2022)
14. California HVIP. Lion LION6 battery electric truck — california hvip eligible vehicles (2026). https://californiahvip.org/vehicles/lion-lion6-battery-electric-truck/
15. Jennings, D., Figliozzi, M.: Study of road autonomous delivery robots and their potential effects on freight efficiency and travel. Transp. Res. Rec. **2674**(9), 1019–1029 (2020)
16. Jeong, J., Ghaddar, B., Zufferey, N., Nathwani, J.: Adaptive robust electric vehicle routing under energy consumption uncertainty. Transportation Research Part C: Emerging Technologies **160**, 104529 (2024)
17. Kara, İ, Laporte, G., Bektaş, T.: A note on the lifted miller-tucker-zemlin subtour elimination constraints for the capacitated vehicle routing problem. Eur. J. Oper. Res. **158**(3), 793–795 (2004)
18. Lee, C., An, D.: Decision-making in fallback scenarios for autonomous vehicles: Deep reinforcement learning approach. Appl. Sci. **13**(22) (2023)
19. Masood, K., Zoppi, M., Fremont, V., Molfino, R.M.: From drive-by-wire to autonomous vehicle: Urban freight vehicle perspectives. Sustainability **13**(3) (2021)
20. Monios, J., Bergqvist, R.: The transport geography of electric and autonomous vehicles in road freight networks. J. Transp. Geogr. **80**, 102500 (2019)
21. Monios, J., Bergqvist, R.: Shipment sizing for autonomous trucks of road freight. Int. J. Logist. Manag. **32**(2), 413–433 (2021)
22. MOTORWATT. Deepway company (2026). https://ev.motorwatt.com/ev-manufacturers/deepway
23. Nielsen, P., Dahanayaka, M., Perera, H.N., Thibbotuwawa, A., Kiliç, D.K.: A systematic review of vehicle routing problems and models in multi-echelon distribution networks. Supply Chain Analytics **7**, 100072 (2024)
24. Nuro, Inc. Introducing our next-generation nuro (2021). https://www.nuro.ai/blog/introducing-our-next-generation-nuro
25. Osorio-Mora, A., Saldanha-da-Gama, F., Toth, P.: A risk-averse latency location-routing problem with stochastic travel times. Eur. J. Oper. Res. **321**(3), 837–850 (2025)
26. Røpke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**(4), 455–472 (2006)
27. International SAE. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (SAE j3016) (2021). https://webstore.ansi.org/standards/sae/sae30162021
28. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. **48**(4), 500–520 (2014)
29. Schulz, A.: Using infeasible path cuts to solve electric vehicle routing problems with realistic charging functions exactly within a branch-and-cut framework. EURO Journal on Transportation and Logistics **13**(1), 100131 (2024)
30. Shahandasht, M., Pudasaini, B.: and Sean Logan McCauley, pp. 1–111. Autonomous vehicles and freight transportation analysis. Department of Civil Engineering, The University of Texas at Arlington (2019)
31. Shamsuddoha, M., Kashem, M. A., Nasir, T.: A review of transportation 5.0: Advancing sustainable mobility through intelligent technology and renewable energy. Future Transportation **5**(1) (2025)
32. Thiebaut, K., Pessoa, A.: Approximating the chance-constrained capacitated vehicle routing problem with robust optimization. 4OR - A. Quarterly Journal of Operations Research **21**, 513–531 (2023)
33. Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA (2002)
34. Waymo. Waymo unveils 24/7 robotaxi service in silicon valley, pioneering autonomous rides in mountain view and beyond (2025). https://blog.waymo.com/2022/12/wheels-up-for-waymo-as-we-expand.htm
35. Wei, W., Zhang, F., Liu, W., Lodewijks, G.: Modelling the traffic in a mixed network with autonomous-driving expressways and non-autonomous local streets. Transportation Research Part E: Logistics and Transportation Review **134**, 101855 (2020)
36. Yaman, H.: Formulations and valid inequalities for the heterogeneous vehicle routing problem. Math. Program. **106**, 365–390 (2006)
37. Zajac, S., Huber, S.: Objectives and methods in multi-objective routing problems: a survey and classification scheme. Eur. J. Oper. Res. **290**(1), 1–25 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.